HYBRID CONTROL OF AIR TRAFFIC MANAGEMENT SYSTEMS

by

Claire Jennifer Tomlin

Memorandum No. UCB/ERL M98/57

30 September 1998

HYBRID CONTROL OF AIR TRAFFIC MANAGEMENT SYSTEMS

by

Claire Jennifer Tomlin

Memorandum No. UCB/ERL M98/57

30 September 1998

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Hybrid Control of Air Traffic Management Systems

by

Claire Jennifer Tomlin

B.A.Sc. (University of Waterloo) 1992

M.Sc. (Imperial College of Science, Technology, and Medicine) 1993

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 at BERKELEY

Committee in charge:

Professor S. Shankar Sastry, Chair
Professor Pravin P. Varaiya
Professor Alexandre J. Chorin
Professor Thomas A. Henzinger

Fall 1998

The dissertation of Claire Jennifer Tomlin is approved:

18 Sept. 1998

Chair Date

16 Sept '98

Date

8/14/98

Date

Thomas A. Henzinger 8/21/98

Date

University of California at Berkeley

Fall 1998

Abstract

Hybrid Control of Air Traffic Management Systems

by

Claire Jennifer Tomlin
Doctor of Philosophy in Engineering-Electrical Engineering and Computer Sciences
University of California at Berkeley
Professor S. Shankar Sastry, Chair

Today's crowded skies and ever-increasing demand for air travel, coupled with new technologies for navigation and surveillance, are fueling a change in the way that the Federal Aviation Administration manages air traffic. Current Air Traffic Control (ATC) practice manually routes aircraft along predefined paths between "fixes", using radar track and flight information from plan view displays and voice communication over radio channels. The use of Global Positioning Systems and datalink communication will enable automation of some ATC functionality, such as the prediction and resolution of trajectory conflicts between aircraft. For such a safety critical system, the integrity and acceptance of new automated control functionality depends on a *provably-safe* design, which requires accurate system models, and procedures for verifying and synthesizing safe control actions.

We present a model and controller synthesis scheme for a *nonlinear hybrid automaton*, a system that combines discrete event dynamics with nonlinear continuous dynamics. The discrete event dynamics model linguistic and qualitative information, such as the flight mode of an aircraft or the interaction between several aircraft. Discrete event models also naturally accommodate mode switching logic, which is triggered by events internal or external to the system. The continuous dynamics model the physical processes themselves, such as the continuous response of an aircraft to the forces of aileron and throttle. We include input variables to model both continuous and discrete control and disturbance parameters.

We translate safety specifications into restrictions on the system's reachable sets of states. Then, using analysis based on two-person zero-sum game theory for automata and continuous dynamical systems, we derive Hamilton-Jacobi equations whose solutions describe the boundaries of reachable sets. These equations are the heart of our general controller synthesis technique for hybrid systems, in which we calculate feedback control laws for the continuous and discrete variables which guarantee that the hybrid system remains in the "safe subset" of the reachable set. We present the extension of a level set method to compute numerical solutions of the Hamilton-Jacobi equations. Throughout, we demonstrate our techniques on examples of interesting nonlinear hybrid automata modeling aircraft conflict resolution and autopilot flight mode switching.

S. Shankar Sastry
Chair

Contents

List of Figures v

1 Introduction 1
1.1 Overview 5
1.2 Notation 6

2 Algorithms for Distributed Air Traffic Management 7
2.1 Overview of the Current System 7
2.2 Technologies to Enable Change 14
2.3 Proposed Architecture 15
2.4 Motivating Examples 17
2.4.1 Conflict Resolution for Aircraft 18
2.4.2 Flight Mode Switching and Envelope Protection 25

3 Nonlinear Hybrid System Model 30
3.1 Background 31
3.1.1 Finite Automata 31
3.1.2 Nonlinear Continuous-Time Dynamics 32
3.2 Nonlinear Hybrid Automata 33
3.3 Controlled Hybrid Systems 36
3.4 Examples 39
3.4.1 Three-Mode Conflict Resolution Example 39
3.4.2 Seven-Mode Conflict Resolution Example 41
3.4.3 Flight Mode Switching and Envelope Protection Example . . . 43

4 Evolution of Boundaries for Discrete and Continuous Games 45
4.1 Discrete Hamilton-Jacobi Equation 46
4.2 Continuous-Time Hamilton-Jacobi Equation 49
4.2.1 Example: The $SE(2)$ Aircraft 60

5 **Controller Synthesis for Nonlinear Hybrid Systems** **65**
5.1 Algorithm 67
5.2 Remarks 72

6 **Application to Distributed Air Traffic Management** **74**
6.1 Conflict Resolution for Two Aircraft in $SE(2)$ 75
6.1.1 Continuous Dynamics 75
6.1.2 Controller Synthesis for Three-Mode Example 78
6.1.3 Controller Synthesis for Seven-Mode Example 84
6.2 Mode Switching for the Longitudinal Axis Dynamics of a CTOL Aircraft 85
6.2.1 Continuous Dynamics 85
6.2.2 Controller Synthesis 96

7 **Computing Boundaries of Safe Sets** **98**
7.1 A Level Set Method for Boundary Approximation 99
7.2 Other Methods 103
7.2.1 Approximating Dynamics with Differential Inclusions 103
7.2.2 Approximating non-smooth sets with smooth sets 104

8 **Future Work** **106**

Bibliography **110**

List of Figures

2.1 Bay Area airports, TRACON, and part of Oakland Center. 10
2.2 A flight strip from the Oakland Center. 10
2.3 Two screens in a typical glass cockpit: (a) a horizontal profile of way points (into Los Angeles airport); (b) an "artificial horizon" showing the current pitch and roll angles of the aircraft, the airspeed and altitude, and the current flight mode. The first three columns in the flight mode are the throttle-vertical-lateral modes, the fourth is the autopilot mode. ARM means "waiting for the throttle to reach required value", MCP SPD means "speed is controlled to the entry in the mode control panel", HDG SEL means "heading is controlled to the entry in the mode control panel", CMD means "pilot has command over pitch and roll values". 13
2.4 Proposed framework for on-board planning and control. 16
2.5 Aircraft Zones. 19
2.6 Conflict Resolution Algorithm. 20
2.7 (a) Two aircraft in a conflict scenario; (b) The relative configuration, showing the relative protected zone. 21
2.8 Two aircraft in three modes of operation: in modes 1 and 3 the aircraft follow a straight course and in mode 2 the aircraft follow a half circle. The initial relative heading (120°) is preserved throughout. 23
2.9 Two aircraft in seven modes of operation: in modes 1, 3, 5, and 7 the aircraft follow a straight course and in modes 2, 4, and 6 the aircraft follow arcs of circles. Again, the initial relative heading (120°) is preserved throughout. 24
2.10 A planar aircraft in flight with attached axes about its center of mass. 26
2.11 (a) Simplified Aerodynamic Flight Envelope in (V, γ)-space: axes are airspeed V, flight path angle γ; (b) Simplified Aerodynamic Flight Envelope in $(h, V, \dot{h})$-space: axes are altitude h, airspeed V, vertical speed $\dot{h}$. 28

3.1 Nonlinear Hybrid Automaton. 35

3.2 Composition of Plant H and State-State Feedback Controller H_c to form the controlled system $\overline{H}$. The plant is assumed to be directly observable, as shown. 38

3.3 In q_1 the aircraft follow a straight course, in q_2 the aircraft follow a half circle; in q_3 the aircraft return to a straight course. 40

3.4 Hybrid automaton modeling seven-mode conflict resolution maneuver. 42

4.1 The capture set G^o, its outward pointing normal ν, and the cones of vector field directions at points on ∂G. 50

4.2 The left column displays four cases of optimal trajectories, starting at x at time s_i, and ending at state x_i at time 0, where $0 > s_1 > s_2 > s_3 > s_4$. The right column displays $J^*(x,t)$ for fixed x. Note that the standard variational problem produces states that can change from "unsafe" to "safe". The figure at the bottom of the right column displays the result of modifying the Hamilton-Jacobi equation so that, once $J^*(x,t)$ is negative, its evolution is non-increasing in negative time. 56

4.3 (a) The sets $\{x \in X \mid J^*(x,0) = 0\}$, $\{x \in X \mid J^*(x,t_1) = 0\}$, $\{x \in X \mid J^*(x,t_2) = 0\}$ for $0 > t_1 > t_2$. (b) The fixed point $\{x \in X \mid J^*(x) < 0\}$, $\{x \in X \mid J^*(x) = 0\}$, and $\{x \in X \mid J^*(x) > 0\}$. 59

4.4 The set $G = \{(x_r, y_r), \psi_r \in (0,\pi) \mid x_r^2 + y_r^2 \leq 5^2\}$ (cylinder) and the set $\{x \in X \mid J^*(x,t) = 0\}$ for $t < 0$ being the time of the first switch in either $s_1(t)$ or $s_2(t)$. The second picture is a top view of the first. 63

4.5 Switching law governing the two aircraft system with angular velocity control inputs. The law is least restrictive in that the control u is not restricted when the state is in $\{x \in X \mid J^*(x,t) > 0\}$. The diagonal transitions in the automaton for the boundary of $\{x \in X \mid J^*(x,t) = 0\}$ are not labeled for legibility. In practice, t should be chosen large enough to take into account aircraft in the alert zone. 64

5.1 In q_1, the portion of the unsafe set which intersects the invariant of q_2 may be made "safe" by switching from q_1 to q_2. 66

5.2 The computation of *Reach*(G, E) in a single discrete state q. 70

6.1 The set $\{x \in X \mid J^*(x,t) \leq 0\}$ shown in the (x_r, y_r)-plane for $[\underline{v}_1, \overline{v}_1] = [2,4]$, $[\underline{v}_2, \overline{v}_2] = [1,5]$ and (a) $\psi_r = \pi/2$, (b) $\psi_r = 0$, (c) $\psi_r = -\pi/4$, (d) $\psi_r = -\pi/2$. 78

6.2 $J^*_{G_i}(x) \leq 0$ for (a) Modes 1 and 3 ($i = 1,3$), $\omega_1 = \omega_2 = 0$ (the jagged edge means the set extends infinitely), (b) Mode 2 ($i = 2$), $\omega_1 = \omega_2 = 1$. In both cases, $\psi_r = 2\pi/3$, and $v_1 = v_2 = 5$. 80

6.3 $(W^0)^c$. 80

6.4 $(W^{-1})^c$. The jagged edge in q_3 means that the set extends infinitely. 81

6.5 (a) $Pre_1(W^{-1})$ and $Pre_2(W^{-1})$ in q_1; (b) $Reach(Pre_2(W^{-1}), Pre_1(W^{-1}))$ in q_1. 82

6.6 $(W^{-2})^c$. 82
6.7 $Reach(Pre_2(W^{-2}), Pre_1(W^{-2}))$ in q_1. 83
6.8 $(W^*)^c = (W^{-3})^c$. 83
6.9 Showing the enabling and forcing boundaries for σ_1 in state q_1; and the result of increasing the radius of the turn in the avoid maneuver to increase W^*. 84
6.10 $J^*_{G_i}(x) \leq 0$ for (a) Modes 1 and 7 ($i = 1,7$), $\omega_1 = \omega_2 = 0$ and $[\underline{v}_1, \overline{v}_1] = [2,4]$, $[\underline{v}_2, \overline{v}_2] = [1,5]$ (the jagged edge means the set extends infinitely); (b) Modes 3 and 5 ($i = 3,5$), $\omega_1 = \omega_2 = 0$ and $[\underline{v}_1, \overline{v}_1] = [2,4]$, $[\underline{v}_2, \overline{v}_2] = [1,5]$; (c) Mode 4 ($i = 4$), $\omega_1 = \omega_2 = 1$ and $v_1 = v_2 = 5$; and (d) Modes 2 and 6 ($i = 2,6$), $\omega_1 = \omega_2 = -1$ and $v_1 = v_2 = 5$. In all cases, $\psi_r = 2\pi/3$. 85
6.11 $(W^*)^c = (W^{-7})^c$ in q_1. The enabling and forcing boundaries for σ_1 are shown, and the controller $(\delta_c, Inv_c) \in H_c$ may be constructed as shown. 86
6.12 Computing the boundary ∂J^a. 89
6.13 Computing the set $\{(V,\gamma) \mid J_1^*(V,\gamma) = 0\}$. 90
6.14 The set $W^*_{V\gamma}$ in (V,γ)-space, with control law as indicated. Values used are for a DC-8: $\gamma_{min} = -\pi/8$ rad, $\gamma_{max} = \pi/8$ rad, $V_{min} = 180$ m/s, $V_{max} = 240$ m/s, $\theta_{min} = -\pi/8$ rad, $\theta_{max} = \pi/8$ rad, $T_{min} = 40$ kN, $T_{max} = 80$ kN. 90
6.15 Upper left boundary and lower right boundary of $F_{V\gamma}$. 94
6.16 The set $W^*_{hV\dot{h}}$ in $(h, V, \dot{h})$-space, with control law as indicated. Altitudes are $h_{min} = 10kft$, $h_{max} = 51kft$. 96

7.1 $\{x \in X \mid J^*(x,t) \leq 0\}$ shown in the (x_r, y_r)-plane for $[\underline{v}_1, \overline{v}_1] = [2,4]$, $[\underline{v}_2, \overline{v}_2] = [1,5]$ and $\psi_r = 2\pi/3$. 102

8.1 Conflict resolution for three aircraft: the *roundabout* maneuver. . . . 108
8.2 Airspace simulation tool, incorporating dynamic models of aircraft in an interactive environment. 109

Acknowledgements

My greatest thanks go to my thesis advisor, Professor Shankar Sastry, who is a superb teacher and an inspirational advisor. His enthusiasm, patience and attention to detail, knowledge and insight, and the care and respect he has for his students make it a pleasure to work with him.

I would like to thank Professor Alexandre Chorin for his support and advice, and for teaching one of the best math courses I've taken, Professor Tom Henzinger for introducing me to hybrid systems from a computer scientist's point of view, and Professor Pravin Varaiya for his valuable comments about my research and thesis. My special thanks go to my host at NASA Ames, Dr. George Meyer, whose insights about flight systems and nonlinear tracking have been invaluable to me. I would also like to thank Mary Jo Hoffman and the Guidance and Control Group at Honeywell Technology Center, where I was introduced to some very interesting flight management system problems. I am pleased to thank Professor Jamie Sethian and Dr. Ravi Malladi for their help in understanding level set algorithms, and Professor Lucien Polak for his advice about non-smooth sets. I would also like to thank Professor Adib Kanafani for discussions about air traffic, Professor Richard Montgomery for conversations about shocks and symplectic geometry, and Professor Michael Heymann for the many conversations about hybrid systems.

It is a pleasure to acknowledge the research collaborations with my colleagues at Berkeley. In particular, controller synthesis for hybrid systems is joint work with John Lygeros, and conflict resolution algorithms for aircraft is joint work with George Pappas, Jana Košecká, and Yi Ma. The dynamic simulation tool was developed with Cedric Ma. In addition, I would like to thank Datta Godbole, Lara Crawford, Dawn Tilbury, Linda Bushnell, Jeff Wendlandt, John-Morten Godhavn, Sepanta Sekhavat, John Koo, Bruno Sinopoli, and Magnus Egerstedt for helping to create such an enriching research environment in our lab at Berkeley. John Lygeros and John-Morten Godhavn deserve special thanks for carefully proofreading my thesis and suggesting several good changes. I would also like to thank Angela Schuett, Grace Chang, Heath Hoffman, and Alexa Brudy for helping to make Berkeley a great place to live.

I am happy to acknowledge the financial support of NASA, and to thank the Natural Sciences and Engineering Council of Canada for the 1967 Fellowship, and the Zonta Foundation for the Amelia Earhart Awards.

My final thanks go to my family. I would especially like to thank my parents for their constant support, encouragement, and interest in my graduate student career.

Chapter 1

Introduction

The introduction of advanced automation into manually operated systems has been extremely successful in increasing the performance and flexibility of such systems, as well as significantly reducing the workload of the human operator. Examples include the automation of mechanical assembly plants, of the telephone system, of the interconnected power grid, as well as transportation system automation such as controllers in high speed trains, automatic braking systems in automobiles, and avionics on board commercial jets. Accompanying this increase in automation is the necessity of ensuring that the automated system always performs as expected. This is especially crucial for *safety critical* systems: if a telephone switch crashes or a power grid node goes down, lives are usually not lost, yet if an error occurs in the automated avionics on board a commercial jet, the results could be disastrous.

Many of today's safety critical systems are growing at such a rate that will make manual operation of them extremely difficult if not impossible in the near future. The Air Traffic Control (ATC) system is an example of such a safety critical system. Air traffic in the United States alone is expected to grow by 5% annually for the next 15 years [1], and rates across the Pacific Rim are expected to increase by more than 15% a year. Even with today's traffic, ground holds and airborne delays in flights due to congestion in the skies have become so common that airlines automatically pad their flight times with built-in delay times. Aging air traffic control equipment certainly contributes to these delays: the plan view displays used by controllers to look at radar tracks and flight information are the very same that were installed in the early 1970's, and they fail regularly. The computer systems which calculate radar tracks and store flight plans were designed in the 1980's, using software code that was written in 1972. The introduction of new computers, display units, and communication technologies for air traffic controllers will help alleviate the problems caused by failing equipment, yet the Federal Aviation Administration (FAA) admits that any significant improvement will require that many of the basic practices of ATC be automated [2]. For example, today's airspace has a rigid route structure based on altitude and on ground-based navigational "fixes": current practice of air traffic controllers is to route aircraft along predefined paths connecting fixes, to manage the complexity of route planning for several aircraft at once. The rigid structure puts strict constraints on aircraft trajectories, which could otherwise follow wind-optimal or user preferred routes. Also, while a data link between aircraft and ground is being investigated as a replacement for the current voice communication over radio channels between pilot and controller, there is a limit to the amount of information processing that a controller can perform with this data. Studies in [2] indicate that, if there is no change to the structure of ATC, then by the year 2015 there could be a major accident every 7 to 10 days.

The result is a perceived need in the air traffic, airline, and avionics communities for a new *architecture*, which integrates new technologies for data storage, processing, communications, and display, into a safe and efficient air traffic management system. The airlines are proponents of a decentralized architecture featuring *free flight*, meaning that each aircraft plans and tracks its own dynamic trajectory with minimal interference from ATC [3]. Many people (air traffic controllers in particular) view this as a radical solution, but a recent study funded by NASA [4] suggests that distributing some of the control authority to each aircraft would help improve the efficiency of the system as a whole. In [5] we propose an architecture for a new air traffic management system along these lines, in which the aircraft's flight management system uses local sensory information from Global Positioning Systems, Inertial Navigation Systems, and broadcast communication with other aircraft to resolve local conflicts without requesting clearances from ATC. While the degree of decentralization and level of automation in a new air traffic management system are still under debate (since it is very difficult to estimate the increase in efficiency from distributing the control authority), the integrity of any automated functionality in a new air traffic management system depends on a *provably-safe* design, and a high confidence that the control actions won't fail.

In the past, high confidence has been achieved by operating the system well within its performance limits. Extensive testing has been used to validate operations, and any errors occurring from untested situations would be compensated for by this degree of "slack" in the system performance. We would like to maintain high confidence but operate the system much closer to its performance limits. In order to do this, we require accurate models of the system, procedures for verifying that the design is safe to within the accuracy of these models, and procedures for synthesizing control actions for the system, so that safety is maintained.

For about the past six years, researchers in the traditionally distinct fields of control theory and computer science verification have proposed models, and verification and controller synthesis techniques for complex, safety critical systems. The area of *hybrid systems* is loosely defined as the study of systems which involve the interaction of discrete event and continuous time dynamics, with the purpose of proving properties such as reachability and stability. The discrete event models naturally accommodate linguistic and qualitative information, and are used to model modes of operation of the system, such as the mode of flight of an aircraft, or the interaction and coordination between several aircraft. The continuous dynamics model the physical processes themselves, such as the continuous response of an aircraft to the forces of aileron and throttle.

One class of approaches to modeling and analysis of hybrid systems has been to extend techniques for finite state automata to include systems with simple continuous dynamics. These approaches generally use one of two analysis techniques: model checking, which verifies a system specification symbolically on all system trajectories, and deductive theorem proving, which proves a specification by induction on all system trajectories. Emphasis is placed on *computability* and *decidability*, or proving that the problem: *Does the system satisfy the specification?* can be solved in a finite number of steps. Models and decidability results have been obtained for timed automata [6], linear hybrid automata [7], and hybrid input/output automata [8]. Linear hybrid automata model or abstract the continuous dynamics by differential inclusions of the form $A\dot{x} \leq b$ and verify properties of the resulting abstracted system [9, 10]. While reachability and eventuality properties for timed automata have been shown to be decidable, the decidability results for linear hybrid automata are fairly narrow. For all but the simplest continuous linear dynamics (two-dimensional rectangular differential inclusions), reachability properties are semi-decidable at best, and in most cases undecidable. Methods for designing discrete controllers for timed and hybrid systems have been developed using this framework [11, 12], and computational tools have been developed for both model checking [13, 14], and theorem proving [15].

A second class of models and analysis techniques for hybrid systems has developed out of research in continuous state space and continuous time dynamical systems and control. The emphasis here has been on extending the standard modeling, reachability and stability analyses, and controller design techniques to capture the interaction between the continuous and discrete dynamics [16, 17, 18, 19, 20, 21]. Analysis and design techniques extend existing control techniques, such as stability theory [17], optimal control [17, 20, 21], and control of discrete event systems [22, 23], to hybrid systems. One area in which results have been hard to come by is the efficient *computation* of reachable sets for hybrid systems whose dynamics are nonlinear or are of order greater than one. Only recently, some attempts to directly approach this problem have been reported in the literature [24, 25].

Our approach to hybrid systems modeling incorporates accurate, nonlinear models of the continuous dynamics with models for discrete event dynamics. We include continuous and discrete input variables to model both parameters that the designer may control as well as disturbance parameters that the designer must control against. Using analysis based on traditional discrete and continuous optimal control techniques, and on two-person zero-sum game theory for automata and continuous dynamical systems, we derive the Hamilton-Jacobi partial differential equations whose solutions describe *exactly* the boundaries of reachable sets. Only then do we approximate: we use a clever numerical technique to solve this equation. These equations are the heart of our general controller synthesis technique for hybrid systems, in which we calculate feedback control laws for the continuous and discrete variables which guarantee that the hybrid system remains in the "safe subset" of the reachable set. While about 10 years ago such a method would have been prohibitively computationally expensive, advances in computational power and new fast methods for integrating PDEs have made such solutions feasible, even for real-time applications. The result is an analytic and numerical method for computing reachable sets and control laws for hybrid systems, which doesn't require a preprocessing step to approximate the dynamics. We have been successful in computing solutions to finite-time examples, but in our method thus far, we have not addressed considerations of decidability and computational complexity.

1.1 Overview

Chapter 2 presents an overview of the current air traffic system, and a discussion of some of the new technologies which are becoming available for more efficient navigation and communication. We propose an architecture for an air traffic management system which incorporates these technologies, and we introduce three problem examples which are developed throughout the dissertation: two examples in deriving safe collision avoidance maneuvers for aircraft, and one example in autopilot mode switching. A more detailed description of the proposed architecture can be found in [5]. The first example has been presented (in less detail) in [26], the second example is unpublished, and the third example is developed in part from the example in [27] and [28]. Our motivation for this work arose out of attempting to verify the safety of a class of conflict resolution maneuvers for aircraft, in [29]. Related previous work is that of [30], in which game theoretic methods were used to prove safety of a set of maneuvers for Automated Highway Systems.

The nonlinear hybrid system model presented in Chapter 3 is based on that of [20], further developed in [26], [31]. We present a model for a controller and we illustrate how the three example problems are modeled as nonlinear hybrid systems.

In Chapter 4 we present algorithms for evolving boundaries of reachable sets for discrete and continuous systems. The discrete algorithm was first presented by Büchi and Landweber in the late 1960's [32], our presentation follows that of [11]. The representation of the discrete algorithm in terms of a "discrete Hamilton-Jacobi equation" is new. The continuous algorithm is classical, its derivation can be found in most books on optimal control and dynamic games (see [33, 34, 35, 36]). The notion of control invariance for continuous systems is described in [37], however its development in our setting is novel.

Chapter 5 presents our algorithm for synthesizing reachable sets and control laws for safety specifications of hybrid systems. The material presented in this chapter is developed from the presentations in [26], [28], and [31].

In Chapter 6 we apply the synthesis algorithm of Chapter 5 to the three example problems. In Chapter 7 we discuss the use of level set methods [38] as a numerical implementation of our algorithm, and Chapter 8 collects a set of future research directions.

1.2 Notation

Let PC^0 denote the space of piecewise continuous functions over $\mathbb{R}$, and PC^1 the space of piecewise differentiable functions over $\mathbb{R}$. Let Q be a finite set of discrete state variables, then $|Q|$ represents the cardinality of Q, and Q^ω represents infinite sequences of elements in Q. Let X be a continuous state space of dimension n, and let $G \subseteq X$. Then G° is the interior of G, ∂G is the boundary of G, and G^c is the complement of G: $G^c = X \backslash G$. We summarize the notation used for discrete and continuous systems in the following table.

Entity	**Discrete**	**Continuous**
States	Q	X
Input Sets	$\Sigma_1 \times \Sigma_2$	$U \times D$
Input Spaces	$\Sigma_1^\omega \times \Sigma_2^\omega$	$\mathcal{U} \times \mathcal{D} \subset PC^0 \times PC^0$
Transitions	$\delta : Q \times \Sigma_1 \times \Sigma_2 \rightarrow 2^Q$	$f : X \times \mathcal{U} \times \mathcal{D} \rightarrow TX$
Trajectories	$(q[\cdot], \sigma_1[\cdot], \sigma_2[\cdot]) \in Q^\omega \times \Sigma_1^\omega \times \Sigma_2^\omega$: $q[i+1] \in \delta(q[i], \sigma_1[i], \sigma_2[i])$	$(x(\cdot), u(\cdot), d(\cdot)) \in PC^1 \times \mathcal{U} \times \mathcal{D}$ $\forall \tau, \dot{x}(\tau) = f(x(\tau), u(\tau), d(\tau))$
Specification	$\Box F$ $(\forall i, x(i) \in F)$, $F \subseteq Q$	$\forall \tau, x(\tau) \in F$, $F \subseteq Q$

Chapter 2

Algorithms for Distributed Air Traffic Management

We first describe the Air Traffic Control (ATC) system used in the United States today, emphasizing the structure of the airspace, and the methods used by air traffic controllers to direct traffic, and by pilots to follow these directions. We then describe some of the technologies, both proposed and under development, to enable a change towards a more efficient system. [2], [39], [40], [41], and [42] provide excellent overviews of the current ATC and some of the new technologies available. We describe a proposed architecture for new Air Traffic Management (ATM) which would move much of the current ATC functionality on board each aircraft. We conclude with three examples representing two crucial problems to be solved in *any* proposed ATM: the problem of conflict resolution between aircraft, and that of consistent and safe flight mode switching in an aircraft's autopilot.

2.1 Overview of the Current System

ATC has its earliest roots in the 1920's, when local airline dispatchers would direct pilots to fly flight plans marked by rudimentary markers on the ground. In 1935, the first inter-airline ATC was organized in the Chicago-Cleveland-Newark corridor, which was taken over in 1937 when the responsibility for ATC was transferred from the airlines to the federal government. The advances in radar and radio technology in the ensuing decades allowed closer surveillance of aircraft, and the growth of the aircraft jet engine industry made it possible for the average aircraft to fly at much faster speeds. The system of aircraft, pilots, and controllers evolved into what today is known as the *National Airspace System*, or NAS, and its management is referred to as *Air Traffic Management*, or ATM.

ATM in the United States is currently organized hierarchically with a single *Air Traffic Control System Command Center (ATCSCC)* supervising the overall traffic flow. This is supported by 22 *Air Route Traffic Control Centers (ARTCCs)* organized by geographical region, which control the airspace up to 60,000 feet. Each Center is sub-divided into about 20 sectors, with at least one air traffic controller responsible for each sector. Coastal ARTCCs have jurisdiction over oceanic airspace: the Oakland Center in California, for example, controls a large part of the airspace above the Pacific Ocean. Within the Center airspace, the low traffic density region away from airports is known as the *en route airspace* and is under jurisdiction of the ARTCC. The high traffic density regions around urban airports are delegated to *Terminal Radar Approach Control (TRACON)* facilities. The TRACONs generally control this airspace up to 15,000 feet. There are more than 150 TRACONS in the United States: one may serve several airports. For example, the Bay Area TRACON includes the San Francisco, Oakland, and San José airports along with smaller airfields at Moffett Field, San Carlos, and Fremont. The regions of airspace directly around an airport as well as the runway and ground operations at the airport are controlled by the familiar *Air Traffic Control Towers.* There are roughly 17,000 landing facilities in the United States serving nearly 220,000 aircraft. Of these there are about 6,000 commercial aircraft: the number of commercially used airstrips is roughly 400 (these are all equipped with control towers).

ATC currently directs air traffic along predefined jet ways, or "freeways in the sky", which are straight line segments connecting a system of beacons (non-directional beacons (NDBs), very high frequency omni-range receivers (VORs), and distance measuring equipment (DME)). These beacons are used by pilots (and autopilots) as navigational aids, to update and correct the current position information provided by the inertial navigation systems (INS) on board each aircraft. Surveillance is performed by ATC through the use of radar: a primary radar system which processes reflected signals from the aircraft skin, and a secondary radar system, which triggers a transmitter in the aircraft to automatically emit an identification signal. The range of the radars depends on the type of airspace being served: in the en route airspace the long-range Air Route Surveillance Radar (ARSR) is used, while in the TRACON the shorter range Automated Radar Terminal System (ARTS) is used. The accuracy of the radars, and their slow (12 second) update rates, contribute to the FAA standards for aircraft separation, which are 5 nautical miles horizontal separation, 1000 feet (2000 feet above 29,000 feet) vertical separation in the Center airspace, and 3 nautical miles horizontal separation, 1000 feet vertical separation in the TRACON. Each ATC facility is equipped with a computer system which takes the radar signals as input and provides a very limited amount of flight data processing, including a rudimentary conflict alert function. This information is displayed to controllers in two-dimensions on the black and green plan view displays (PVDs). Controllers issue directives to pilots using two-way voice (radio) channels. Figure 2.1 shows a flight map (horizontal profile) of a portion of the San Francisco Bay Area: the circular "dials" indicate VOR beacons (including airports), the boundary of the TRACON is shown as well as a part of the Oakland Center airspace.

Prior to a commercial aircraft's departure, the airline files a flight plan with ATC, which indicates information about the aircraft and its desired trajectory from origin to destination airports in the form of a very coarse sequence of *way points.* ATC modifies the flight plan according to constraints of the NAS and other aircraft, and issues a clearance to the pilot. After take-off, the control of the aircraft is passed through the Tower, TRACON, and possibly several Center facilities until the destination TRACON is reached. Information about the part of the filed flight plan relevant to his sector is passed via the computer system to each TRACON and Center controller, and the information is printed out on "flight strips" (Figure 2.2) which indicate the planned position of the aircraft at several points along the route.

The main goal of ATC is to maintain safe separation between aircraft while guiding them to their destinations. However, the tight control that it has over the motion of

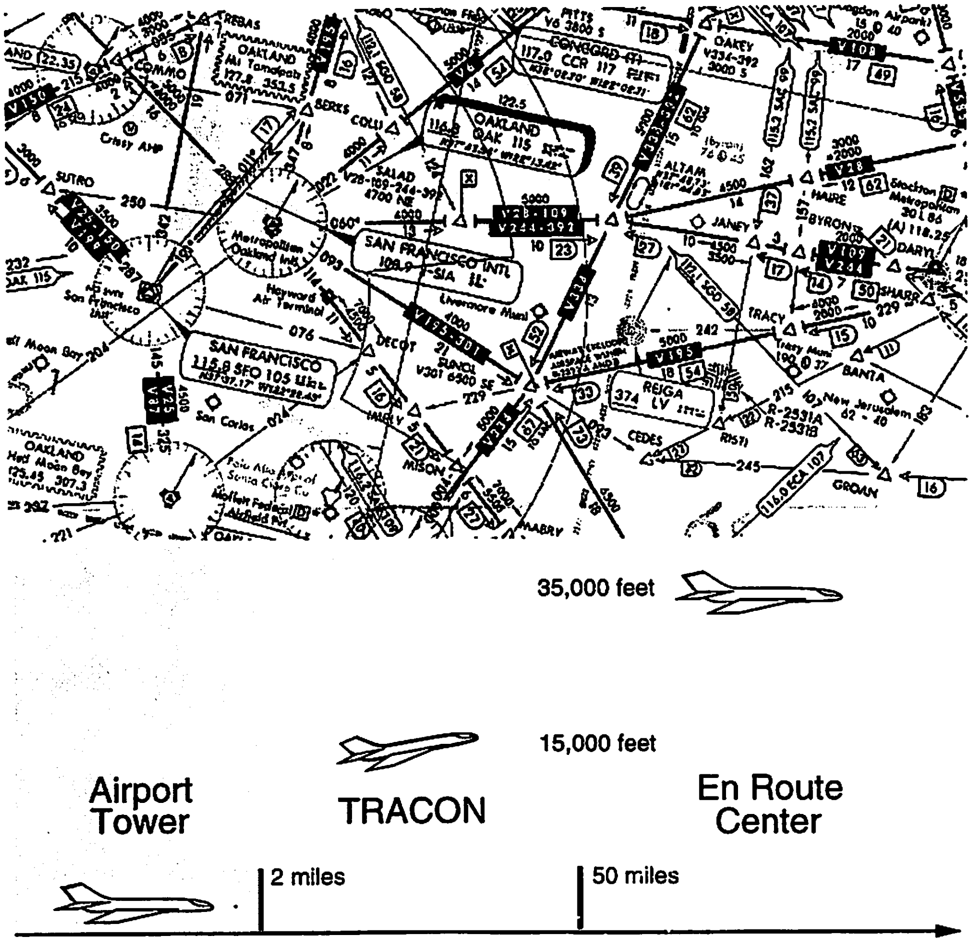
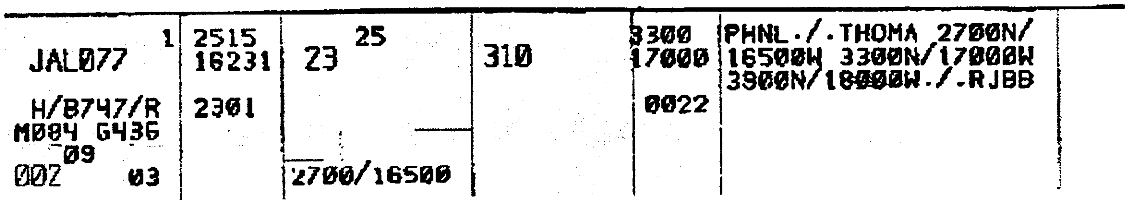


Figure 2.1: Bay Area airports, TRACON. and part of Oakland Center.



1 JALØ77 H/B747/R MØØ4 G436 Ø9 ØØ2 Ø3	2515 16231 23Ø1	25 23 27ØØ/165ØØ	31Ø	33ØØ 17ØØØ ØØ22	PHNL./.THOMA 27ØØN/ 165ØØW 33ØØN/17ØØØW 35ØØN/18ØØØW./.RJBB

Figure 2.2: A flight strip from the Oakland Center.

every aircraft in the system frequently causes bottlenecks to develop. Uncertainties in the positions, velocities, and wind speeds, as well as the inability of a single controller to handle large numbers of aircraft at once lead to overly conservative controller actions and procedures to maintain safety. An example of this is the methods used by air traffic controllers to predict and avoid conflicts between aircraft. If a controller predicts that the separation between two aircraft will become less than the regulatory separation, the controller will issue a directive to one or both of the pilots to alter their paths, speed, or both. Often the resolution is not needed, and usually it is too severe. Also, the so-called "user preferred routes" (shorter or lower fuel consumption routes that take advantage of tailwinds) are disallowed because of the requirement to use prescribed jet ways.

Airspace capacity is the maximum number of operations that can be processed per unit time in a certain volume of the airspace given a continuous demand [41]. In this definition a distinction is made between different modes of operation, such as *level flight at fixed heading*, *climbing*, *descending*, and *changes in heading.* Airspace capacity is a function of aircraft count, activity mix, protocols for conflict detection and resolution, and FAA regulations. It is our contention that this capacity can be increased by better protocols which do not compromise safety.

An area of current activity is the development of decision support tools for air traffic controllers. One such tool is the *Center-TRACON Automation System (CTAS)* [43] which is currently under development at NASA Ames, and under field test at Denver and Dallas-Fort Worth airports. CTAS is software code which runs on computer workstations next to the air traffic controller; it uses radar data, current weather information, aircraft flight plans and simplified dynamic aircraft models to predict the aircraft trajectories, alert the controllers about potential conflicts, and provide advisories to the controller about landing sequences.

We conclude this section with a short introduction to the automated flight management system (FMS) on board commercial jets, such as those of the Boeing B777 and the Airbus A320. In contrast to the "low technology" ATC operation, modern FMSs are highly automated systems which assist the pilot in constructing and flying four-dimensional trajectories, as well as altering these trajectories on line in response to ATC directives. An FMS typically controls the throttle input and the vertical and lateral trajectories of the aircraft to automatically perform such functions as: acquiring a specified altitude and then leveling (ALT ACQ), holding a specified altitude (ALT HLD), acquiring a specified vertical climb or descend rate (V/S), automatic vertical or lateral navigation between specified way points (VNAV, LNAV), or holding a specified throttle value (THR HLD). The combination of these throttle-vertical-lateral modes is referred to as the *flight mode* of the aircraft. A typical autopilot has several hundred flight modes (see [44] for a discussion of the Boeing B737 flight modes). It is interesting to note that these flight modes were designed to automate the way pilots fly aircraft manually: by controlling the lateral and vertical states of the aircraft to set points for fixed periods of time, pilots simplify the complex task of flying an aircraft. Figure 2.3 illustrates two screens in the cockpit of such an FMS-equipped jet: a horizontal profile showing the current position of the aircraft as it follows an approach route, marked by way points, into the Los Angeles airport, and an "artificial horizon" which shows the current pitch and roll angles of the aircraft, the airspeed and altitude, and the current flight mode. Prior to take-off, the pilot can enter the approved flight plan into the FMS computer on board the aircraft, and during flight can choose the desired level of automation. For example, if the pilot selects the LNAV or VNAV mode, the FMS determines the altitudes, speeds, pitch, roll, and throttle values to navigate between way points; if the HDG SEL or ALT ACQ modes are chosen, the pilot chooses the desired heading and altitude values.

While the introduction of automation to on-board avionics has resulted in increased performance of commercial autopilots, the need for automation designs which guarantee safe operation of the aircraft has become paramount. Currently, designers and manufacturers of FMSs "verify" the safety of the systems by simulating them for long periods of time with various initial conditions and inputs. This procedure is not adequate, since trajectories to unsafe states may be overlooked. "Automation surprises" have been extensively studied [44, 45, 46] *after* the unsafe situation occurs, and "band-aids" are added to the FMS design to ensure the same problem does not occur again. One of the goals of this dissertation is to present a system design method, in which safety properties are *a priori* verified in the design phase, so that

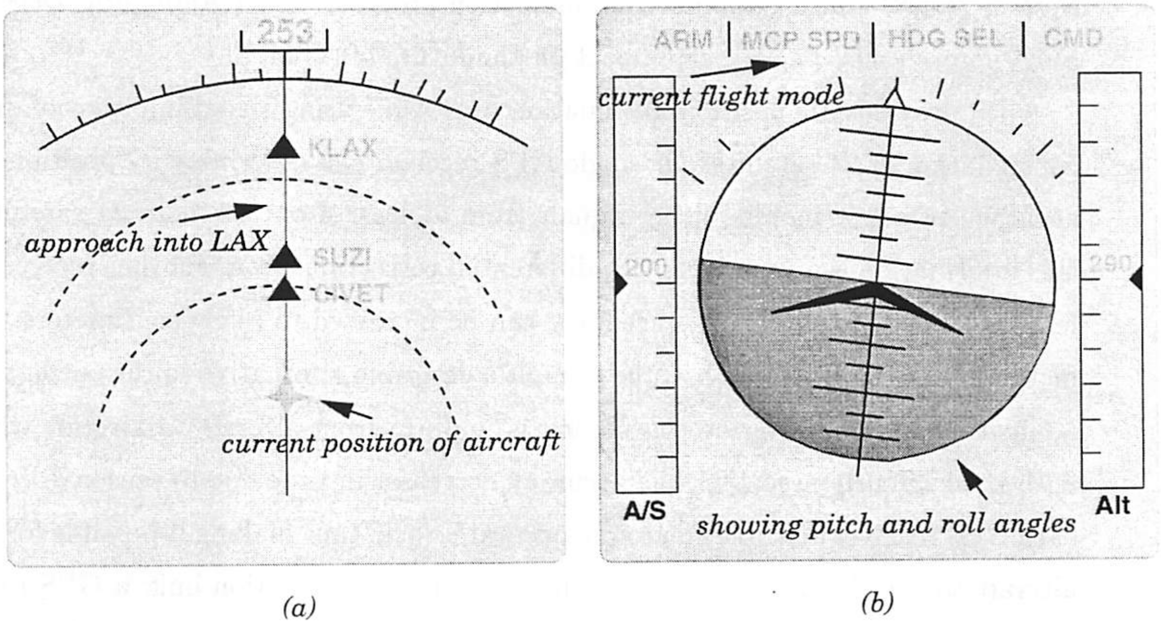

253
KLAX
approach into LAX
SUZI
GIVET
current position of aircraft
ARM | MCP SPD | HDG SEL | CMD
current flight mode
200
290
A/S
Alt
showing pitch and roll angles
(a)
(b)

Figure 2.3: Two screens in a typical glass cockpit: (a) a horizontal profile of way points (into Los Angeles airport); (b) an "artificial horizon" showing the current pitch and roll angles of the aircraft, the airspeed and altitude, and the current flight mode. The first three columns in the flight mode are the throttle-vertical-lateral modes, the fourth is the autopilot mode. ARM means "waiting for the throttle to reach required value", MCP SPD means "speed is controlled to the entry in the mode control panel", HDG SEL means "heading is controlled to the entry in the mode control panel", CMD means "pilot has command over pitch and roll values".

no automation surprises occur.

2.2 Technologies to Enable Change

Several new technologies are under development and certification, and are fueling a change in the structure of ATM. In this section we discuss the *Global Positioning System (GPS)* and a datalink communication protocol called *Automatic Dependent Surveillance (ADS)* and their impact on the future of ATM.

GPS provides 3D position information worldwide using signal information from a constellation of 24 satellites. A single GPS receiver can determine its position to an accuracy of a few meters, using signals from at least 4 out of these 24 satellites; if this information is augmented with differential corrections from another receiver (differential GPS or DGPS), this accuracy can be increased to a few centimeters. Many factors make the use of GPS in the cockpit a desirable alternative to the current ATM navigation methods [42]: the accuracy is uniform from aircraft to aircraft whereas with the currently used INS, the accuracy decreases in time due to sensor drift rates; each GPS receiver acts like an atomic-accurate clock, thus making it possible for many aircraft to coordinate among each other over a communication link; a GPS receiver is much cheaper than an INS system, and orders of magnitude cheaper than a VOR beacon. One disadvantage of relying on GPS position information is that the satellite signal may be lost temporarily if the GPS receiver is obscured from the direct path of the signal. Current studies [47] suggest an integrated use of both INS and GPS, in which the accurate position information from GPS is used to continually correct the INS position.

ADS is a communication protocol by which aircraft would transmit over digital satellite communication their GPS position information, velocity, as well as information about their intended trajectory, to the ground ATC. ADS-B (for broadcast) is a protocol for broadcasting this information to neighboring aircraft [3]. Its major advantage over the current ATM surveillance methods is its ability to provide very accurate information for trajectory prediction, without relying on the radar system. Two immediate benefits of such a communication link are a huge improvement in surveillance over oceanic airspace, which is not covered by radar, and the possibility of reducing the separation standards between aircraft in all airspace.

Despite the short-term benefits that these new technologies provide, the real long-term benefits will depend on how the airspace system and its management evolve around such new technologies. Aviation in the next century will, more than ever before, be based on *systems related issues*: the need to integrate highly automated aircraft, advanced navigation and surveillance technology, sophisticated computation, and user preferences, into a system which meets the demands resulting from skyrocketing growth in air travel, without compromising the standards of such a safety critical system. The aviation community has accepted that today's controller-based system will not meet these requirements, and a new system structure is needed. A concept called *free flight* [48] has been proposed in recent years. Free flight is loosely defined to mean that pilots are allowed to choose their own routes, altitude and speed, and would share the tasks of navigation, surveillance, aircraft separation, and weather prediction, with ground-based controllers. User preference would be restricted only in congested or special use (military) airspace.

In the following section, we present an architecture for a "next generation" air traffic management system [5], which incorporates user preference and moves some of the current ATC functionality on board the aircraft. Our purpose in presenting this architecture is to provide a framework for the examples presented in this dissertation: the modeling, verification, and controller synthesis techniques which are at the heart of this dissertation are general, and may be applied to any ATM architecture.

2.3 Proposed Architecture

We assume, as in the current ATC practice, that user (airline) preferences are incorporated in the initial flight planning stage, in which the airline and ATC can "negotiate" the sequence of way points that comprises the nominal flight plan for the aircraft. This nominal plan is designed to be time-optimal and conflict-free, within the constraints of the schedules of the other aircraft in the system. Once a commercial aircraft is airborne and outside of the TRACON, it starts to play an active role in its

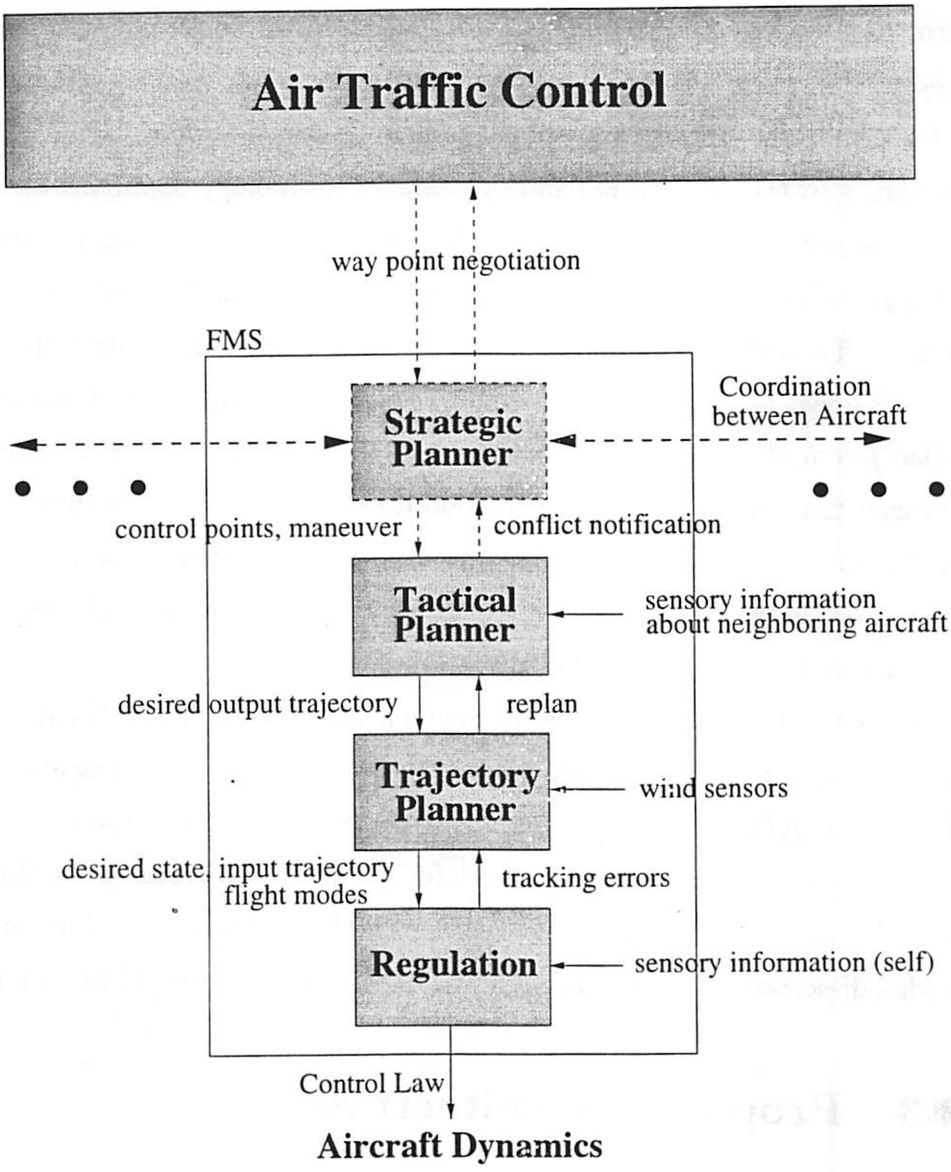


Figure 2.4: Proposed framework for on-board planning and control.

own navigation and surveillance. As shown in Figure 2.4, the flight management system on board each aircraft may be interpreted as a hierarchical system, which takes as input the nominal flight plan from ATC, information about neighboring aircraft, about its own aircraft state, and about wind and weather, and produces a conflict-free full state and input trajectory [49]. The *strategic planner* interpolates the nominal trajectory's way points with a set of *control points* which delineate the constant control segments between way points. The *tactical planner* refines the strategic plan by joining the control points with a smooth output trajectory. The *trajectory planner* uses a detailed dynamic model of the aircraft, sensory input about the wind's magnitude and direction, and the tactical plan, to design a full state and input trajectory for the aircraft, and the sequence of *flight modes* necessary to execute the dynamic plan. The *regulation layer* is a simple, fast control scheme, which closes the loop on the dynamics of the aircraft. Tracking errors are passed back to the trajectory planner, to facilitate replanning if necessary.

Often, as with the current ATM, bad weather, high winds, or schedule delays which cause conflicts with other aircraft may force the aircraft to deviate from the nominal route. The strategic planner on board the aircraft has the ability to coordinate with neighboring aircraft to determine a sequence of maneuvers which will result in conflict-free trajectories. We propose a conflict resolution methodology based on a set of *protocols*, easily understood by pilots and easily programmed into an FMS, to allow aircraft to coordinate among each other to avoid conflict. Each strategic planner then commands its own tactical planner to follow these maneuvers.

2.4 Motivating Examples

We now concentrate on two systems in an ATM architecture: a *provably-safe* algorithm for resolving trajectory conflicts between aircraft, and a *provably-safe* algorithm for a single aircraft to switch between different flight modes. The notion of "safety" in each case is crucial:

Definition 1 (Safety) *A system is* safe *if its state trajectories always remain within a* safe subset *of the state space.*

In the conflict resolution problem, the system is safe if the aircraft always maintain *minimum separation* with each other. In the flight mode switching problem, system safety means that the state of the aircraft remains within minimum and maximum bounds imposed on its velocities, angles etc. so that the aircraft doesn't stall, causing it to plunge out of the sky. The latter is referred to as *aerodynamic envelope protection.* We present these systems through examples, which are introduced in this section and developed throughout the dissertation.

2.4.1 Conflict Resolution for Aircraft

Consider a system of aircraft, each navigating using a combination of GPS and INS, and each providing surveillance information through an ADS link with ATC, and an ADS-B link with neighboring aircraft. Each aircraft is surrounded by two virtual *cylinders*, the *protected zone* and *alert zone* shown in Figure 2.5 as a top view. The radius and height of the protected zone depends on the FAA separation standards (2.5 nautical miles by 1000 feet in Center, 1.5 nautical miles by 1000 feet in TRACON). The size and shape of the alert zone depends on various factors including airspeed, altitude, accuracy of sensing equipment, traffic situation, aircraft performance and average human and system response times: it is shown as an ellipsoid in Figure 2.5. A *conflict* or loss of separation between aircraft occurs when their protected zones overlap. The system of aircraft is defined to be *safe* if the aircraft trajectories are such that their protected zones never overlap.

We propose a conflict resolution algorithm which may be executed either on board each aircraft, as suggested by the architecture of the previous section, or in an ATC TRACON or ARTCC facility on the ground. The algorithm has access to the state and intent information of the other aircraft involved in the conflict, through the GPS/INS system linked to the ADS/ADS-B communication link, to information about the aerodynamics and performance characteristics of the other aircraft, and to information about the constraints imposed by the global traffic flow (see Figure

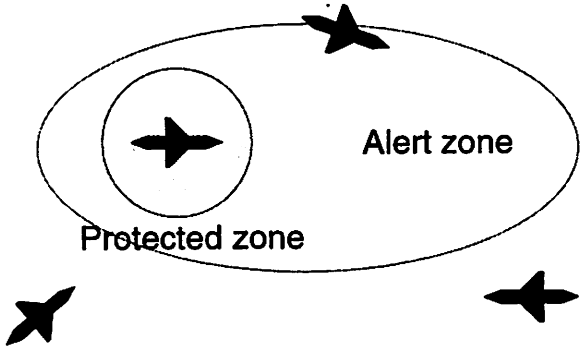


Figure 2.5: Aircraft Zones.

2.6). When aircraft enter the alert zone of another aircraft, an alert is issued to ATC as well as to the FMS of each involved aircraft, and depending on the relative configurations (positions, velocities) of the aircraft, a maneuver is generated which resolves the conflict. From a database of flight modes, such as segments of constant heading, of constant bank angle, of constant airspeed, the conflict resolution algorithm synthesizes the *parameters of the maneuver*, such as the proper sequencing of these modes, the numerical values associated to each segment (heading angle, bank angle, airspeed), and the conditions for switching between flight modes. The result is a maneuver, proven to be safe within the limits of the models used, which is a familiar sequence of commands easily executable by the FMSs. The resulting maneuvers may be viewed as *protocols*, or "rules of the road".

Conflict prediction and resolution have been sources of interest for the air traffic, control, and computational geometry communities in recent years. Spatial and temporal approaches, such as [50, 51], calculate the four dimensional coordinates of a possible conflict. Probabilistic approaches, such as [52, 53] assume stochastic uncertainty in the measured information and determine the probability of collision. A feature of our algorithm is that it is *provably safe* to within the limits of our models. We account for uncertainty or incompleteness in any of the information: as the bounds on the uncertainties increase, so does the conservatism of the resulting maneuver.

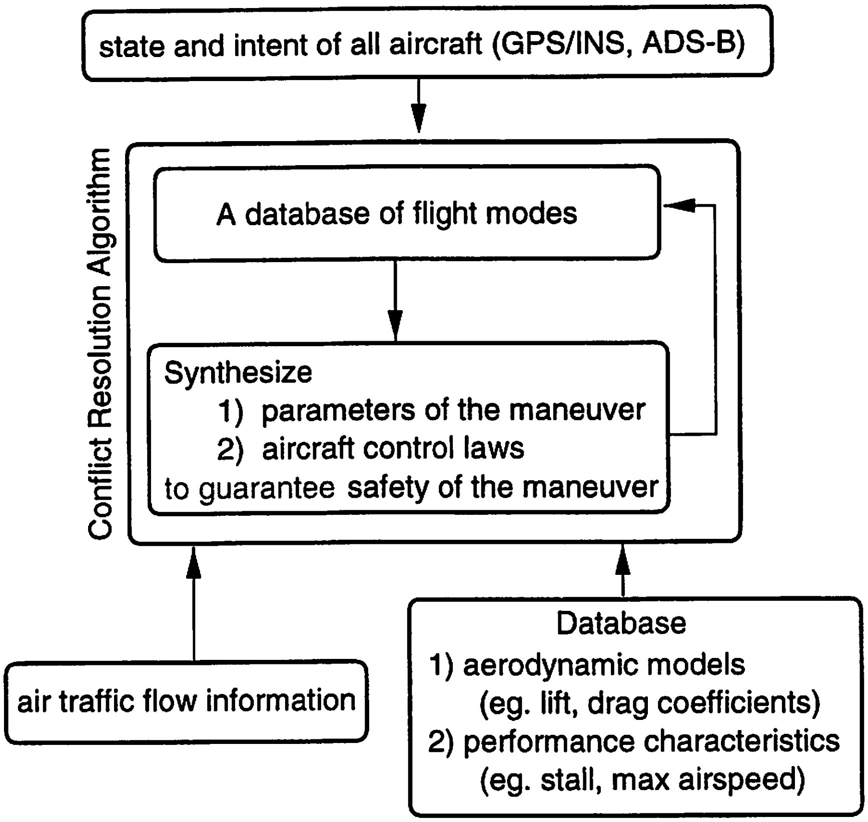

state and intent of all aircraft (GPS/INS, ADS-B)
Conflict Resolution Algorithm
A database of flight modes
Synthesize
1) parameters of the maneuver
2) aircraft control laws
to guarantee safety of the maneuver
air traffic flow information
Database
1) aerodynamic models
(eg. lift, drag coefficients)
2) performance characteristics
(eg. stall, max airspeed)

Figure 2.6: Conflict Resolution Algorithm.

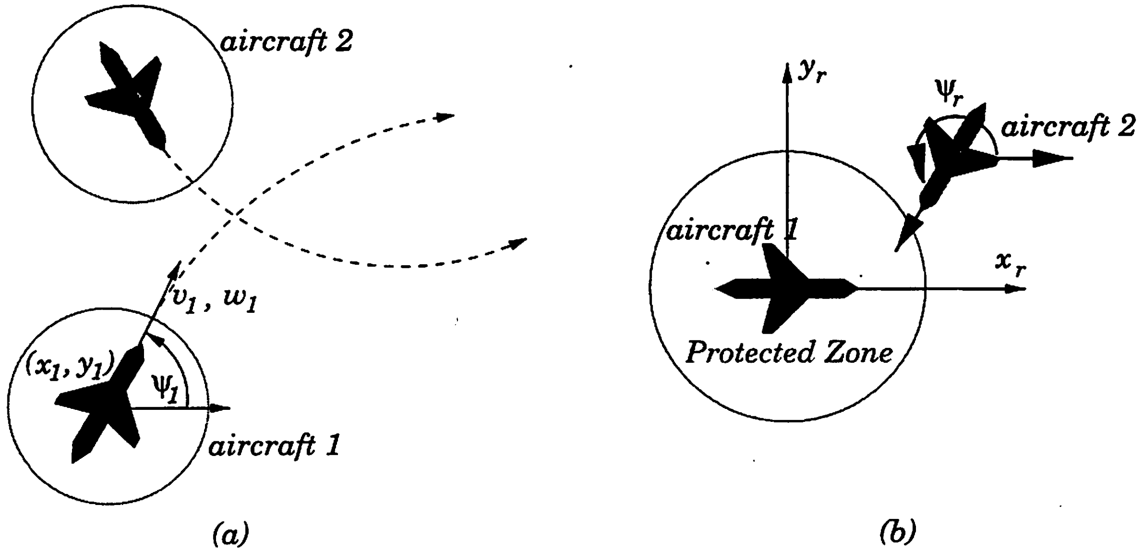


Figure 2.7: (a) Two aircraft in a conflict scenario; (b) The relative configuration, showing the relative protected zone.

Conflict Resolution for Two Aircraft in $SE(2)$

We present as motivating example a model for the kinematic motions of two aircraft at a fixed altitude, as shown in Figure 2.7(a). The position and heading of each aircraft is described by an element of the Lie group G of rigid motions in $\mathbb{R}^2$, called $SE(2)$ for the *Special Euclidean* group in $\mathbb{R}^2$. Let $g_i \in G$ denote the configuration of aircraft i:

$$g_i = \begin{bmatrix} \cos\psi_i & -\sin\psi_i & x_i \\ \sin\psi_i & \cos\psi_i & y_i \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

where (x_i, y_i) denotes the position of aircraft i and ψ_i is its heading. The motion of the aircraft may be modeled as a left-invariant vector field on G:

$$\dot{g}_i = g_i X_i \tag{2.2}$$

where $X_i \in \mathcal{G}$, the Lie algebra associated with the Lie group G. The Lie algebra in this case is $\mathcal{G} = se(2)$, with $X_i \in se(2)$ represented as

$$X_i = \begin{bmatrix} 0 & -\omega_i & v_i \\ \omega_i & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.3}$$

where ω_i is the aircraft's angular velocity, and v_i is its airspeed.

A coordinate change is performed to place the identity element of the Lie group G on aircraft 1, as shown in Figure 2.7(b). Let $g_r \in G$ denote the relative configuration of aircraft 2 with respect to aircraft 1. Then

$$g_2 = g_1 g_r \Rightarrow g_r = g_1^{-1} g_2 \tag{2.4}$$

In local coordinates, the coordinate transformation is expressed as

$$\begin{aligned} \begin{bmatrix} x_r \\ y_r \end{bmatrix} &= R(-\psi_1) \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} = \begin{bmatrix} \cos(-\psi_1) & -\sin(-\psi_1) \\ \sin(-\psi_1) & \cos(-\psi_1) \end{bmatrix} \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} && (2.5) \\ \psi_r &= \psi_2 - \psi_1 && (2.6) \end{aligned}$$

and g_r is given by

$$g_r = \begin{bmatrix} \cos\psi_r & -\sin\psi_r & x_r \\ \sin\psi_r & \cos\psi_r & y_r \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

in which $(x_r, y_r, \psi_r) \in \mathbb{R}^2 \times [-\pi, \pi)$ represent the relative position and orientation of aircraft 2 with respect to aircraft 1. Differentiating g_r, we obtain

$$\dot{g}_r = g_r X_2 - X_1 g_r \tag{2.8}$$

which may be written in (x_r, y_r, ψ_r) coordinates as

$$\begin{aligned} \dot{x}_r &= -v_1 + v_2 \cos\psi_r + \omega_1 y_r \\ \dot{y}_r &= v_2 \sin\psi_r - \omega_1 x_r \\ \dot{\psi}_r &= \omega_2 - \omega_1 \end{aligned} \tag{2.9}$$

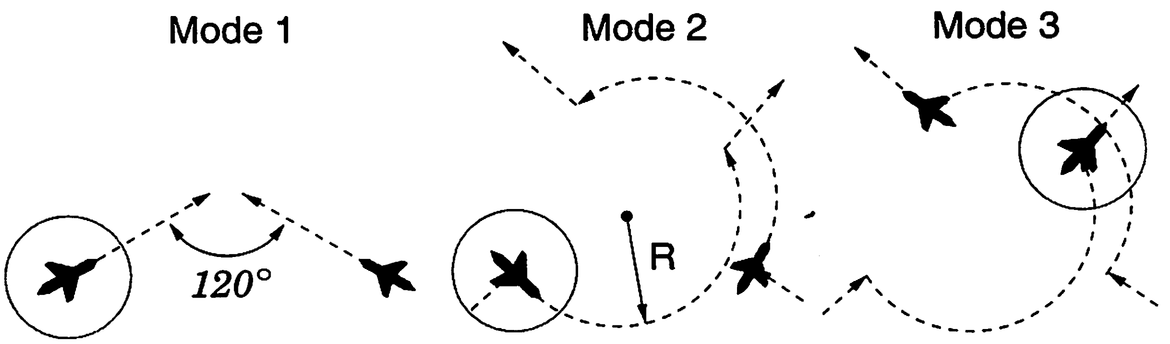


Figure 2.8: Two aircraft in three modes of operation: in modes 1 and 3 the aircraft follow a straight course and in mode 2 the aircraft follow a half circle. The initial relative heading (120°) is preserved throughout.

The protected zone of aircraft 2 may be translated to the origin as shown in Figure 2.7(b).

In order to maintain safe separation, the relative position (x_r, y_r) must remain outside of the protected zone, defined as

$$\{(x_r, y_r, \psi_r) : x_r^2 + y_r^2 < 5^2\} \tag{2.10}$$

for the lateral 5 nautical mile separation in Center airspace.

The flight modes for this system of two aircraft are based on the linear and angular velocities of the aircraft. We consider two possibilities: $\omega_i = 0$, meaning that aircraft i follows a straight line, and $\omega_i \neq 0$, but is a constant, meaning that aircraft i follows an arc of a circle. Thus the *database of maneuvers* for the example in this section are straight line segments of varying length and associated varying airspeed, and arcs of circles of varying length and radii. These maneuvers approximate closely the behavior of pilots flying aircraft: straight line segments (constant heading) and arcs of circles (constant bank angle) are easy to fly both manually and on autopilot.

Three-Mode Example

Consider a scenario in which there are three modes of operation: a *cruise* mode in which both aircraft follow a straight path; an *avoid* mode in which both aircraft follow a circular arc path; and a second *cruise* mode in which the aircraft return to the straight path. The protocol of the maneuver is that as soon as the aircraft

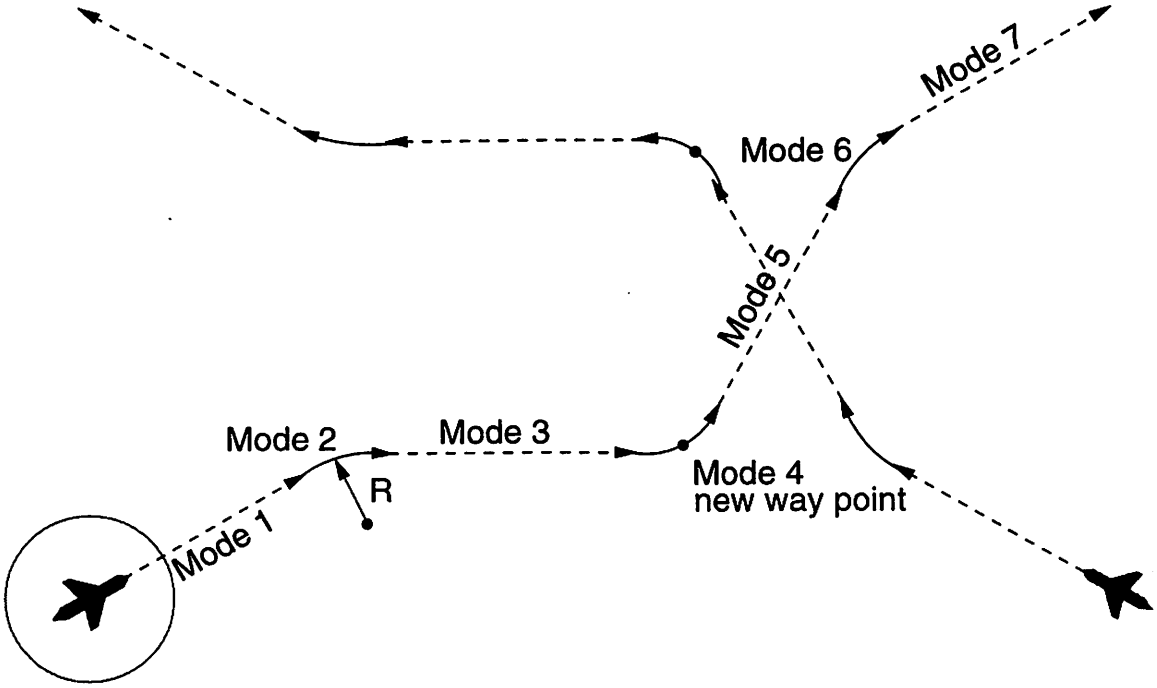


Figure 2.9: Two aircraft in seven modes of operation: in modes 1, 3, 5, and 7 the aircraft follow a straight course and in modes 2, 4, and 6 the aircraft follow arcs of circles. Again, the initial relative heading (120°) is preserved throughout.

are within a certain distance of each other, each aircraft turns 90° to its right and follows a half circle. Once the half circle is complete, each aircraft returns to its original heading and continues on its straight path (Figure 2.8). In each mode, the continuous dynamics may be expressed in terms of the *relative motion* of the two aircraft (2.9). In the cruise mode, $\omega_i = 0$ for $i = 1, 2$ and in the avoid mode, $\omega_i = 1$ for $i = 1, 2$. We assume that both aircraft switch modes simultaneously, so that the relative orientation ψ_r is constant. This assumption simply allows us to display the state space in two dimensions, making the results easier to present.

Problem statement: Generate the relative distance between aircraft at which the aircraft may switch safely from mode 1 to mode 2, and the minimum turning radius R in mode 2, to ensure that the 5 nautical mile separation is maintained.

Seven-Mode Example

The previous example is somewhat academic (aircraft cannot change heading instantaneously), yet (as so often happens with academic examples) its simplicity makes it a good vehicle to illustrate the controller synthesis methods of this dissertation. To show that our methods are not confined to academia and may indeed be applied to real-world situations, we present a "seven-mode example" which much better approximates current ATC practice. The example is illustrated in Figure 2.9. When two aircraft come within a certain distance of each other, each aircraft starts to turn to its right, following a trajectory which is a sequence of arcs of circles of fixed radii, and straight lines. As in the previous example, we assume that both aircraft switch modes simultaneously. We also assume that the angles of the avoid maneuver are fixed, so that the straight path of mode 3 is at a $-45°$ angle to the straight path of mode 1, and that of mode 5 is at a $45°$ to that of mode 1. Also, the length of each arc is fixed at a prespecified value, and the lengths of the segments in modes 3 and 5 are equal to each other, but unspecified.

Problem statement: Given some uncertainty in the actions of the aircraft, generate the relative distance between aircraft at which the aircraft may switch safely from mode 1 to mode 2, and the minimum lengths of the segments in modes 3 and 5, to ensure that the 5 nautical mile separation is maintained.

2.4.2 Flight Mode Switching and Envelope Protection

We would like to design a *safe* automatic flight mode switching algorithm for an FMS which interacts with both the dynamical system consisting of the aircraft and autopilot as well as with Air Traffic Control (ATC), and guides the aircraft safely through a sequence of waypoints in the presence of disturbances. As overviewed in the previous section, the FMS accepts a high level trajectory plan from ATC and constructs a sequence of elementary flight modes to effect a conflict-free version of this plan. The trajectory planner in the FMS is responsible for the safe sequencing of these modes. In this case, the aircraft is safe if its state trajectory remains within the *aerodynamic flight envelope*, which is a subset of the state space delineated by

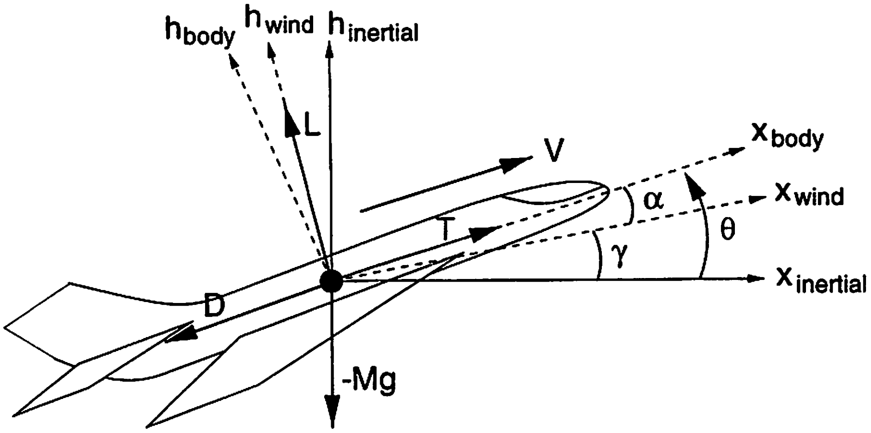


Figure 2.10: A planar aircraft in flight with attached axes about its center of mass.

allowable limits on the airspeed, vertical velocity, flight path angle, and altitude. Our algorithm must ensure that the FMS will attempt to select and activate only those flight modes for which the state trajectory is guaranteed to stay within the envelope. This is known as *envelope protection.*

Mode Switching for the Longitudinal Axis Dynamics of a CTOL Aircraft

The example is inspired by the work of [54], in which the flight modes for the airspeed and flight path angle dynamics of an aircraft are derived.

We consider a nonlinear model of the longitudinal axis dynamics of a conventional take-off and landing (CTOL) aircraft in normal aerodynamic flight in still air [55, 56], shown in Figure 2.10. The horizontal and vertical axes are respectively the $(x_{inertial}, h_{inertial})$ (denoted x, h) axes and the *pitch angle* θ is the angle made by the aircraft body axis, x_{body} with the x axis. The *flight path angle* γ and the *angle of attack* α are defined as: $\gamma = \tan^{-1}(\frac{\dot{h}}{\dot{x}})$, $\alpha = \theta - \gamma$. Expressions for the lift (L) and drag (D) forces are given by

$$\begin{aligned} L &= a_L(\dot{x}^2 + \dot{h}^2)(1 + c\alpha) \\ D &= a_D(\dot{x}^2 + \dot{h}^2)(1 + b(1 + c\alpha)^2) \end{aligned} \tag{2.11}$$

where a_L, a_D are dimensionless *lift* and *drag coefficients*, and b and c are positive constants. We assume that the autopilot has direct control over both the forward thrust

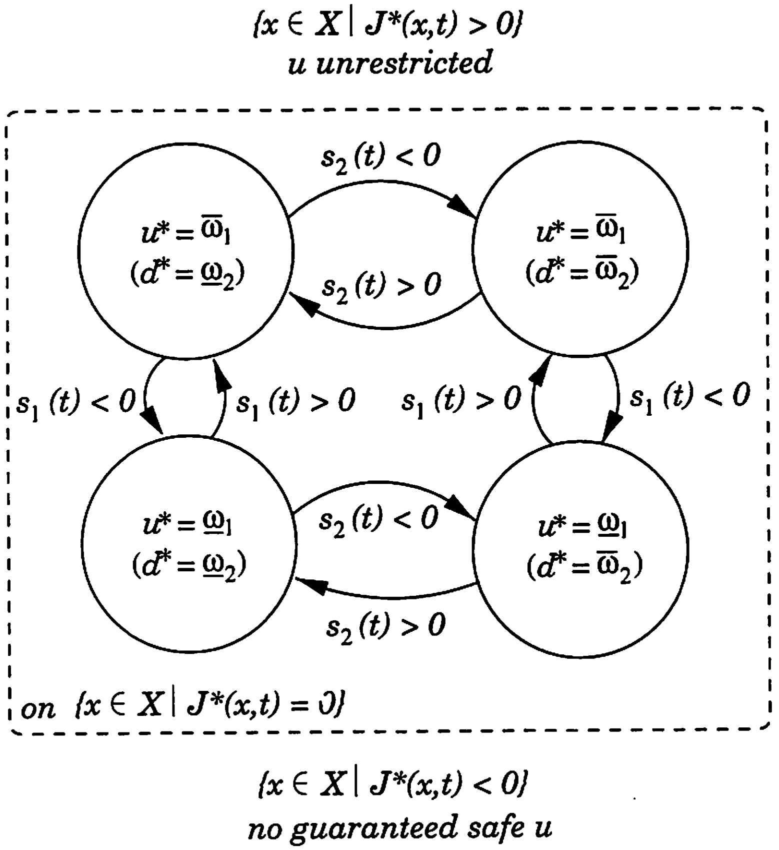

$\{x \in X \mid J^*(x,t) > 0\}$
u unrestricted
$s_2(t) < 0$
$u^* = \overline{\omega}_1$
$(d^* = \underline{\omega}_2)$
$s_2(t) > 0$
$u^* = \overline{\omega}_1$
$(d^* = \overline{\omega}_2)$
$s_1(t) < 0$
$s_1(t) > 0$
$s_1(t) > 0$
$s_1(t) < 0$
$u^* = \underline{\omega}_1$
$(d^* = \underline{\omega}_2)$
$s_2(t) < 0$
$u^* = \underline{\omega}_1$
$(d^* = \overline{\omega}_2)$
$s_2(t) > 0$
on $\{x \in X \mid J^*(x,t) = 0\}$
$\{x \in X \mid J^*(x,t) < 0\}$
no guaranteed safe u

Figure 4.5: Switching law governing the two aircraft system with angular velocity control inputs. The law is least restrictive in that the control u is not restricted when the state is in $\{x \in X \mid J^*(x,t) > 0\}$. The diagonal transitions in the automaton for the boundary of $\{x \in X \mid J^*(x,t) = 0\}$ are not labeled for legibility. In practice, t should be chosen large enough to take into account aircraft in the alert zone.

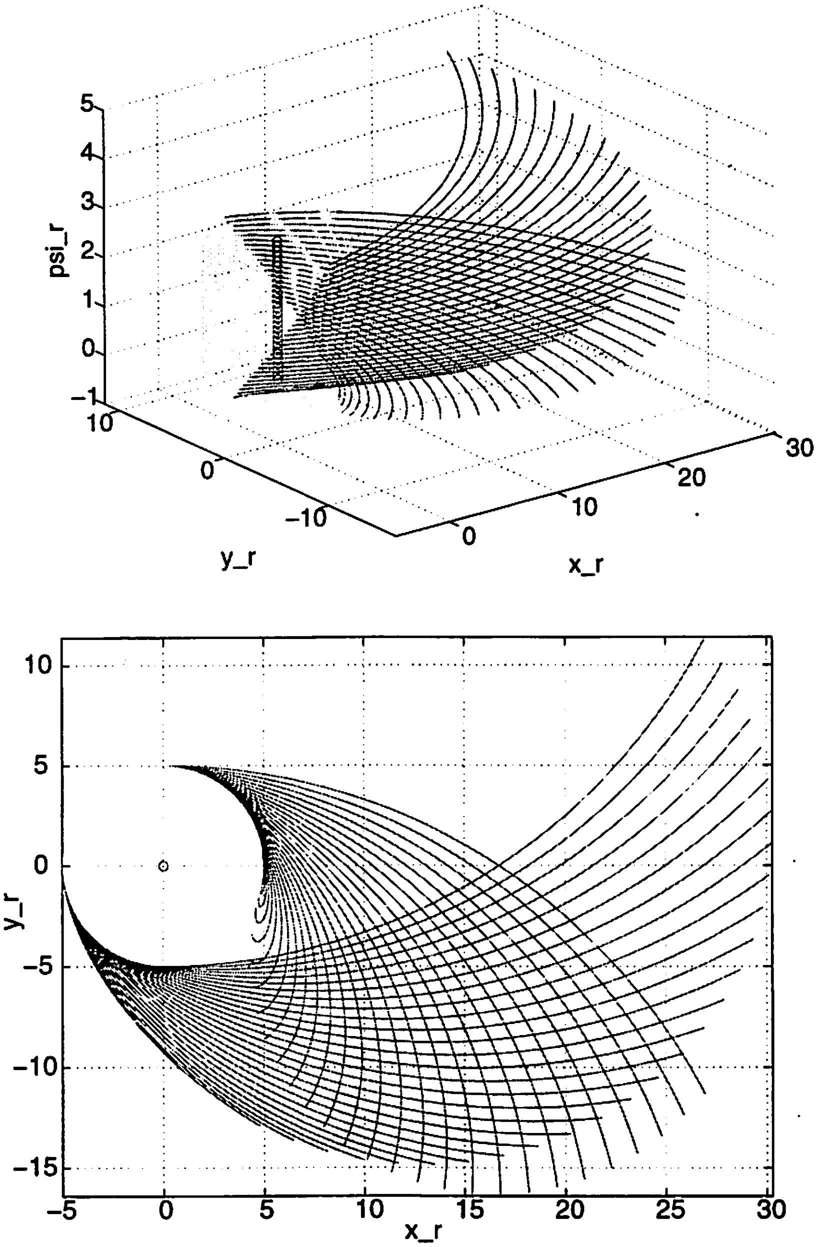


Figure 4.4: The set $G = \{(x_r, y_r), \psi_r \in (0, \pi) \mid x_r^2 + y_r^2 \leq 5^2\}$ (cylinder) and the set $\{x \in X \mid J^*(x,t) = 0\}$ for $t < 0$ being the time of the first switch in either $s_1(t)$ or $s_2(t)$. The second picture is a top view of the first.

For points $(x_r, y_r, \psi_r) \in \partial G$ such that $\psi_r \in (0, \pi)$ it is straightforward to show that $\dot{s}_1 > 0$ and $\dot{s}_2 > 0$, meaning that for values of t slightly less than 0, $s_1 < 0$ and $s_2 < 0$. Thus for this range of points along ∂G, $u^*(0) = -1$ and $d^*(0) = 1$. These values for u^* and d^* remain valid for $t < 0$ as long as $s_1(t) < 0$ and $s_2(t) < 0$. When $s_1(t) = 0$ and $s_2(t) = 0$, the optimal solution (u^*, d^*) switches and the computation of the boundary continues with the new values of u^* and d^*, thus introducing "kinks" into the boundary. These points correspond to loss of smoothness in the Hamilton-Jacobi equation. Figure 4.4 displays the resulting boundary $\{x \in X \mid J^*(x,t) = 0\}$, computed by solving the Hamilton-Jacobi equation (4.37) locally using Hamilton's equations, for $t < 0$ being the first time that either $s_1(t)$ or $s_2(t)$ switches.

The *least restrictive control scheme* for safety is shown in Figure 4.5.

The optimal Hamiltonian is

$$H^*(x,p) = \max_{u\in U}\min_{d\in D}[-p_1v_1+p_1v_2\cos\psi_r+p_2v_2\sin\psi_r+(p_1y_r-p_2x_r-p_3)u+p_3d] \quad (4.53)$$

Defining the *switching functions* $s_1(t)$ and $s_2(t)$, as

$$\begin{aligned} s_1(t) &= p_1(t)y_r(t) - p_2(t)x_r(t) - p_3(t) \\ s_2(t) &= p_3(t) \end{aligned} \quad (4.54)$$

the optimal control and disturbance u^* and d^* exist when $s_1 \neq 0$ and $s_2 \neq 0$ and are calculated as

$$\begin{aligned} u^* &= sgn(s_1) \\ d^* &= -sgn(s_2) \end{aligned} \quad (4.55)$$

The equations for $\dot{p}$ are obtained through Hamilton's equation (4.20) and are

$$\begin{aligned} \dot{p}_1 &= u^*p_2 \\ \dot{p}_2 &= -u^*p_1 \\ \dot{p}_3 &= p_1v_2\sin\psi_r - p_2v_2\cos\psi_r \end{aligned} \quad (4.56)$$

with $p(0) = (x_r, y_r, 0)^T = \nu$, the outward pointing normal to ∂G at any point (x_r, y_r, ψ_r) on ∂G.

The UP of ∂G is calculated using (4.8) with $\nu = (x_r, y_r, 0)^T$:

$$UP = \{(x_r, y_r, \psi_r) \in \partial G \mid -v_1x_r + v_2(x_r\cos\psi_r + y_r\sin\psi_r) < 0\} \quad (4.57)$$

with boundary

$$\{(x_r, y_r, \psi_r) \in \partial G \mid -v_1x_r + v_2(x_r\cos\psi_r + y_r\sin\psi_r) = 0\} \quad (4.58)$$

To solve for $p^*(t)$ and $x^*(t)$ for $t < 0$, we must first determine $u^*(0)$ and $d^*(0)$. Equations (4.55) are not defined at $t = 0$, since $s_1 = s_2 = 0$ on ∂G, giving rise to "abnormal extremals" [68] (meaning that the optimal Hamiltonian loses dependence on u and d at these points). Analogously to [36] (pages 442-443), we use an indirect method to calculate $u^*(0)$ and $d^*(0)$: at any point (x_r, y_r, ψ_r) on ∂G, the derivatives of the switching functions s_1 and s_2 are

$$\dot{s}_1 = y_rv_1 \quad (4.59)$$

$$\dot{s}_2 = x_rv_2\sin\psi_r - y_rv_2\cos\psi_r \quad (4.60)$$

Remark: In practice, we do not usually need to compute the fixed point W^*, rather just the set $\{x \in X \mid J^*(x,t) \geq 0\}$ for t a large enough "look-ahead" time.

A feedback controller for u that renders W^* invariant can now be constructed. The controller should be such that on ∂W^* only the u for which:

$$\min_{d \in D} \frac{\partial J^*(x)}{\partial x} f(x,u,d) \geq 0$$

are applied. In the interior of W^*, u is free to take on any value in U. Existence of such u for $x \in W^*$ is guaranteed by construction.

4.2.1 Example: The $SE(2)$ Aircraft

Consider the relative model of two aircraft in $SE(2)$, given in equations (2.9), for the case in which the linear velocities of both aircraft are fixed, $v_1, v_2 \in \mathbb{R}$, and the control inputs of the aircraft are the angular velocities, $u = \omega_1$ and $d = \omega_2$:

$$\begin{aligned} \dot{x}_r &= -v_1 + v_2 \cos\psi_r + u y_r \\ \dot{y}_r &= v_2 \sin\psi_r - u x_r \\ \dot{\psi}_r &= d - u \end{aligned} \tag{4.49}$$

with state variables $(x_r, y_r, \psi_r) \in \mathbb{R}^2 \times [-\pi, \pi)$ and control and disturbance inputs $u \in U = [\underline{\omega}_1, \overline{\omega}_1] \subset \mathbb{R}$, $d \in D = [\underline{\omega}_2, \overline{\omega}_2] \subset \mathbb{R}$. Without loss of generality (we scale the coefficients of u and d if this is not met), assume that $\underline{\omega}_i = -1$ and $\overline{\omega}_i = 1$, for $i = 1, 2$.

The set G is defined in the relative frame:

$$G = \{(x_r, y_r) \in \mathbb{R}^2, \psi_r \in [-\pi, \pi) \mid x_r^2 + y_r^2 \leq 5^2\} \tag{4.50}$$

and the capture set is defined as the interior of G

$$G^\circ = \{(x_r, y_r) \in \mathbb{R}^2, \psi_r \in [-\pi, \pi) \mid x_r^2 + y_r^2 < 5^2\} \tag{4.51}$$

which is a 5-mile-radius cylindrical block in the (x_r, y_r, ψ_r) space denoting the protected zone in the relative frame. The function $l(x)$ is defined as

$$l(x) = x_r^2 + y_r^2 - 5^2 \tag{4.52}$$

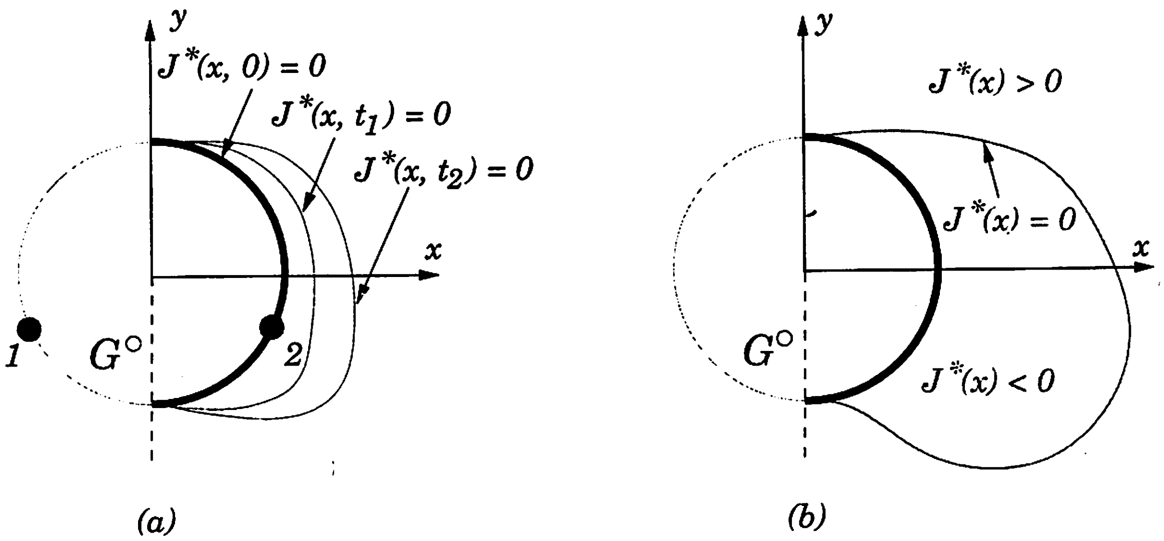


Figure 4.3: (a) The sets $\{x \in X \mid J^*(x,0) = 0\}$, $\{x \in X \mid J^*(x,t_1) = 0\}$, $\{x \in X \mid J^*(x,t_2) = 0\}$ for $0 > t_1 > t_2$. (b) The fixed point $\{x \in X \mid J^*(x) < 0\}$, $\{x \in X \mid J^*(x) = 0\}$, and $\{x \in X \mid J^*(x) > 0\}$.

Proof: Let $x_0 \in \{x \in X \mid J^*(x) < 0\}$. Therefore, by construction, for all $u(\cdot) \in \mathcal{U}$ there exists $d(\cdot) \in \mathcal{D}$ such that the state trajectory $x(\cdot)$, starting at $(x_0, 0)$, will eventually enter G°. Thus $x_0 \notin W^*$.

Now let $x_0 \in \{x \in X \mid J^*(x) \geq 0\}$. Assume for the sake of contradiction that for all $u(\cdot) \in \mathcal{U}$, there exists a $d(\cdot) \in \mathcal{D}$ such that the trajectory $x(\cdot)$, starting at $(x_0, 0)$, enters G°. Since for all $x \in G^\circ$, $J^*(x) < 0$, there exists a time $t_1 > 0$ at which this trajectory crosses $\{x \in X \mid J^*(x) = 0\}$. However, for all x such that $J^*(x) = 0$, there must exist a $u \in U$ such that for all $d \in D$, $f(x, u, d)$ points outside of $\{x \in X \mid J^*(x) < 0\}$. The set $\{x \in X \mid J^*(x) = 0\}$ therefore acts like a "barrier" to disturbance functions $d(\cdot)$, and the fact that the trajectory must cross this set contradicts the assumption of existence of a $d(\cdot)$ which drives the system into G°. Thus $x_0 \in W^*$. Therefore, $W^* = \{x \in X \mid J^*(x) \geq 0\}$. ∎

Proposition 4 (Characterization of W^*) *W^* is the largest controlled invariant set contained in $F = (G^\circ)^c$.*

Proof: From equation (4.37), $\frac{\partial J^*(x,t)}{\partial t} \geq 0$, when $J^*(x,t) \leq 0$. Thus $J^*(x,t)$ is a monotone non-increasing function of $(-t)$, so that as t decreases, the set $\{x \in X \mid J^*(x,t) \leq 0\}$ does not decrease in size. ■

We claim that $\{x \in X \mid J^*(x,t) < 0\}$, where $J^*(x,t)$ is the solution to (4.37), is the set of states from which the environment can force the system into G° in at most $|t|$ seconds. Before proving this, we give an intuitive explanation.

Clearly, $\partial G = \{x \in X \mid J^*(x,0) = 0\}$. Consider $x_0 \in \partial G$ such that x_0 does not belong to the UP (shown as "1" in Figure 4.3(a)). Thus, there exists a $u \in U$ such that for all $d \in D$, $f(x_0, u, d)$ points outside of G°. Therefore,

$$\exists u \in U \; \forall d \in D \; \frac{\partial J^*(x_0,t)}{\partial x} f(x_0,u,d) \geq 0 \tag{4.42}$$

Therefore,

$$H^*(x_0, \frac{\partial J^*(x_0,t)}{\partial x}) \geq 0 \tag{4.43}$$

Thus, from (4.37),

$$\frac{\partial J^*(x_0,t)}{\partial t} = 0 \tag{4.44}$$

Thus the part of ∂G which is not in the UP remains stationary under (4.37). Now consider $x_0 \in UP$ (shown as "2" in Figure 4.3(a)). For all $u \in U$ there exists a $d \in D$ such that $f(x_0,u,d)$ points into G°. Therefore

$$\forall u \in U \; \exists d \in D \; \frac{\partial J^*(x_0,t)}{\partial x} f(x_0,u,d) < 0 \tag{4.45}$$

Therefore,

$$H^*(x_0, \frac{\partial J^*(x_0,t)}{\partial x}) < 0 \tag{4.46}$$

and thus

$$\frac{\partial J^*(x_0,t)}{\partial t} > 0 \tag{4.47}$$

meaning that $J^*(x,t)$ is decreasing with decreasing t. Thus the UP "grows outwards" under (4.37).

Proposition 3 (Winning States W^*) *Assume that $J^*(x,t)$ satisfies the Hamilton-Jacobi equation (4.37) for all t, and that it converges uniformly in x as $t \to -\infty$ to a function $J^*(x)$. Then the set of winning states for the controller is*

$$W^* = \{x \in X \mid J^*(x) \geq 0\} \tag{4.48}$$

as t evolves. The discontinuity on the right hand side of equation (4.37) further complicates the solution, as does the discontinuous switching of the optimal control and disturbance u^* and d^*. In addition, we are often interested in cases in which G has non-smooth boundary, so that the boundary conditions of the Hamilton-Jacobi equation are not differentiable. In order to admit discontinuous solutions, a "weak" derivative and "weak" solution to the Hamilton-Jacobi equations was developed by Crandall, Lions, and Evans in the early 1980's [66, 67]. We define a *viscosity solution* to (4.37) as the limit as ϵ goes to zero of solutions $J^*_\epsilon(x,t)$ to the partial differential equation:

$$-\frac{\partial J^*_\epsilon(x,t)}{\partial t} = \begin{cases} H^*(x, \frac{\partial J^*_\epsilon(x,t)}{\partial x}) + \epsilon\Delta J^*_\epsilon(x,t) & \text{for } \{x \in X \mid J^*_\epsilon(x,t) > 0\} \\ \min\{0, H^*(x, \frac{\partial J^*_\epsilon(x,t)}{\partial x})\} + \epsilon\Delta J^*_\epsilon(x,t) & \text{for } \{x \in X \mid J^*_\epsilon(x,t) \leq 0\} \end{cases} \tag{4.39}$$

with initial data $J^*_\epsilon(x,0) = l_\epsilon(x)$, a smooth outer approximation to the boundary of G. Here, ΔJ^* refers to the Laplacian of J^*, namely

$$\Delta J^* = \sum_{i=1}^{n} \frac{\partial^2 J^*}{\partial x_i^2} \tag{4.40}$$

For $\epsilon > 0$ and for smooth Hamiltonian[3] it may be shown [66, 67] that there exists a unique continuous solution to the Hamilton-Jacobi equation: the second derivative term $\Delta J_\epsilon(x,t)$ acts like a smoothing term and is called a "viscosity" term for that reason. As $\epsilon \to 0$, the solution $J_\epsilon(x,t)$ approaches the viscosity solution to the Hamilton-Jacobi equation. Thus, even when classical smooth solutions do not exist, solutions in this "weak sense" exist. In Chapter 7, we present a numerical scheme, developed by Osher and Sethian [38], which computes this viscosity solution.

Lemma 1 *For all $t_2 \leq t_1 \leq 0$,*

$$\{x \in X \mid J^*(x,t_1) \leq 0\} \subseteq \{x \in X \mid J^*(x,t_2) \leq 0\} \tag{4.41}$$

[3] The Crandall-Evans-Lions definition of a viscosity solution is for

$$-\frac{\partial J_\epsilon(x,t)}{\partial t} = H(x, \frac{\partial J_\epsilon(x,t)}{\partial x}) + \epsilon\Delta J_\epsilon(x,t)$$

for smooth Hamiltonians $H(x, \frac{\partial J_\epsilon(x,t)}{\partial x})$. Our current work involves extending these results to our cases, with piecewise smooth Hamiltonians.

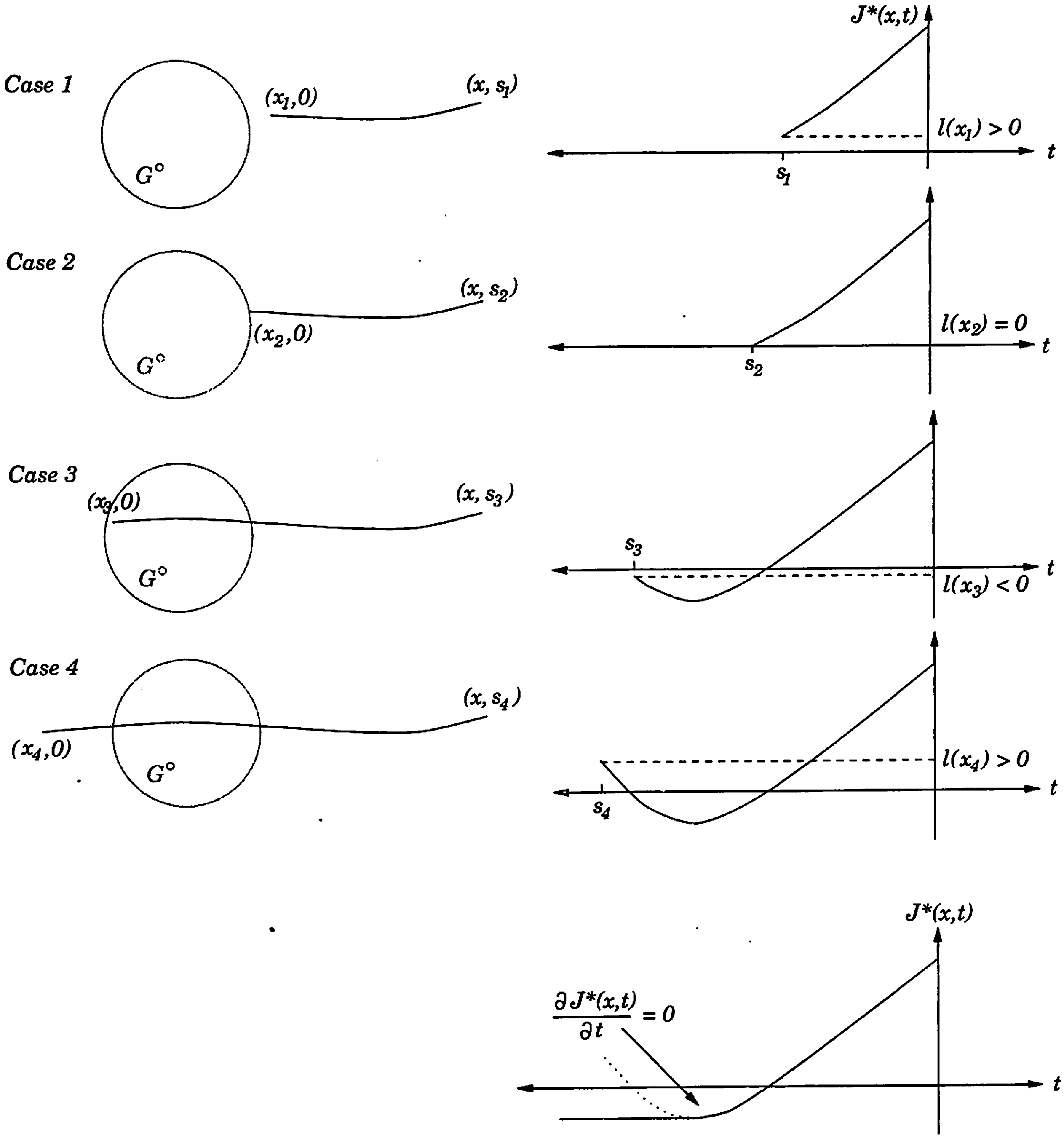


Figure 4.2: The left column displays four cases of optimal trajectories, starting at x at time s_i, and ending at state x_i at time 0, where $0 > s_1 > s_2 > s_3 > s_4$. The right column displays $J^*(x,t)$ for fixed x. Note that the standard variational problem produces states that can change from "unsafe" to "safe". The figure at the bottom of the right column displays the result of modifying the Hamilton-Jacobi equation so that, once $J^*(x,t)$ is negative, its evolution is non-increasing in negative time.

Since $H = p^T f(x, u, d)$, we have the *Hamilton-Jacobi partial differential equation:*

$$-\frac{\partial J^*(x,t)}{\partial t} = H^*(x, \frac{\partial J^*(x,t)}{\partial x}) \tag{4.36}$$

with boundary condition $J^*(x, 0) = l(x)$.

As indicated in the discussion following (4.15), the solution $J^*(x,t)$ to equation (4.36) counts as safe those states for which optimal trajectories pass through G° and end up outside G° at time 0. Figure 4.2 illustrates such a situation as a sequence of "snapshots" for times s_1, s_2, s_3, and s_4, where $0 > s_1 > s_2 > .s_3 > s_4$. In this example, $J^*(x, s_1) > 0$, $J^*(x, s_2) = 0$, $J^*(x, s_3) < 0$, and $J^*(x, s_4) > 0$, indicating that the optimal trajectory which starts at state x will end up in G° after s_3 seconds, but will leave G° before s_4 seconds is up. To force such trajectories to stay inside G°, we modify equation (4.36) to guarantee that, if for any $x \in X$ there exists an $s \in [t, 0]$ such that $J^*(x, s) < 0$, then $J^*(x,t)$ is non-increasing for time less than s:

$$-\frac{\partial J^*(x,t)}{\partial t} = \begin{cases} H^*(x, \frac{\partial J^*(x,t)}{\partial x}) & \text{for } \{x \in X \mid J^*(x,t) > 0\} \\ \min\{0, H^*(x, \frac{\partial J^*(x,t)}{\partial x})\} & \text{for } \{x \in X \mid J^*(x,t) \leq 0\} \end{cases} \tag{4.37}$$

with boundary condition $J^*(x, 0) = l(x)$. Equation (4.37) is the continuous analog to equation (4.6) of the preceding discrete game, and describes the relationship between the time and state evolution of $J^*(x,t)$.

If the control and disturbance act optimally, the set $\{x \in X \mid J^*(x,t) \geq 0\}$ describes those states from which the controller can keep the system state outside of G° for at least $|t|$ seconds, and the set $\{x \in X \mid J^*(x,t) < 0\}$ is the set of states from which the environment can force the system into G° in at most $|t|$ seconds. The continuous-time analog to (4.1), the iterative method of calculating the winning states for the controller, is therefore:

$$\begin{aligned} W^0 &= (G^\circ)^c \\ W^t &= \{x \in X \mid J^*(x,t) \geq 0\} \end{aligned} \tag{4.38}$$

A restrictive assumption in this derivation is that the function $J^*(x,t)$ is a *smooth* function of x and t. In general, even if the boundary condition $J^*(x, 0) = l(x)$ is differentiable, the solution $J^*(x,t)$ may develop discontinuities in x, known as *shocks*,

The Hamilton-Jacobi Equation

With the solutions to the optimal input and disturbance, we now derive the partial differential equation which the optimal value function $J^*(x,t)$ satisfies. Recall that $J^*(x,t)$ is defined as

$$J^*(x,t) = \max_{u(\cdot)\in\mathcal{U}} \min_{d(\cdot)\in\mathcal{D}} l(x(0)) \tag{4.30}$$

and is interpreted as the cost of the system starting from state x at time t and proceeding to state $x(0)$ at time 0, using $u^*(s)$ and $d^*(s)$ for $s \in [t,0]$. We make the assumptions that $J^*(x,t)$ exists and is a smooth function[2] of x and t. Now suppose the system starts at (x,t) but uses input and disturbance trajectories, $u(s)$ and $d(s)$, not necessarily equal to the optimal ones, for the first Δt seconds. At $(x+\Delta x, t+\Delta t)$, the system switches back to using $u^*(s)$ and $d^*(s)$ until $(x(0),0)$. Bellman's principle of optimality states that the value function for such a trajectory is

$$J^1(x,t) \triangleq J^*(x+\Delta x, t+\Delta t) \tag{4.31}$$

$J^1(x,t)$ is equal to the optimal value function $J^*(x,t)$ if $(u^*(s), d^*(s))$ is used in the initial interval $s \in [t, t+\Delta t]$:

$$J^*(x,t) = \max_{u(\cdot)\in\mathcal{U}} \min_{d(\cdot)\in\mathcal{D}} J^*(x+\Delta x, t+\Delta t) \tag{4.32}$$

for $u(\cdot), d(\cdot)$ valid over $[t, t+\Delta t]$. Since $J^*(x,t)$ is assumed continuous and differentiable, the right hand side may be approximated by

$$\max_u \min_d \left(J^*(x,t) + \frac{\partial J^*(x,t)}{\partial x} f(x,u,d)\Delta t + \frac{\partial J^*(x,t)}{\partial t}\Delta t \right) \tag{4.33}$$

Taking the limit as $\Delta t \to 0$ yields

$$-\frac{\partial J^*(x,t)}{\partial t} = \max_u \min_d \frac{\partial J^*(x,t)}{\partial x} f(x,u,d) \tag{4.34}$$

From equation (4.18), it is evident that along the optimal trajectory, small perturbations in x produce small perturbations in J^* according to $\delta J^* = p^T \delta x$, so that

$$\frac{\partial J^*}{\partial x} = p^T \tag{4.35}$$

[2] While this very restrictive assumption is necessary for the derivation of the Hamilton-Jacobi equation, we will show how it may be relaxed for the algorithms in this dissertation.

To be an extremum, $\delta\tilde{J}$ must be zero for arbitrary δu and δd and δp, which happens only if:

$$\begin{aligned}\left(\frac{\partial H(x,p,u,d)}{\partial p}\right)^T - \dot{x} &= 0 && (4.21)\\ \frac{\partial H(x,p,u,d)}{\partial u} &= 0 && (4.22)\\ \frac{\partial H(x,p,u,d)}{\partial d} &= 0 && (4.23)\end{aligned}$$

Equations (4.20) and (4.21) are known as *Hamilton's equations.* Equations (4.19) through (4.23) are the *necessary conditions for local optimality* of $(x(\cdot),u(\cdot),d(\cdot))$: we denote a system trajectory which satisfies equations (4.19) through (4.23) as $(x^*(\cdot),u^*(\cdot),d^*(\cdot))$.

Sufficient conditions for (local) optimality of $(x^*(\cdot),u^*(\cdot),d^*(\cdot))$ are that the Hessians of the Hamiltonian,

$$\frac{\partial^2 H(x^*,p^*,u^*,d^*)}{\partial u^2} \qquad (4.24)$$

$$\frac{\partial^2 H(x^*,p^*,u^*,d^*)}{\partial d^2} \qquad (4.25)$$

be respectively negative and positive definite along the optimal trajectory.

While the preceding results are local, they have been generalized (see [33, 34, 35]) to be global, and the local optimality conditions (4.22), (4.23), (4.24), (4.25) globalized to:

$$\begin{aligned} u^* &= \arg\max_{u\in U}\min_{d\in D} H(x,p,u,d) && (4.26)\\ d^* &= \arg\min_{d\in D} H(x,p,u^*,d) && (4.27)\end{aligned}$$

The *optimal Hamiltonian* is therefore given by

$$\begin{aligned} H^*(x,p) &= \max_{u\in U}\min_{d\in D} H(x,p,u,d) && (4.28)\\ &= \max_{u\in U}\min_{d\in D} p^T f(x,u,d) && (4.29)\end{aligned}$$

and satisfies Hamilton's equations, with final boundary condition $p(0) = \frac{\partial l}{\partial x}(x(0)) = \nu$ given by equation (4.19).

a functional minimization problem to a static optimization problem). From these conditions we calculate $(u^*(\cdot), d^*(\cdot))$. We then derive the associated Hamilton-Jacobi partial differential equation, whose solution is $J^*(x,t)$ as defined in equation (4.12).

However, the set $\{x \in X \mid J^*(x,t) \geq 0\}$ may include states from which there exist trajectories $x(\cdot)$ which enter G° and then leave G° within the interval $[t, 0)$. In order to ensure that trajectories are *captured* once they enter G°, we will show how the Hamilton-Jacobi equation may be modified so that its solution counts as unsafe those states for which the optimal trajectories *pass through* G° but end up *outside* G° at time 0.

Define the *modified cost function* as

$$\tilde{J}(x, p, u(\cdot), d(\cdot), t) = l(x(0)) + \int_t^0 p^T(s)(f(x(s), u(s), d(s)) - \dot{x}(s))ds \qquad (4.16)$$

where $p(s) \in \mathbb{R}^n$ is the vector of Lagrange multipliers. Clearly, a system trajectory is an optimal solution to (4.9) with dynamic constraints (3.3) if it is an optimal solution to the modified cost function (4.16). The first term in the integrand is defined to be the *Hamiltonian* of the system:

$$H(x, p, u, d) \triangleq p^T f(x, u, d) \qquad (4.17)$$

for all $s \in [t, 0]$. The calculus of variations involves perturbing u and d in $\tilde{J}$ by small amounts δu and δd, and analyzing the resulting variations in $\delta \tilde{J}$, δx, $\delta x(0)$, and δp:

$$\begin{aligned}
\delta \tilde{J} &= \frac{\partial l}{\partial x}(x(0))\,\delta x(0) + \int_t^0 \left[\frac{\partial H(x,p,u,d)}{\partial x}\delta x + \left(\left(\frac{\partial H(x,p,u,d)}{\partial p} \right)^T - \dot{x} \right) \delta p \right. \\
&+ \left. \frac{\partial H(x,p,u,d)}{\partial u}\delta u + \frac{\partial H(x,p,u,d)}{\partial d}\delta d - p^T \delta \dot{x} \right] ds \\
&= \left(\frac{\partial l}{\partial x}(x(0)) - p^T(0) \right) \delta x(0) + p^T(t)\delta x(t) + \int_t^0 \left[\left(\frac{\partial H(x,p,u,d)}{\partial x} + \dot{p}^T \right) \delta x + \right. \\
& \left. \left(\left(\frac{\partial H(x,p,u,d)}{\partial p} \right)^T - \dot{x} \right) \delta p + \frac{\partial H(x,p,u,d)}{\partial u}\delta u + \frac{\partial H(x,p,u,d)}{\partial d}\delta d \right] ds \qquad (4.18)
\end{aligned}$$

The Lagrange multipliers p are chosen to make the coefficients of δx and $\delta x(0)$ vanish:

$$\begin{aligned}
\frac{\partial l}{\partial x}(x(0)) - p^T(0) &= 0 \qquad (4.19) \\
\frac{\partial H(x,p,u,d)}{\partial x} + \dot{p}^T &= 0 \qquad (4.20)
\end{aligned}$$

controller's play. This corresponds to

$$\max_{u(\cdot)\in\mathcal{U}}\min_{d(\cdot)\in\mathcal{D}} J(x,u(\cdot),d(\cdot),t) \tag{4.10}$$

The drawback of this formalism is that the disturbance player at time t has knowledge of $u(\cdot)$ at future times. We will need to modify $\mathcal{U}$ to a space which does not allow for this "non-causal" behavior of the disturbance player. Assume that the game is played from $[t,0]$ and let $s\in(t,0]$. Define the space of controls which is truncated at time s as $\mathcal{U}_s$, and the *admissible* space of controls as

$$\mathcal{U}^{adm}=\cup_{t<s\leq 0}\,\mathcal{U}_s \tag{4.11}$$

At each $s\in(t,0]$, the maximization in (4.10) is performed over the set $\mathcal{U}^{adm}$. As far as the disturbance is concerned, if the full state x is known at time s, the knowledge of the input $u(\cdot)$ in the interval $[t,s)$ is not relevant. Thus in the full state observation case only $u(s)$ is needed. For games with more complex information patterns, such as imperfect or partial state information, the problem becomes very interesting and quite difficult to solve [64]. For aesthetic purposes, we will represent $\mathcal{U}^{adm}$ as $\mathcal{U}$ in the remainder of this dissertation.

We define $J^*(x,t)$, the optimal cost, as

$$J^*(x,t)=\max_{u(\cdot)\in\mathcal{U}}\min_{d(\cdot)\in\mathcal{D}} J(x,u(\cdot),d(\cdot),t) \tag{4.12}$$

and the corresponding optimal input and disturbance as

$$\begin{aligned} u^*(\cdot) &= \arg\max_{u(\cdot)\in\mathcal{U}}\min_{d(\cdot)\in\mathcal{D}} J(x,u(\cdot),d(\cdot),t) && (4.13)\\ d^*(\cdot) &= \arg\min_{d(\cdot)\in\mathcal{D}} J(x,u^*(\cdot),d(\cdot),t) && (4.14)\end{aligned}$$

In the following, we use standard results in optimal control theory [33], [34], [35], [65] to derive the necessary conditions for optimality of the system trajectory

$$(x(\cdot),u(\cdot),d(\cdot)) \tag{4.15}$$

by applying the calculus of variations to a modified value function which incorporates the dynamic constraints (3.3) (known as a *Legendre transformation*, converting

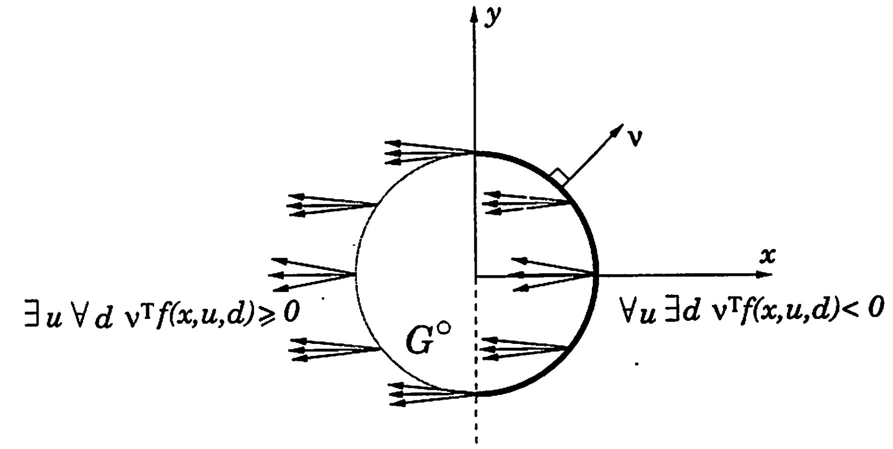

y
v
x
$\exists u \, \forall d \;\, v^T f(x,u,d) \geq 0$
G°
$\forall u \, \exists d \;\, v^T f(x,u,d) < 0$

Figure 4.1: The capture set G°, its outward pointing normal ν, and the cones of vector field directions at points on ∂G.

The Value Function and Hamilton's Equations

Consider the system (3.3) over the time interval $[t, 0]$, where $t < 0$. The value function of the game is defined by:

$$\begin{aligned} & J(x, u(\cdot), d(\cdot), t) : X \times \mathcal{U} \times \mathcal{D} \times \mathbb{R}_- \to \mathbb{R} \\ & \text{such that} \\ & J(x, u(\cdot), d(\cdot), t) = l(x(0)) \end{aligned} \tag{4.9}$$

This value function is interpreted as the cost of a trajectory $x(\cdot)$ which starts at x at initial time $t \leq 0$ (free), evolves according to (3.3) with input $(u(\cdot), d(\cdot))$, and ends at the final state $x(0)$, with cost $l(x(0))$. Note that the value function depends only on the final state: there is no running cost, or *Lagrangian*. This encodes the fact that when we are considering system safety, we are only interested in whether or not the system trajectory ends in G° and are not concerned with intermediate states. The game is won by the environment if the terminal state $x(0)$ is in G° (i.e. $J(x, u(\cdot), d(\cdot), t) < 0$), and is won by the controller otherwise.

The optimal action of the controller is one which tries to maximize the minimum cost, to try to counteract the optimal disturbance action of pushing the system towards G. As in the discrete game, the disturbance is given the advantage: the control $u(\cdot)$ plays first and disturbance $d(\cdot)$ plays second with the knowledge of the

A feedback controller for σ_1 that renders W^* invariant can now be constructed. For all $q \in W^*$ the controller allows only the $\sigma_1 \in \Sigma_1$ for which:

$$\min_{\sigma_2 \in \Sigma_2} \min_{q' \in \delta(q,\sigma_1,\sigma_2)} J^*(q') = 1$$

Existence of such σ_1 for all $q \in W^*$ is guaranteed by construction.

Proposition 2 (Characterization of W^*) *W^* is the largest controlled invariant subset of F.*

4.2 Continuous-Time Hamilton-Jacobi Equation

Consider now the dynamic counterpart of the above class of discrete games: two-player zero-sum dynamic games on nonlinear continuous-time systems (3.3), called *pursuit-evasion games.* The controller wins if it can keep the system from entering the interior of the set G, denoted $G^\circ = \{x \in X \mid l(x) < 0\}$ for a differentiable function $l : X \to \mathbb{R}$, with boundary ∂G. Conversely, the environment wins if it can drive the system into G°. As in the previous section, we describe the calculation of the set of states from which the controller can always win.

State Space Partition

The winning states for the controller are those states $W^* \subseteq X$ from which there exists a control law $u(\cdot) \in \mathcal{U}$ which can keep the system outside G° despite the disturbance $d(\cdot) \in \mathcal{D}$. Define the outward pointing normal to G as:

$$\nu = \frac{\partial l(x)}{\partial x} \tag{4.7}$$

The states on ∂G which can be forced into G° infinitesimally constitute the *usable part* (UP) of ∂G [36]. These are the states for which the disturbance can force the vector field to point inside G°:

$$\text{UP} = \{x \in \partial G \mid \forall u \in U \; \exists d \in D \quad \nu^T f(x,u,d) < 0\} \tag{4.8}$$

Figure 4.1 displays an example, with the UP of ∂G shown in bold.

with equality occurring when the action (σ_1, σ_2) is a *saddle solution*, or a *no regret* solution for each player. Here, we do not need to assume the existence of a saddle solution, rather we always give advantage to the environment, the player doing its worst to drive the system out of F, in order to ensure a conservative solution.

The iteration process (4.1) may be summarized by the difference equation:

$$J(q, i-1) - J(q, i) = \min\{0, \max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} [\min_{q' \in \delta(q, \sigma_1, \sigma_2)} J(q', i) - J(q, i)]\} \qquad (4.6)$$

which describes the relationship between the change in $J(\cdot)$ due to one step of the iteration and the change in $J(\cdot)$ due to one state transition. We call equation (4.6) the "discrete Hamilton-Jacobi equation" for this reason. The first "min" in the equation ensures that states outside W^i that can be forced by the controller to transition into W^i are prevented from appearing in W^{i-1}. This means that once a state has associated to it a value of zero, the value stays at zero for all subsequent iterations: enforcing the requirement that "once a state becomes unsafe, it remains unsafe".

Proposition 1 (Winning States W^*) *A fixed point $J^*(q)$ of (4.6) is reached in a finite number of steps. The set of winning states for the controller is $W^* = \{q \in Q \mid J^*(q) = 1\}$.*

Proof: First note that, by induction on equation (4.6), once $J(q, i) = 0$ for some i, then $J(q, j) = 0$ for $j < i$. That the fixed point $J^*(q)$ is reached in a finite number of steps follows from this and the fact that $|Q|$ is finite.

Suppose that the fixed point is reached at $i = k$. Let q be a winning state. Thus the controller has a sequence of actions which ensures that the system, starting at q, remains in F for at least k steps. Thus $q \in W^k$. Thus $q \in \{q \in Q \mid J^*(q) = 1\}$. Therefore, $W^* \subseteq \{q \in Q \mid J^*(q) = 1\}$.

Now suppose that $q \in \{q \in Q \mid J^*(q) = 1\}$, and the environment has a sequence of actions which drives the system out of F. Thus, for some $i \in \{0, -1, \ldots, k\}$,

$$\max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} \min_{q' \in \delta(q, \sigma_1, \sigma_2)} J(q', i+1) = 0$$

which implies, from equation (4.6) that $J(q, i) = 0$. This in turn implies that $J(q, j) = 0$ for $j < i$. Thus $J^*(q) = 0$, which is a contradiction. Therefore, $\{q \in Q \mid J^*(q) = 1\} \subseteq W^*$. ■

to remain in F despite the actions of the environment $\sigma_2[\cdot]$. The set W^* can be calculated as the fixed point of the following iteration (where a negative index $i \in \mathbb{Z}_-$ is used to indicate that each step is a predecessor operation):

$$\begin{aligned} W^0 &= F \\ W^{i-1} &= W^i \cap \{q \in Q \mid \exists \sigma_1 \in \Sigma_1\ \forall \sigma_2 \in \Sigma_2\ \delta(q, \sigma_1, \sigma_2) \subseteq W^i\} \end{aligned} \tag{4.1}$$

The iteration terminates when $W^i = W^{i-1} \triangleq W^*$. At each step of the iteration, $W^{i-1} \subseteq W^i$, thus due to the assumption of the finiteness of $|Q|$, the iteration terminates in a finite number of steps. The set W^i contains those states for which the controller has a sequence of actions $\sigma_1[i]\sigma_1[i+1]\ldots\sigma_1[0]$ which will ensure that the system remains in F for at least i steps, for all possible actions $\sigma_2[\cdot] \in \Sigma_2$.

The Value Function

Define the *value function* for this system as

$$J(q,i) : Q \times \mathbb{Z}_- \to \{0,1\} \tag{4.2}$$

such that

$$J(q,i) = \begin{cases} 1 & q \in W^i \\ 0 & q \in (W^i)^c \end{cases} \tag{4.3}$$

Therefore, $W^i = \{q \in Q \mid J(q,i) = 1\}$. Since the controller tries to keep the system inside F while the environment tries to force the system out of F,

$$\max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} \min_{q' \in \delta(q,\sigma_1,\sigma_2)} J(q', i) = \begin{cases} 1 & \text{if } \exists \sigma_1 \in \Sigma_1 \forall \sigma_2 \in \Sigma_2, \delta(q, \sigma_1, \sigma_2) \subseteq W^i \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

The "$\min_{q' \in \delta(q,\sigma_1,\sigma_2)}$" in the above compensates for the nondeterminism in δ; the order of operations $\max_{\sigma_1} \min_{\sigma_2}$ means that the controller *plays first*, trying to maximize the minimum value of $J(\cdot)$. The environment has the advantage in this case, since it has "prior" knowledge of the controller's action when making its own choice. Therefore, in general,

$$\max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} \min_{q' \in \delta(q,\sigma_1,\sigma_2)} J(\cdot) \leq \min_{\sigma_2 \in \Sigma_2} \max_{\sigma_1 \in \Sigma_1} \min_{q' \in \delta(q,\sigma_1,\sigma_2)} J(\cdot) \tag{4.5}$$

In both the discrete and continuous cases, it was assumed that the goal of the environment could be directly orthogonal to that of the controller's. This is a key assumption in our derivation of controllers for safety critical systems: the control law must protect against worst case uncertainty in the actions of the environment. With most realistic systems, the designer has a model of the environment and its actions: the better the model, the more flexibility the designer has in choosing a control law.

We first summarize a class of two-player games on the finite state automaton, in which the goal of the controller is to force the system to remain inside a certain safe subset of the discrete state space, and the goal of the environment is to force the system to leave this same subset. We then present the continuous counterpart: a dynamic game on the continuous nonlinear system in which the control input tries to keep the system inside a safe subset of the continuous state space in the face of an environmental disturbance. Emphasis is placed in each case on the derivation of a *Hamilton-Jacobi equation*, whose solution delineates those states from which the system can remain inside the safe set from those states from which the system may be driven out of this set. These derivations serve as background for the next chapter, in which we treat the corresponding problem for nonlinear hybrid automata.

4.1 Discrete Hamilton-Jacobi Equation

Consider the finite automaton (3.1) with trajectory acceptance condition $\Omega = \Box F$, for $F \subseteq Q$ representing a safe set of states. We first describe the iteration process for calculating the set of states from which the controller can always keep the system inside F. We then show how this iteration process can be written as the difference equation of a *value* function, which we denote as the "discrete Hamilton-Jacobi equation".

State Space Partition

We define the *winning states* W^* for the controller as the subset of F from which the system (3.1) has a sequence of control actions $\sigma_1[\cdot]$ which can force the system

Chapter 4

Evolution of Boundaries for Discrete and Continuous Games

Consider the discrete finite state automaton of Section 3.1, given by (3.1):

$$(Q, \Sigma, \delta, Q_0, \Omega)$$

The problem of synthesizing control laws $\sigma_1[\cdot] \in \Sigma_1^\omega$ in the presence of uncertain actions $\sigma_2[\cdot] \in \Sigma_2^\omega$ was first posed by Church in 1962 [57], who was studying solutions to digital circuits, and was solved by Büchi and Landweber [32] and Rabin [58] in the late 1960's and early 1970's using a version of the von Neumann-Morgenstern discrete game [59]. More recently, Ramadge and Wonham [60] added new insight into the structure of the control law. A temporal logic for modeling such games is introduced in [61].

For the continuous nonlinear dynamics, described by equation (3.3):

$$\dot{x}(t) = f(x(t), u(t), d(t)), \quad x(0) \in X_0$$

the solution of an optimal control law $u(\cdot)$ in the presence of environmental uncertainties $d(\cdot)$ was solved as a zero-sum dynamic game by Isaacs in the early 1950's [62][1]. Solutions for linear differential games were presented by Pontrjagin in [63]. An excellent modern reference is [36].

[1] Isaacs was then a researcher at the Rand Corporation and was motivated by military problems in the U.S. Air Force (aircraft dog fights, target missiles).

$$
\begin{aligned}
Q \times X &= \{q_1, \ldots, q_5\} \times (\mathbb{R} \times \mathbb{R} \times \mathbb{R}^+ \times \mathbb{R}) \\
U &= [T_{min}, T_{max}] \times [\theta_{min}, \theta_{max}] \\
\Sigma_1 &= \{\sigma_1^{ij}\}, i \in \{1, \ldots, 5\}, j \in \{1, \ldots, 5\} \\
f(q, \mathbf{x}, u) &= \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{h} \\ \ddot{h} \end{bmatrix} \text{ where } \begin{bmatrix} \ddot{x} \\ \ddot{h} \end{bmatrix} = R(\theta) \left[R^T(\alpha) \begin{bmatrix} -D \\ L \end{bmatrix} + \begin{bmatrix} T \\ 0 \end{bmatrix} \right] + \begin{bmatrix} 0 \\ -Mg \end{bmatrix} \\
\delta(q_i, \mathbf{x}, \sigma_1^{ij}) &= (q_j, \mathbf{x}) \\
Inv &= (q_1, X) \cup (q_2, X) \cup (q_3, X) \cup (q_4, X) \cup (q_5, X) \\
I &= (q_1, F) \\
Y &= X \\
h(q, \mathbf{x}) &= \begin{cases} (V, \gamma) & \text{if } q = q_1 \\ V & \text{if } q = q_2 \\ \gamma & \text{if } q = q_3 \\ (V, h) & \text{if } q = q_4 \\ h & \text{if } q = q_5 \end{cases} \\
\Omega &= \Box F
\end{aligned}
$$

$$\begin{aligned}
\delta\left(q_4, \begin{bmatrix} x_r \\ y_r \\ \psi_r \\ \pi/2 \end{bmatrix}, \epsilon, \sigma_2^3\right) &= \left(q_5, \begin{bmatrix} x_r \\ y_r \\ \psi_r \\ 0 \end{bmatrix}\right) \\
Inv &= (q_1, X) \cup (q_2, \{x \in X \mid 0 \leq z \leq \pi/4\}) \\
&\quad \cup (q_3, \{x \in X \mid 0 \leq z \leq T\}) \\
&\quad \cup (q_4, \{x \in X \mid 0 \leq z \leq \pi/2\}) \\
&\quad \cup (q_5, \{x \in X \mid 0 \leq z \leq T\}) \\
&\quad \cup (q_6, \{x \in X \mid 0 \leq z \leq \pi/4\}) \cup (q_7, X) \\
I &= (q_1, \{x \in X \mid x_r^2 + y_r^2 > 5^2, z = 0\}) \\
Y &= X \\
h(q, x) &= (q, x) \\
\Omega &= \Box G^c
\end{aligned}$$

3.4.3 Flight Mode Switching and Envelope Protection Example

For the envelope protection example of Section 2.4.2, the discrete state may take on one of five possible values, $Q = \{q_1, \ldots, q_5\}$, corresponding to the five flight modes: (Speed, Flight Path), (Speed), (Flight Path), (Speed, Altitude), (Altitude). The continuous state of the system is $\mathbf{x} = (x, \dot{x}, h, \dot{h})^T$, with continuous dynamics specified by equation (2.13). The control inputs are the throttle T and pitch θ with input constraint set $U = [T_{min}, T_{max}] \times [\theta_{min}, \theta_{max}]$, and we assume for simplicity that there are no continuous disturbance inputs ($D = \emptyset$) (a possible extension to this problem would be to consider wind as a continuous disturbance). The controllable actions label transitions from each mode to every other mode: let σ_1^{ij}, for $i \in \{1, \ldots, 5\}$ and $j \in \{1, \ldots, 5\}$ be the action labeling the transition from q_i to q_j. We assume that there are no disturbance actions ($\Sigma_2 = \emptyset$) (although it is a very nice extension to introduce disturbance actions representing pilot error in manually switching modes). The safe set F is illustrated in Figure 2.11.

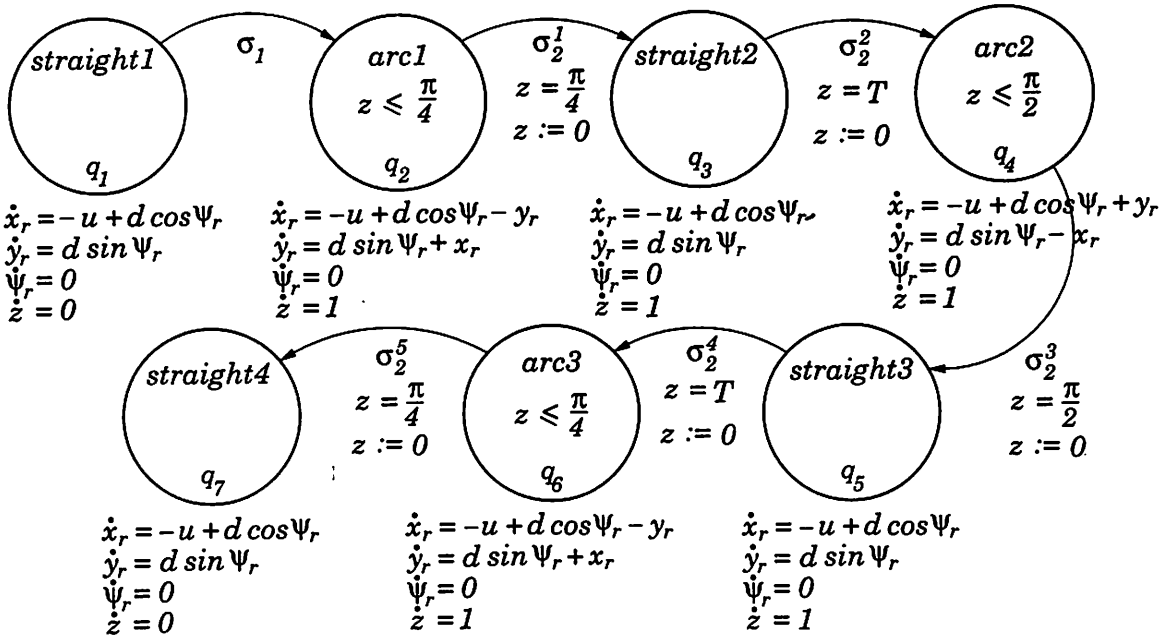

straight1
q_1
σ_1
arc1
$z \leqslant \frac{\pi}{4}$
q_2
σ_2^1
$z = \frac{\pi}{4}$
$z := 0$
straight2
q_3
σ_2^2
$z = T$
$z := 0$
arc2
$z \leqslant \frac{\pi}{2}$
q_4
$\dot{x}_r = -u + d\cos\Psi_r$
$\dot{y}_r = d\sin\Psi_r$
$\dot{\Psi}_r = 0$
$\dot{z} = 0$
$\dot{x}_r = -u + d\cos\Psi_r - y_r$
$\dot{y}_r = d\sin\Psi_r + x_r$
$\dot{\Psi}_r = 0$
$\dot{z} = 1$
$\dot{x}_r = -u + d\cos\Psi_r$
$\dot{y}_r = d\sin\Psi_r$
$\dot{\Psi}_r = 0$
$\dot{z} = 1$
$\dot{x}_r = -u + d\cos\Psi_r + y_r$
$\dot{y}_r = d\sin\Psi_r - x_r$
$\dot{\Psi}_r = 0$
$\dot{z} = 1$
straight4
q_7
σ_2^5
$z = \frac{\pi}{4}$
$z := 0$
arc3
$z \leqslant \frac{\pi}{4}$
q_6
σ_2^4
$z = T$
$z := 0$
straight3
q_5
σ_2^3
$z = \frac{\pi}{2}$
$z := 0$
$\dot{x}_r = -u + d\cos\Psi_r$
$\dot{y}_r = d\sin\Psi_r$
$\dot{\Psi}_r = 0$
$\dot{z} = 0$
$\dot{x}_r = -u + d\cos\Psi_r - y_r$
$\dot{y}_r = d\sin\Psi_r + x_r$
$\dot{\Psi}_r = 0$
$\dot{z} = 1$
$\dot{x}_r = -u + d\cos\Psi_r$
$\dot{y}_r = d\sin\Psi_r$
$\dot{\Psi}_r = 0$
$\dot{z} = 1$

Figure 3.4: Hybrid automaton modeling seven-mode conflict resolution maneuver.

$$\begin{aligned}
U \times D &= \mathbb{R}^2 \\
\Sigma_1 \times \Sigma_2 &= \{\sigma_1, \epsilon\} \times \{\sigma_2^1, \sigma_2^2, \sigma_2^3, \sigma_2^4, \sigma_2^5, \epsilon\} \\
f(q, x, u, d) &= \begin{bmatrix} -u + d\cos\psi_r + \omega_1 y_r \\ d\sin\psi_r - \omega_1 x_r \\ \omega_2 - \omega_1 \end{bmatrix} \text{ where } \omega_i = \begin{cases} -1 & \text{if } q = q_2, q_6 \\ 1 & \text{if } q = q_4 \\ 0 & \text{otherwise} \end{cases} \\
\delta(q_1, x, \sigma_1, \epsilon) &= (q_2, x) \\
\delta\left(q_i, \begin{bmatrix} x_r \\ y_r \\ \psi_r \\ \pi/4 \end{bmatrix}, \epsilon, \sigma_2^{i-1}\right) &= \left(q_{i+1}, \begin{bmatrix} x_r \\ y_r \\ \psi_r \\ 0 \end{bmatrix}\right), \text{ for } i = 2, 6 \\
\delta\left(q_i, \begin{bmatrix} x_r \\ y_r \\ \psi_r \\ T \end{bmatrix}, \epsilon, \sigma_2^{i-1}\right) &= \left(q_{i+1}, \begin{bmatrix} x_r \\ y_r \\ \psi_r \\ 0 \end{bmatrix}\right), \text{ for } i = 3, 5
\end{aligned}$$

$$\begin{aligned}
\delta\left(q_1, x, \sigma_1, \epsilon\right) &= \left(q_2, \begin{bmatrix} R(\pi/2)\begin{bmatrix} x_r \\ y_r \end{bmatrix} \\ \psi_r \\ 0 \end{bmatrix}\right) \\
\delta\left(q_2, \begin{bmatrix} x_r \\ y_r \\ \psi_r \\ \pi \end{bmatrix}, \epsilon, \sigma_2\right) &= \left(q_3, \begin{bmatrix} R(\pi/2)\begin{bmatrix} x_r \\ y_r \end{bmatrix} \\ \psi_r \\ 0 \end{bmatrix}\right) \\
\delta(q, x, \sigma_1, \sigma_2) &= \emptyset \text{ otherwise} \\
Inv &= (q_1, X) \cup (q_2, \{x \in X \mid 0 \leq z \leq \pi\}) \cup (q_3, X) \\
I &= (q_1, \{x \in X \mid x_r^2 + y_r^2 > 5^2, z = 0\}) \\
Y &= X \\
h(q, x) &= (q, x) \\
\Omega &= \Box G^c
\end{aligned}$$

It may be easily verified that:

Fact 1 *If* $q \in \{q_1, q_3\}$ *then* $z = 0$.

3.4.2 Seven-Mode Conflict Resolution Example

For the seven-mode conflict resolution example shown in Figure 2.9, the dynamics can be modeled by the automaton of Figure 3.4.

As before, the flight management system of aircraft 1 predicts the velocity of aircraft 2 up to some uncertainty, and computes the parameters σ_1, the relative distance at which the maneuver must start, and T, the time in the "straight2" and "straight4" modes to ensure separation is maintained. The unsafe set G is represented as

$$G = \{q_1, \ldots, q_7\} \times \{x \in X \mid x_r^2 + y_r^2 \leq 5^2\} \tag{3.12}$$

Let ϵ represent *any action* of either aircraft 1 or 2.

$$Q \times X = \{q_1, \ldots, q_7\} \times (\mathbb{R}^2 \times [-\pi, \pi) \times \mathbb{R})$$

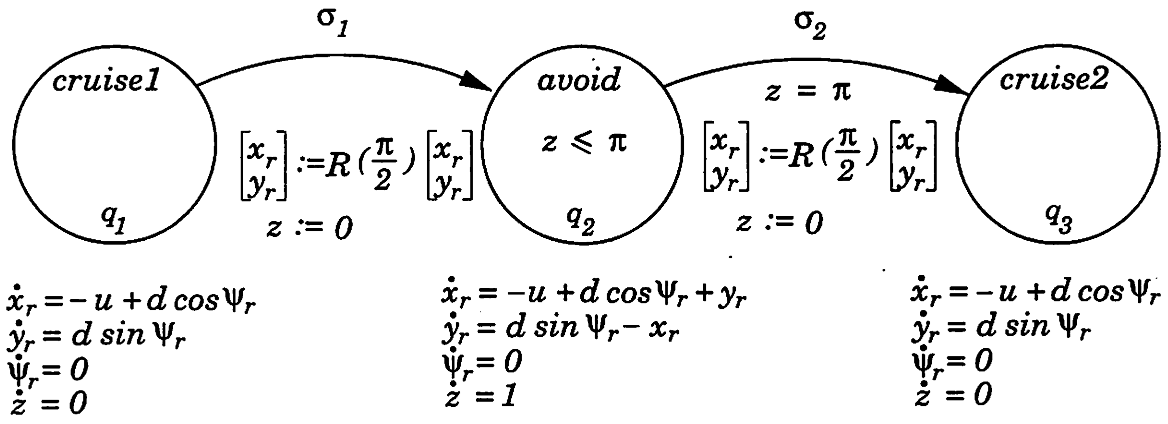

σ_1
σ_2
cruise1
avoid
cruise2
$z = \pi$
$z \leqslant \pi$
$\begin{bmatrix} x_r \\ y_r \end{bmatrix} := R(\frac{\pi}{2}) \begin{bmatrix} x_r \\ y_r \end{bmatrix}$
$z := 0$
$\begin{bmatrix} x_r \\ y_r \end{bmatrix} := R(\frac{\pi}{2}) \begin{bmatrix} x_r \\ y_r \end{bmatrix}$
$z := 0$
q_1
q_2
q_3
$\dot{x}_r = -u + d\cos\Psi_r$
$\dot{y}_r = d\sin\Psi_r$
$\dot{\Psi}_r = 0$
$\dot{z} = 0$
$\dot{x}_r = -u + d\cos\Psi_r + y_r$
$\dot{y}_r = d\sin\Psi_r - x_r$
$\dot{\Psi}_r = 0$
$\dot{z} = 1$
$\dot{x}_r = -u + d\cos\Psi_r$
$\dot{y}_r = d\sin\Psi_r$
$\dot{\Psi}_r = 0$
$\dot{z} = 0$

Figure 3.3: In q_1 the aircraft follow a straight course, in q_2 the aircraft follow a half circle; in q_3 the aircraft return to a straight course.

The control input is defined to be the linear velocity of aircraft 1, $u = v_1 \in U$, and the disturbance input as that of aircraft 2, $d = v_2 \in D$, where U and D denote the range of possible linear velocities of each aircraft. Thus the flight management system of aircraft 1 computes the parameters σ_1, v_1, and the radius of its avoidance maneuver, predicting the velocity of aircraft 2 only up to some uncertainty. Safety is defined in terms of the relative distance between the two aircraft: we define the region at which *loss of separation* occurs as a 5 nautical mile cylinder around the origin in the (x_r, y_r, ψ_r, z) space:

$$G = \{q_1, q_2, q_3\} \times \{x \in X \mid x_r^2 + y_r^2 \leq 5^2\} \tag{3.11}$$

The dynamics of the maneuver can be encoded by the automaton of Figure 3.3. Let ϵ represent *any action* of either aircraft 1 or 2.

$$\begin{aligned}
Q \times X &= \{q_1, q_2, q_3\} \times (\mathbb{R}^2 \times [-\pi, \pi) \times \mathbb{R}) \\
U \times D &= \mathbb{R}^2 \\
\Sigma_1 \times \Sigma_2 &= \{\sigma_1, \epsilon\} \times \{\sigma_2, \epsilon\} \\
f(q, x, u, d) &= \begin{bmatrix} -u + d\cos\psi_r + \omega_1 y_r \\ d\sin\psi_r - \omega_1 x_r \\ \omega_2 - \omega_1 \end{bmatrix} \quad \text{where } \omega_i = \begin{cases} 1 & \text{if } q = q_2 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

We treat the controller synthesis problem as a *dynamic game* between two players, P_1 and P_2. The first player represents the controller and is responsible for choosing u and σ_1, in the face of possible environmental disturbances, which are modeled by d and σ_2. In the following chapters, we use discrete and differential game theory to derive controlled invariant sets and least restrictive control laws which guarantee that the trajectory acceptance condition Ω is always met.

3.4 Examples

We now return to the examples of Section 2.4, to show how they may be modeled using the formalism of this chapter.

3.4.1 Three-Mode Conflict Resolution Example

Consider the three-mode conflict resolution example shown in Figure 2.8, with dynamics in each mode given by equation (2.9), such that in modes 1 and 3, $\omega_1 = \omega_2 = 0$, and in mode 2, $\omega_1 = \omega_2 = 1$.

.The discrete state takes on one of three possible values, $Q = \{q_1, q_2, q_3\}$. The state q_1 corresponds to cruising before the avoid maneuver, q_2 corresponds to the avoid mode and q_3 corresponds to cruising after the avoid maneuver has been completed. There are two discrete actions. The first (σ_1) corresponds to the initiation of the avoid maneuver and can be controlled by choosing the range at which the aircraft start turning. The second transition (σ_2) corresponds to the completion of the avoid maneuver. This transition is required to take place after the aircraft have completed a half circle: the continuous state space is augmented with a timer $z \in \mathbb{R}$ to force this transition. Let $x = (x_r, y_r, \psi_r, z)$.

At each transition, both aircraft change heading instantaneously by $\pi/2$ radians. Because the origin of the relative frame is placed on aircraft 1, meaning that aircraft 1 always has a relative position and heading of $(0,0,0)^T$ in the relative frame, the transitions rotate the state variables (x_r, y_r) by $\pi/2$ radians. We represent this with the standard rotation matrix $R(\pi/2)$.

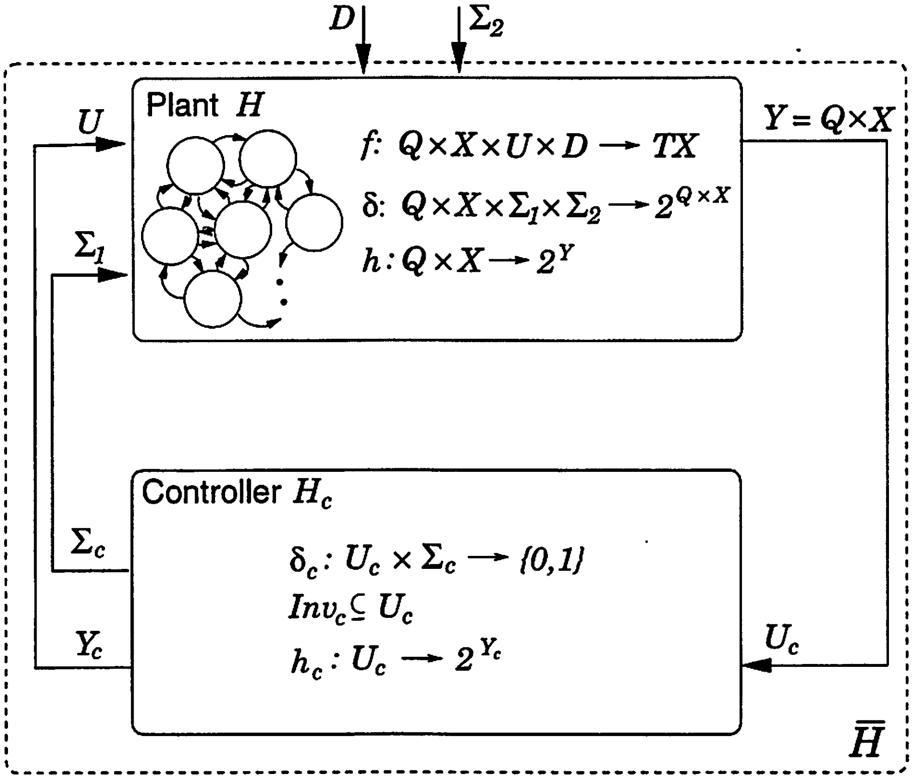

D
Σ_2
Plant H
f: $Q \times X \times U \times D \rightarrow TX$
δ: $Q \times X \times \Sigma_1 \times \Sigma_2 \rightarrow 2^{Q \times X}$
$h: Q \times X \rightarrow 2^Y$
U
Σ_1
$Y = Q \times X$
Controller H_c
$\delta_c: U_c \times \Sigma_c \rightarrow \{0,1\}$
$Inv_c \subseteq U_c$
$h_c: U_c \rightarrow 2^{Y_c}$
Σ_c
Y_c
U_c
$\overline{H}$

Figure 3.2: Composition of Plant H and State-State Feedback Controller H_c to form the controlled system $\overline{H}$. The plant is assumed to be directly observable, as shown.

is defined as

$$H_c = (U_c, \Sigma_c, \delta_c, Inv_c, Y_c, h_c) \tag{3.10}$$

where $U_c = Y$, $\Sigma_c = \Sigma_1$, $\delta_c : U_c \times \Sigma_c \rightarrow \{0,1\}$, $Inv_c \subseteq U_c$, $Y_c = U$, *and* $h_c : U_c \rightarrow 2^{Y_c}$.

We assume that the plant is directly observable, meaning that $Y = Q \times X$, so that the controller has complete access to the state.

The controller generates continuous control inputs $u \in U$ through its output function h_c: since $U_c = Y = Q \times X$, then

$$h_c : Q \times X \rightarrow 2^U$$

Thus the control inputs depend on the state of the plant.

The controllable actions $\sigma_1 \in \Sigma_1$ are generated in the controller via the discrete transition function $\delta_c : Q \times X \times \Sigma_1 \rightarrow \{0,1\}$ and the invariants $Inv_c \subseteq Q \times X$. When $\delta_c(q, x, \sigma_1) = \{1\}$, σ_1 is *enabled*, meaning that it can occur at any time. When $\delta_c(q, x, \sigma_1) = \{0\}$, σ_1 is *disabled* from occurring. Transitions are *forced* to occur through the controller's invariants: σ_1 is forced to occur if it is enabled ($\delta_c(q, x, \sigma_1) = \{1\}$), and if $(q, x) \in Inv_c$.

The composition of plant and controller is shown in Figure 3.2. For slightly more general definitions of H and H_c, it may be shown that the composition of H and H_c is itself a hybrid automaton. Since we will not have occasion to use this generalization, we refer the reader to [20].

Definition 6 (Controlled Invariant Set) *A set of states* $W \subset Q \times X$ *is said to be* **controlled invariant** *if there exists a feedback controller such that if* $I \subseteq W$ *then* $(q(t), x(t)) \in W$ *for all trajectories* $(\tau, q[\cdot], x(\cdot), u(\cdot), d(;), \sigma_1[\cdot], \sigma_2[\cdot])$ *and all* $t \in \tau$ *with* $u(\cdot)$ *and* $\sigma_1[\cdot]$ *generated by the controller.*

A feedback controller, H_c, that renders W invariant is called *least restrictive* if for all $(q, x) \in Q \times X$ all other feedback controllers that render W invariant have output maps, transition relations, and invariants which are contained in the output map h_c, the transition relation δ_c and the invariants Inv_c of H_c.

We assume that f is globally Lipschitz in x and continuous in u and d. Then, by the existence and uniqueness theorem of solutions for ordinary differential equations, for each interval $[\tau_i, \tau_i']$, given the value of $(q[i], x(t))$ for some $t \in [\tau_i, \tau_i']$, and input and disturbance trajectories $u(\cdot), d(\cdot)$ over $[\tau_i, \tau_i']$ there exists a unique solution $x(\cdot)$ over $[\tau_i, \tau_i']$. However, existence and uniqueness of trajectories over $\tau \in \mathcal{T}$ are not guaranteed in general. If the set $\delta(q, x, \sigma_1, \sigma_2)$ were empty for any q, then the hybrid system could deadlock. Also, since the hybrid automaton is non-deterministic, multiple trajectories could occur for the same initial conditions and inputs. Finally, there is no guarantee with the above model that the hybrid automaton is *non-Zeno*, meaning that only a finite number of discrete transitions are allowed in finite time[1]. In fact, one of our air traffic examples is a Zeno hybrid automaton; in Chapter 6 we describe the difficulties in synthesizing controllers for such a system.

Again, we are interested in safety specifications, which translate to trajectory acceptance conditions of the form $\Omega = \Box F$, where F represents a safe subset $F \subseteq Q \times X$ (meaning for all $t \in \tau$ and $i \in \mathbb{Z}$, the state trajectory $(q[i], x(t)) \in F$).

3.3 Controlled Hybrid Systems

Consider a nonlinear hybrid automaton H, with an acceptance condition $\Omega = \Box F$ for $F \subseteq Q \times X$. Do all trajectories of H satisfy Ω? If not, how can we restrict the trajectories of H so that the restricted set satisfies Ω? In this section, we describe the mechanism by which a hybrid automaton may be composed with a controller automaton, so that the result is a hybrid automaton which exhibits the desired behavior.

We refer to the nonlinear hybrid automaton, H, whose behavior we wish to control, as the *plant.* A *controller* may affect the behavior of H though its continuous and discrete control inputs $u \in U$ and $\sigma_1 \in \Sigma_1$:

Definition 5 (Static State Feedback Controller) *A* **static state feedback controller** H_c *of a hybrid automaton* $H = (Q \times X, U \times D, \Sigma_1 \times \Sigma_2, f, \delta, Inv, I, Y, h, \Omega)$

[1] The name "Zeno" comes from the ancient Greek Zeno who lived in Elea, a Greek colony in southern Italy, in the fifth century B.C.. Zeno spent his time posing paradoxes about time.

[74] J. M. Berg, A. Yezzi, and A. R. Tannenbaum. Phase transitions, curve evolution, and the control of semiconductor manufacturing processes. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3376–3381, Kobe, 1996.

[75] A. Puri. *Theory of Hybrid Systems and Discrete Event Systems.* PhD thesis, Department of Electrical Engineering, University of California, Berkeley, 1995.

[76] A. Puri, P. Varaiya, and V. Borkar. ϵ-approximation of differential inclusions. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2892–2897, New Orleans, LA, 1995.

[77] D. P. Bertsekas. *Constraint Optimization and Lagrange Multiplier Methods.* Academic Press, New York, 1982.

[78] E. Polak. *Optimization: Algorithms and Consistent Approximations.* Springer Verlag, New York, 1997.

[79] A. B. Kurzhanski and I. Valyi. *Ellipsoidal calculus for estimation and control.* Birkhauser, Boston, 1997.

[80] L. Vandenberghe, S. Boyd, and S.-P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, 1998.

[81] C. Tomlin, Y. Ma, and S. Sastry. Free flight in 2000: Games on Lie Groups. In *Proceedings of the IEEE Conference on Decision and Control*, Tampa, FL, 1998.

[82] A. Bicchi, A. Marigo, G. Pappas, M. Pardini, G. Parlangeli, C. Tomlin, and S. Sastry. Decentralized air traffic management systems: Performance and fault tolerance. In *Proceedings of the IFAC Workshop on Motion Control*, Grenoble, France, 1998.

[64] H. S. Witsenhausen. A class of hybrid-state continuous time dynamic models. *IEEE Transactions on Automatic Control*, 11(2):161–167, 1966.

[65] S. S. Sastry. Lectures in optimal control and dynamic games. Notes for the course EECS290A, Advanced Topics in Control Theory, University of California, Berkeley, 1996.

[66] M. G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 277(1):1–42, 1983.

[67] M. G. Crandall, L. C. Evans, and P.-L. Lions. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 282(2):487–502, 1984.

[68] R. Montgomery. Abnormal minimizers. *SIAM Journal of Control and Optimization*, 32(6):1605–1620, 1994.

[69] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, 1995.

[70] A. Puri and P. Varaiya. Verification of hybrid systems using abstractions. In *Hybrid Systems II*, number 999 in LNCS. Springer Verlag, 1995.

[71] Adam L. Schwartz. *Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems.* PhD thesis, Department of Electrical Engineering, University of California, Berkeley, 1996.

[72] J. A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science.* Cambridge University Press, New York, 1996.

[73] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[53] R. A. Paielli and H. Erzberger. Conflict probability and estimation for free flight. In *Proceedings of the 35th AIAA Aerospace Sciences Meeting & Exhibit, AIAA 97-0001*, Reno, NV, January 1997.

[54] C. Hynes and L. Sherry. Synthesis from design requirements of a hybrid system for transport aircraft longitudinal control. Preprint, NASA Ames Research Center, Honeywell Air Transport Division, 1996.

[55] C. Tomlin, J. Lygeros, L. Benvenuti, and S. Sastry. Output tracking for a nonminimum phase dynamic CTOL aircraft model. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1867–1872, New Orleans, LA, 1995.

[56] C. Tomlin and S. Sastry. Bounded tracking for nonminimum phase nonlinear systems with fast zero dynamics. *International Journal of Control*, 68(4):819–847, 1997.

[57] A. Church. Logic, arithmetic, and automata. In *Proceedings of the International Congress of Mathematicians*, pages 23–35. 1962.

[58] M. O. Rabin. Automata on infinite objects and Church's problem. In *Regional Conference Series in Mathematics*, 1972.

[59] J. von Neumann and O. Morgenstern. *Theory of games and economic behavior.* Princeton university press, 1947.

[60] P. J. G. Ramadge and W. M. Wonham. The control of discrete event dynamical systems. *Proceedings of the IEEE*, Vol.77(1):81–98, 1989.

[61] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 100–109. IEEE Computer Society Press, 1997.

[62] R. Isaacs. *Differential Games.* John Wiley, 1967.

[63] L. Pontrjagin. Linear differential games. *Soviet Mathematics Doklady*, 8(3):769–771 and 910–912, 1967.

[44] A. Degani. *Modeling Human-Machine Systems: On Modes, Error, and Patterns of Interaction.* PhD thesis, Department of Industrial and Systems Engineering, Georgia Institute of Technology, 1996.

[45] E. Palmer. Oops, it didn't arm - a case study of two automation surprises. In *8th International Symposium on Aviation Psychology*, Columbus, Ohio, 1995.

[46] N. Leveson and E. Palmer. Designing automation to reduce operator errors. In *In the Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, pages 1144–1150, Orlando, FL, 1997.

[47] K. S. Mostov, A. A. Soloviev, and T.-K. Koo. Initial attitude determination and correction of gyro-free INS angular orientation on the basis of GPS linear navigation parameters. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 1034–1039, Boston, MA, 1997.

[48] Radio Technic: ' Commission for Aeronautics. Final report of RTCA Task Force 3: Free flight implementation. Technical report, RTCA, Washington DC, October 1995.

[49] G. Meyer. Design of flight vehicle management systems. Plenary Talk at the IEEE Conference on Decision and Control, 1994.

[50] J. Krozel, T. Mueller, and G. Hunter. Free flight conflict detection and resolution analysis. In *Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA-96-3763*, San Diego, CA, August 1996.

[51] J.-H. Oh and E. Feron. Fast detection and resolution of multiple conflicts for 3-Dimensional free flight. In *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, 1997.

[52] L. Yang and J. Kuchar. Prototype conflict alerting logic for free flight. In *Proceedings of the 35th AIAA Aerospace Sciences Meeting & Exhibit, AIAA 97-0220*, Reno, NV, January 1997.

[33] A. E. Bryson and Y-C. Ho. *Applied Optimal Control.* Blaisdell Publishing Company, Waltham, 1969.

[34] L. C. Young. *Optimal Control Theory.* Cambridge University Press, 1980. 2nd Edition.

[35] W. Fleming and R. Rishel. *Deterministic and Stochastic Optimal Control.* Springer Verlag, 1975.

[36] T. Başar and G. J. Olsder. *Dynamic Non-cooperative Game Theory.* Academic Press, second edition, 1995.

[37] W. M. Wonham. *Linear Multivariable Control: a geometric approach.* Springer Verlag, 1979.

[38] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[39] J. W. Jackson and S. M. Green. Control applications and challenges in air traffic management. In *Proceedings of the American Control Conference*, Philadelphia, PA, 1998.

[40] M. S. Nolan. *Fundamentals of Air Traffic Control.* Wadsworth Inc., 1990.

[41] S. Kahne and I. Frolow. Air traffic management: Evolution with technology. *IEEE Control Systems Magazine*, 16(4):12–21, 1996.

[42] R. Y. Gazit. *Aircraft Surveillance and Collision Avoidance using GPS.* PhD thesis, Department of Aeronautics and Astronautics, Stanford University, 1996.

[43] H. Erzberger, T. J. Davis, and S. Green. Design of Center-TRACON Automation System. In *Proceedings of the AGARD Guidance and Control Syposium on Machine Intelligence in Air Traffic Management*, pages 11.1–11.12, Berlin, Germany, 1993.

[24] T. Dang and O. Maler. Reachability analysis via face lifting. In S. Sastry and T.A. Henzinger, editors, *Hybrid Systems: Computation and Control*, number 1386 in LNCS, pages 96–109. Springer Verlag, 1998.

[25] M.R. Greenstreet and I. Mitchell. Integrating projections. In S. Sastry and T.A. Henzinger, editors, *Hybrid Systems: Computation and Control*, number 1386 in LNCS, pages 159–174. Springer Verlag, 1998.

[26] C. Tomlin, J. Lygeros, and S. Sastry. Synthesizing controllers for nonlinear hybrid systems. In T. Henzinger and S. Sastry, editors, *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science 1386, pages 360–373. Springer Verlag, New York, 1998.

[27] C. Tomlin, J. Lygeros, and S. Sastry. Aerodynamic envelope protection using hybrid control. In *Proceedings of the American Control Conference*, pages 1793–1796, Phildelphia, PA, 1998.

[28] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 1999. To appear.

[29] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A case study in multi-agent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, April 1998.

[30] J. Lygeros, D. N. Godbole, and S. Sastry. Verified hybrid controllers for automated vehicles. *IEEE Transactions on Automatic Control*, 43(4):522–539, April 1998.

[31] J. Lygeros, C. Tomlin, and S. Sastry. On controller synthesis for nonlinear hybrid systems. In *Proceedings of the IEEE Conference on Decision and Control*, Tampa, FL, 1998.

[32] J. R. Büchi and L. H. Landweber. Solving sequential conditions by finite-state operators. In *Proceedings of the American Mathematical Society*, pages 295–311, 1969.

[16] R.W. Brockett. Hybrid models for motion control systems. In H. Trentelman and J.C. Willems, editors, *Perspectives in Control*, pages 29–54. Birkhauser, Boston, 1993.

[17] M. S. Branicky. *Control of Hybrid Systems.* PhD thesis, Department of Electrical Engineering and Computer Sciences, Massachusetts Institute of Technology, 1994.

[18] L. Tavernini. Differential automata and their discrete simulators. *Nonlinear Analysis, Theory, Methods and Applications*, 11(6):665–683, 1987.

[19] A. Deshpande. *Control of Hybrid Systems.* PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 1994.

[20] J. Lygeros. *Hierarchical, Hybrid Control of Large Scale Systems.* PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 1996.

[21] A. Nerode and W. Kohn. Models for hybrid systems: Automata, topologies, controllability, observability. In Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors, *Hybrid System*, Lecture Notes in Computer Science 736, pages 317–356. Springer Verlag, New York, 1993.

[22] M. Lemmon, J. A. Stiver, and P. J. Antsaklis. Event identification and intelligent hybrid control. In Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors, *Hybrid Systems*, Lecture Notes in Computer Science 736, pages 268–296. Springer Verlag, New York, 1993.

[23] M. Heymann, F. Lin, and G. Meyer. Control synthesis for a class of hybrid systems subject to configuration-based safety constraints. In O. Maler, editor, *Hybrid and Real Time Systems*, Lecture Notes in Computer Science 1201, pages 376–391. Springer Verlag, 1997.

[8] N. Lynch, R. Segala, F. Vaandrager, and H.B. Weinberg. Hybrid I/O automata. In *Hybrid Systems III*, Lecture Notes in Computer Science 1066, pages 496–510. Springer Verlag, 1996.

[9] T.A. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, pages 278–292. IEEE Computer Society Press, 1996.

[10] A. Puri and P. Varaiya. Decidability of of hybrid systems with rectangular differential inclusions. In *CAV94: Computer-Aided Verification*, Lecture Notes in Computer Science 818, pages 95–104. Springer Verlag, Stanford, CA, 1995.

[11] O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of discrete controllers for timed systems. In Ernst W. Mayr and Claude Puech, editors, *STACS 95: Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science 900, pages 229–242. Springer Verlag, Munich, 1995.

[12] H. Wong-Toi. The synthesis of controllers for linear hybrid automata. In *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, 1997.

[13] T. A. Henzinger, P. H. Ho, and H. Wong-Toi. A user guide to HYTECH. In E. Brinksma, W. Cleaveland, K. Larsen, T. Margaria, and B. Steffen, editors, *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science 1019, pages 41–71. Springer Verlag, 1995.

[14] C. Daws, A. Olivero, S. Tripakis, and S. Yovine. The tool KRONOS. In *Hybrid Systems III, Verification and Control*, pages 208–219. Lecture Notes in Computer Science 1066, Springer Verlag, 1996.

[15] N. Bjorner, A. Browne, E. Chang, M. Colon, A. Kapur, Z. Manna, H. Sipma, and T. Uribe. STeP: The Stanford Temporal Prover (educational release), user's manual. Technical report, STAN-CS-TR-95-1562, Department of Computer Science, Stanford University, 1995.

Bibliography

[1] Honeywell Inc. Markets Report. Technical Report NASA Contract NAS2-114279, Final Report for AATT Contract, 1996.

[2] T. S. Perry. In search of the future of air traffic control. *IEEE Spectrum*, 34(8):18–35, 1997.

[3] Radio Technical Commission for Aeronautics. Minimum aviation system performance standards for Automatic Dependent Surveillance-Broadcast (ADS-B). Technical report, RTCA-186, February 1997. DRAFT 4.0.

[4] Honeywell Inc. Concepts and Air Transportation Systems Report. Technical Report NASA Contract NAS2-114279, Final Report for AATT Contract, 1996.

[5] C. Tomlin, G. Pappas, J. Košecká, J. Lygeros, and S. Sastry. Advanced air traffic automation: a case study in distributed decentralized control. In B. Siciliano and K. Valavanis, editors, *Control Problems in Robotics and Automation*, pages 261–295. Springer Verlag, 1997.

[6] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[7] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors, *Hybrid Systems*, pages 366–392. Springer Verlag, New York, 1993.

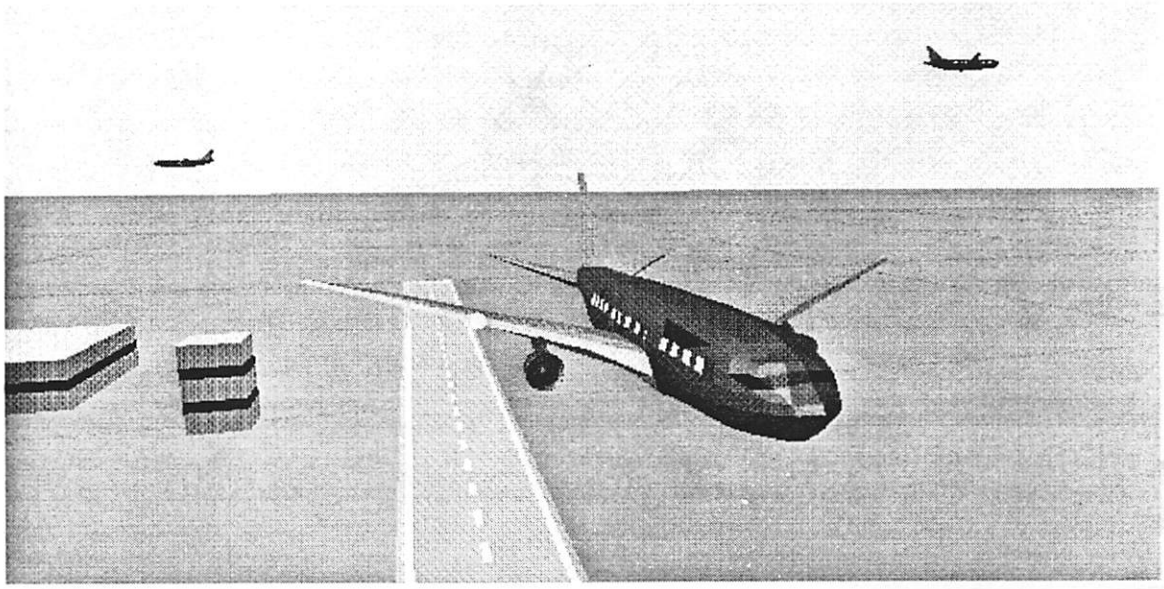


Figure 8.2: Airspace simulation tool, incorporating dynamic models of aircraft in an interactive environment.

elevator as inputs. Figure 8.2 displays a view of three aircraft, on a prototype version of our software tool, called SmartPlanesII.

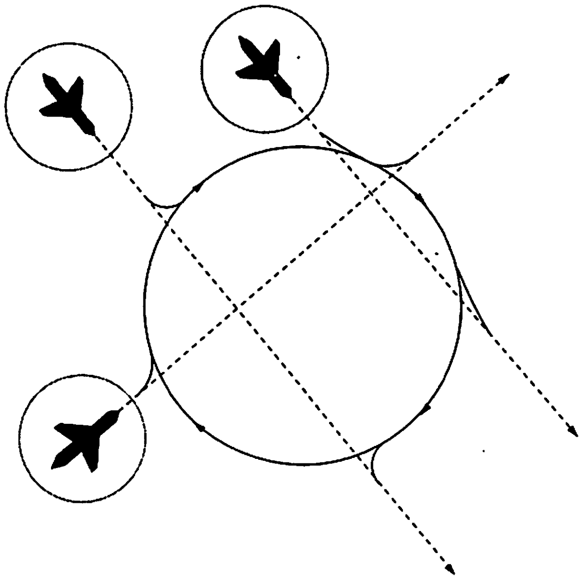


Figure 8.1: Conflict resolution for three aircraft: the *roundabout* maneuver.

aircraft has access to different amounts of information about the system as a whole. In this work, we generate protocols for maneuvers using Reeds-Shepp paths from robotic motion planning [82].

Future Directions in Air Traffic Management

Part of our research program is to build a *design and simulation platform* on a network of workstations to be used as a testbed for these algorithms. The platform will include dynamic models of different aircraft and their autopilots, a hybrid system modeling and simulation tool, as well as the standard computation tools of Matlab and Mathematica. It will provide an environment in which different concepts for a new air traffic management system can be tested, and it will be set up hierarchically so that the user will be able to implement "macroscopic" algorithms on a complete air traffic system at the same time as "microscopic" algorithms on each individual aircraft. Realistic simulation is the first step to successful implementation of the algorithms in future air traffic systems. A first version of this simulation tool has been developed for three dimensional dynamic models of aircraft with throttle, aileron, rudder, and hybrid system model and controller synthesis scheme presents a method by which the system design process may be automated (which is important for inexpensive and rapid prototyping of real-time software systems), and by which one may achieve better performance, handle larger systems, and have greater confidence that the system will work as planned. While there would still be need for simulation and testing, as we discuss below, one should not have to rely on these methods.

Future Directions in Controller Synthesis

The emphasis in this dissertation has been on developing least restrictive control laws for safety critical systems: our controllers restrict actions of the system only when the state approaches unsafe regions of operation, and all possible mode switches which satisfy the safety constraints are derived. Our original motivation for studying flight modes and the switching between them was in work with Meyer [49] and Hynes [54], in which control schemes were sought which guided an aircraft safely through "optimal" sequences of flight modes. Our current work focuses on designing specific control laws for each mode, which may be integrated with the constraints for safety to provide bumpless transfer between modes.

We are also exploring different techniques to simplify the computation of optimal control and disturbance trajectories $(u^*(\cdot), d^*(\cdot))$. One promising technique comes from exploiting special geometric properties of the continuous dynamics. In [81], we describe the dynamical games solution when the underlying dynamics correspond to left-invariant control systems on a Lie group. In this formulation, some simplification in the derivation of saddle and Nash strategies follows from the use of Marsden-Weinstein reduction techniques: we give an outline for the solution of N-aircraft conflict resolution using Nash type strategies. This simplification allows us to efficiently compute optimal solutions to complex conflict resolution problems for more than 2 aircraft, using numerical techniques which could be programmed into the flight management computers on board each aircraft. For three aircraft coming into conflict this approach produces the *roundabout* maneuver, shown in Figure 8.1.

We have also begun work on investigating the optimal trajectories when each

Chapter 8

Future Work

In this dissertation we have presented a model for hybrid systems, and an algorithm, based on two-player zero-sum game theory for automata and continuous systems, for computing reachable sets and for synthesizing control schemes for hybrid systems. The reachable set calculation is *exact*: the solution to the coupled Hamilton-Jacobi equations in Chapter 5 describes the reachable set for nonlinear hybrid systems of any continuous and discrete state space dimension. One of the main topics of our current and future work is the numerical computation and approximation of reachable sets. In Chapter 7 we presented our initial results in using a level set method for computing solutions to the Hamilton-Jacobi equation, and our future plans include developing connections to polygonal and ellipsoidal methods to approximate and efficiently store reachable sets.

Our goal is to develop a *software tool* to perform these calculations automatically or semi-automatically for hybrid systems with an arbitrary number of discrete states, and an arbitrary continuous state dimension. In this tool, the user will specify the system model and the desired property to verify, and the tool will verify that either the safety property is maintained by the system, or will provide a trace to the user as to why the system fails the safety test.

The second goal of this dissertation is to present a very rich application domain in air traffic systems. The control used in these systems today is either manual, by a human controller, or by automation which has not been formally verified. Our

Ellipsoidal Methods

A similar idea is to use ellipsoids as inner and outer approximations to the reach set [79], [80]. [80] presents efficient algorithms for calculating both the minimum volume ellipsoid containing given points, and the maximum volume ellipsoid in a polyhedron, using matrix determinant maximization subject to linear matrix inequality constraints.

used to approximate the differential inclusion is known to be decidable, thus one can guarantee that the reachable set as $t \to -\infty$ can be computed in a finite number of steps. The amount of preprocessing required to initially approximate the dynamics may be quite formidable however, especially to achieve a close approximation of the true reach set.

7.2.2 Approximating non-smooth sets with smooth sets

We have shown that the reach set at any time $t \in (-\infty, 0]$ may have a non-smooth boundary due to switches in (u^*, d^*), non-smooth initial data, or the formation of shocks. The level set scheme propagates these discontinuities, yet its implementation may require a very small time step to do this accurately. In [31] we present a method for over-approximating such non-smooth sets with sets for which the boundary is continuously differentiable. Suppose that there exist differentiable functions l_G^i, $i = 1, \ldots, k$ such that

$$G = \{x \in X \mid \forall i \in \{1, \ldots, k\},\ l_G^i(x) \leq 0\} \tag{7.18}$$

Following [77, 78] we define two smooth functions:

$$\begin{aligned} G^\epsilon(x) &= \epsilon \ln \left[\Sigma_{i=1}^k e^{l_G^i(x)/\epsilon}\right] \\ G_\epsilon(x) &= G^\epsilon(x) - \epsilon \ln k \end{aligned}$$

Now defining

$$\begin{aligned} G_\epsilon &= \{x \in X \mid G_\epsilon(x) \leq 0\} \\ G^\epsilon &= \{x \in X \mid G^\epsilon(x) \leq 0\} \end{aligned}$$

it is easy to show that $G_\epsilon \subseteq G \subseteq G^\epsilon$ and we can prove that $\lim_{\epsilon \to 0} G_\epsilon = G$, and $\lim_{\epsilon \to 0} G^\epsilon = G$. By applying Algorithm (5.6) to smooth inner and outer approximations of the sets G and E, we calculate smooth inner and outer approximations to the true reach set.

Figure 7.1 displays the result of applying this algorithm to the two-aircraft example with zero angular velocity and $[\underline{v}_1, \overline{v}_1] = [2, 4]$, $[\underline{v}_2, \overline{v}_2] = [1, 5]$ and $\psi_r = 2\pi/3$ (Figure 6.10).

This example presents the very basic idea in level set methods; for special forms of the Hamilton-Jacobi equation, many extremely efficient variants of this method exist [72]. In particular, the *narrow band* and *fast marching* methods speed up the algorithm by confining the computation to a narrow band around the evolving front.

It is essential that a bound on the error due to approximation be known at each step of the algorithm, in order to guarantee that the computed surface is a conservative approximation to the actual surface.

7.2 Other Methods

Other methods have been presented for approximating the reach set calculation: here we discuss twc .nethods, one which approximates the continuous dynamic equations with simpler equations, and one which approximates the reach set itself.

7.2.1 Approximating Dynamics with Differential Inclusions

Suppose the continuous dynamics in the nonlinear hybrid automaton (3.6) were approximated with the differential inclusion

$$\dot{x} \in g(q, x) \tag{7.16}$$

where $g(q,x) = \{f(q,x,u,d) \mid \forall u \in U, d \in D\}$. A computationally efficient method for approximating the reach set of $g(q,x)$ is to conservatively approximate $g(q,x)$ by a set of constant inclusions, each of the form

$$\dot{x} \in [g_{\min}, g_{\max}] \tag{7.17}$$

and then to compute the reach set of the constant inclusions. This method is presented in [75], [76] where it is proved that the approximation error can be made arbitrarily small by approximating the differential inclusion arbitrarily closely (ϵ-approximation). An advantage of this method is that the class of constant inclusions

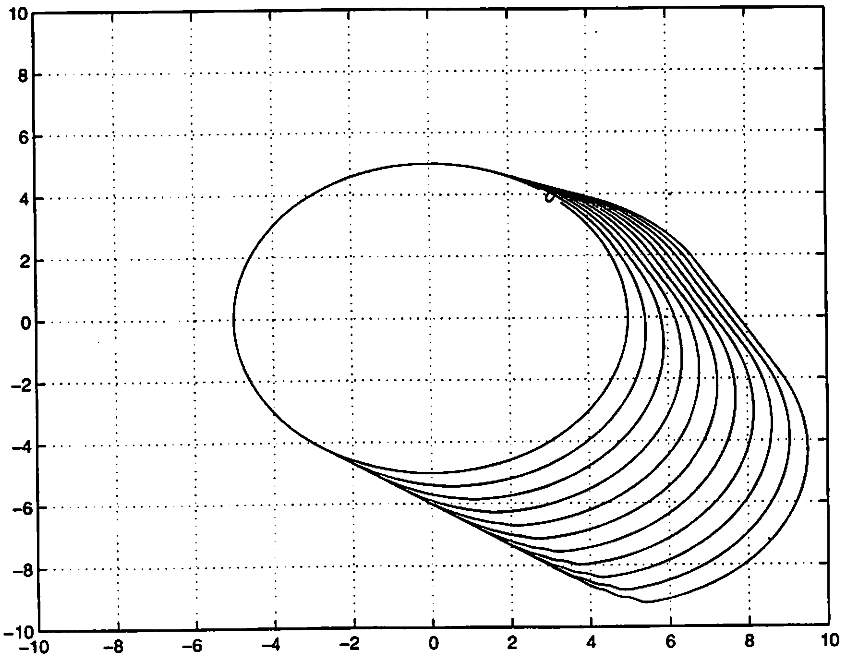


Figure 7.1: $\{x \in X \mid \tilde{J}^*(x,t) \leq 0\}$ shown in the (x_r, y_r)-plane for $[\underline{v}_1, \overline{v}_1] = [2,4]$, $[\underline{v}_2, \overline{v}_2] = [1,5]$ and $\psi_r = 2\pi/3$.

1. Compute

$$u^*(x_{ij}, \frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x_1}, \frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x_2}) \tag{7.7}$$

$$d^*(x_{ij}, \frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x_1}, \frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x_2}) \tag{7.8}$$

using the initial approximations to the derivatives

$$\frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x_1} = D^{0x_1}, \quad \frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x_2} = D^{0x_2} \tag{7.9}$$

2. Calculate $f(x_{ij}, u^*, d^*)$

3. If $-f(x_{ij}, u^*, d^*)$ flows from larger to smaller values of x_1, let

$$\frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x_1} = D^{+x_1} \tag{7.10}$$

otherwise let

$$\frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x_1} = D^{-x_1} \tag{7.11}$$

If $-f(x_{ij}, u^*, d^*)$ flows from larger to smaller values of x_2, let

$$\frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x_2} = D^{+x_2} \tag{7.12}$$

otherwise let

$$\frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x_2} = D^{-x_2} \tag{7.13}$$

4. Compute $\tilde{J}^*(x_{ij}, t - \Delta t)$:

For x_{ij} such that $J^*(\tilde{x}_{ij}, t) > 0$,

$$\tilde{J}^*(x_{ij}, t - \Delta t) = \tilde{J}^*(x_{ij}, t) + \Delta t \frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x} f(x_{ij}, u^*, d^*) \tag{7.14}$$

For x_{ij} such that $J^*(\tilde{x}_{ij}, t) \leq 0$,

$$\tilde{J}^*(x_{ij}, t - \Delta t) = \begin{cases} \tilde{J}^*(x_{ij}, t) + \Delta t \frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x} f(x_{ij}, u^*, d^*) \\ \quad \text{if } \frac{\partial \tilde{J}^*(x_{ij},t)}{\partial x} f(x_{ij}, u^*, d^*) < 0 \\ \tilde{J}^*(x_{ij}, t) \text{ otherwise} \end{cases} \tag{7.15}$$

derivative must be used. Consider an example in two dimensions, with X discretized into a grid with spacing Δx_1 and Δx_2. The *forward difference operator* D^{+x_i} at $x = (x_1, x_2)$ is defined as:

$$D^{+x_1}J^*(x,t) = \frac{J^*((x_1+\Delta x_1, x_2), t) - J^*(x,t)}{\Delta x_1} \tag{7.1}$$

$$D^{+x_2}J^*(x,t) = \frac{J^*((x_1, x_2+\Delta x_2), t) - J^*(x,t)}{\Delta x_2} \tag{7.2}$$

The *backward difference operator* D^{-x_i} is defined as

$$D^{-x_1}J^*(x,t) = \frac{J^*(x,t) - J^*((x_1-\Delta x_1, x_2), t)}{\Delta x_1}. \tag{7.3}$$

$$D^{-x_2}J^*(x,t) = \frac{J^*(x,t) - J^*((x_1, x_2-\Delta x_2), t)}{\Delta x_2} \tag{7.4}$$

The *central difference operator* D^{0x_i} is defined as

$$D^{0x_1}J^*(x,t) = \frac{J^*((x_1+\Delta x_1, x_2), t) - J^*((x_1-\Delta x_1, x_2), t)}{2\Delta x_1} \tag{7.5}$$

$$D^{0x_2}J^*(x,t) = \frac{J^*((x_1, x_2+\Delta x_2), t) - J^*((x_1, x_2-\Delta x_2), t)}{2\Delta x_2} \tag{7.6}$$

At each grid point $x = (x_1, x_2)$, the partial derivatives $\frac{\partial J^*(x,t)}{\partial x_1}$ and $\frac{\partial J^*(x,t)}{\partial x_2}$ may be approximated to first order using either the forward, backward, or central difference operators. The correct choice of operator depends on the direction of $f(x, u^*, d^*)$ (in our case it depends on $-f(x, u^*, d^*)$ since we compute backwards in time). If $-f(x, u^*, d^*)$ flows from left to right (from smaller to larger values of x_1), then D^{-x_1} should be used to approximate $\frac{\partial J^*(x,t)}{\partial x_1}$ (and vice versa); and if $-f(x, u^*, d^*)$ flows from bottom to top (from smaller to larger values of x_2), then D^{-x_2} should be used to approximate $\frac{\partial J^*(x,t)}{\partial x_2}$ (and vice versa). Such an approximation is called an *upwind* scheme, since it uses information upwind of the direction that information propagates.

The algorithm for the two dimensional example proceeds as follows. Choose a domain of interest in X and discretize the domain with a grid of spacing $\Delta x_1, \Delta x_2$. Let x_{ij} represent the grid point $(i\Delta x_1, j\Delta x_2)$ and let $\tilde{J}^*(x_{ij}, t)$ represent the numerical approximation of $J^*(x_{ij}, t)$. Using the boundary condition $J^*(x, 0) = l(x)$, compute $\tilde{J}^*(x_{ij}, 0)$ for each x_{ij}.

Let $t = 0$.

While $\tilde{J}^*(x_{ij}, t) \neq \tilde{J}^*(x_{ij}, t - \Delta t)$ perform the following steps:

7.1 A Level Set Method for Boundary Approximation

Consider the Hamilton-Jacobi equation (4.37), repeated here:

$$-\frac{\partial J^*(x,t)}{\partial t} = \begin{cases} H^*(x, \frac{\partial J^*(x,t)}{\partial x}) & \text{for } \{x \in X \mid J^*(x,t) > 0\} \\ \min\{0, H^*(x, \frac{\partial J^*(x,t)}{\partial x})\} & \text{for } \{x \in X \mid J^*(x,t) \leq 0\} \end{cases}$$

with boundary condition $J^*(x,0) = l(x)$. Recall from the discussion in Chapter 4 that we define a *viscosity solution* [66, 67] to (4.37) as the solution as $\epsilon \rightarrow 0$ of the partial differential equation (4.39):

$$-\frac{\partial J_\epsilon^*(x,t)}{\partial t} = \begin{cases} H^*(x, \frac{\partial J_\epsilon^*(x,t)}{\partial x}) + \epsilon \Delta J_\epsilon^*(x,t) & \text{for } \{x \in X \mid J_\epsilon^*(x,t) > 0\} \\ \min\{0, H^*(x, \frac{\partial J_\epsilon^*(x,t)}{\partial x})\} + \epsilon \Delta J_\epsilon^*(x,t) & \text{for } \{x \in X \mid J_\epsilon^*(x,t) \leq 0\} \end{cases}$$

with boundary condition $J_\epsilon^*(x,0) = l_\epsilon(x)$.

The level set methods of Osher and Sethian [38] ([72] provides a comprehensive survey) is a set of computation schemes for propagating interfaces in which the speed of propagation is governed by a partial differential equation. These numerical techniques compute the viscosity solution to the Hamilton-Jacobi partial differential equation, ensuring that shocks are preserved. The methods have proved fruitful in many applications, including shape recovery problems in computer vision [73], and plasma etching problems in micro chip fabrication [74].

The key idea of the level set method is to embed the curve or surface to be evolved, for example the n-dimensional boundary of the capture set, as the zero level set of a function in $n+1$-dimensional space. The advantage of this formulation is that the $n+1$-dimensional function always remains a function as long as its speed of propagation is smooth, while the n-dimensional boundary may develop shocks or change topology under this evolution. The numerical methods of [72] choose the solution of (4.37) (with zero viscosity) to be the one obtained from (4.39) as the viscosity coefficient ϵ vanishes. We present an outline of the method below for a two-dimensional example.

In order for the numerical scheme to closely approximate the gradient $\frac{\partial J^*(x,t)}{\partial x}$, especially at points of discontinuity, an appropriate approximation to the spatial

Chapter 7

Computing Boundaries of Safe Sets

In practice, the usefulness of the algorithm for hybrid controller synthesis depends on our ability to efficiently compute the optimal control and disturbance trajectories $(u^*(\cdot), d^*(\cdot))$, as well as solutions to the Hamilton-Jacobi partial differential equation (4.37). As discussed in Chapter 4, numerical solutions are potentially complicated by the facts that the right hand side of (4.37) is non-smooth and that the initial data F may have non-smooth boundary, that $(u^*(\cdot), d^*(\cdot))$ may be discontinuous, and that the solution $J^*(x,t)$ may develop shocks over time. New optimal control tools [71] can make the solution of computing $(u^*(\cdot), d^*(\cdot))$ feasible, at least numerically, and in this section, we discuss a numerical technique developed by Osher and Sethian which computes the viscosity solution to the Hamilton-Jacobi equation, ensuring that discontinuities are preserved. We present the results of applying this technique to the two-aircraft example.

Similarly, $W^{-2} = F$, $W^{-3} = F$, and the fixed point is $W^* = W^0$, meaning that the maximal controlled invariant set contained in F is F itself! This is clearly incorrect for the real system: the calculations to produce Figures 6.14 and 6.16 in the previous section showed that certain "corners" of F are not controlled invariant. The error lies in the fact that this system is Zeno: if forced into one of these corners, the system could avoid flowing out of F by switching infinitely often in zero time between discrete states. Unlike the previous examples, there is no specified minimum time for the system to stay in each discrete state.

A possible remedy is to enforce that the system remain in each discrete state for some minimum time $T > 0$. If this is the case, then the algorithm calculates W^* as the union of $W^*_{hV\dot{h}}$ and $W^*_{V\gamma}$ for their applicable discrete modes. The mode switching logic is implicit in these calculations: as the aircraft approaches maximum or minimum altitude, the FMS must force the autopilot to switch to modes 4 or 5 and choose a control scheme which satisfies the limits on $\ddot{h}$. As the aircraft approaches its maximum or minimum speed and flight path angle, the FMS must force the system into modes 1, 2 or 3 and select those control inputs which either drive the aircraft back inside the envelope, or keep it on the boundary of the envelope.

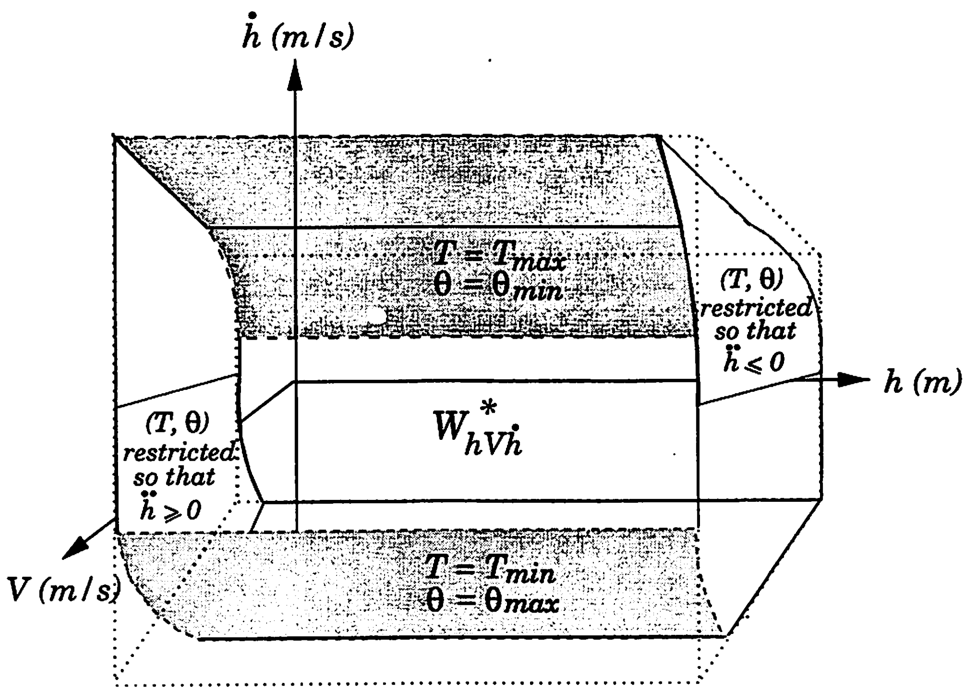


Figure 6.16: The set $W^*_{hV\dot{h}}$ in $(h, V, \dot{h})$-space, with control law as indicated. Altitudes are $h_{min} = 10kft$, $h_{max} = 51kft$.

scheme is as indicated. This calculation incorporates the limits on the altitude h into the previous calculation: at $h = h_{max}$, the control must be chosen so that $\ddot{h} \leq 0$, whereas at $h = h_{min}$, the control is restricted to force $\ddot{h} \geq 0$.

6.2.2 Controller Synthesis

We would now like to apply Algorithm (5.6) to generate the controllable actions σ_1^{ij} which force transitions between discrete states to ensure safety. However, we quickly run into a problem. At the first step of the algorithm, $W^0 = F$, and since there are no uncontrollable actions, $Pre_2(F) = F^c$. However, since the controllable actions are always enabled, $Pre_1(F) = F$. Thus

$$Reach(Pre_2(F), Pre_1(F)) = F^c \tag{6.50}$$

and therefore

$$W^{-1} = F \backslash F^c = F \tag{6.51}$$

Additional Constraints for Passenger Comfort

Cost functions involving the linear and angular accelerations can be used to encode the requirement for passenger comfort (we use J_5, J_6 in the following, after J_1 to J_4 of the previous section):

$$J_5((V,\gamma),u(\cdot),t)) = -\max_{t\geq 0}|\dot{V}(t)|, \quad J_6((V,\gamma),u(\cdot),t)) = -\max_{t\geq 0}|V(t)\dot{\gamma}(t)| \tag{6.45}$$

The requirement that the linear and angular accelerations remain within the limits determined for comfortable travel are encoded by thresholds:

$$J_5((V,\gamma),u(\cdot),t)) \geq -0.1g, \quad J_6((V,\gamma),u(\cdot),t)) \geq -0.1g \tag{6.46}$$

Within the class of safe controls, a control scheme which addresses the passenger comfort requirement can be constructed. To do this, we solve the optimal control problem:

$$J_5^*((V,\gamma)) = \max_{u(\cdot)\in g(V,\gamma)} J_5, \quad J_6^*((V,\gamma)) = \max_{u(\cdot)\in g(V,\gamma)} J_6 \tag{6.47}$$

From this calculation, it is straightforward to determine the set of "comfortable" states:

$$\{(V,\gamma) \in W_{V\gamma}^* \mid J_5^*(V,\gamma) \geq -0.1g \wedge J_6^*(V,\gamma) \geq -0.1g\} \tag{6.48}$$

The set of comfortable controls may be calculated by substituting the bounds on the accelerations into equation (2.14), (2.15) to get

$$\begin{array}{rcccl} -0.1Mg + a_D V^2 + Mg\sin\gamma & \leq & T & \leq & 0.1Mg + a_D V^2 + Mg\sin\gamma \\ -\frac{0.1Mg}{a_L V^2 c} - \frac{1-c\gamma}{c} + \frac{Mg\cos\gamma}{a_L V^2 c} & \leq & \theta & \leq & \frac{0.1Mg}{a_L V^2 c} - \frac{1-c\gamma}{c} + \frac{Mg\cos\gamma}{a_L V^2 c} \end{array} \tag{6.49}$$

These constraints provide lower and upper bounds on the thrust and the pitch angle which may be applied at any point (V,γ) in $W_{V\gamma}^*$ while maintaining comfort.

Speed, Altitude Modes

Repeating these calculations for the speed and altitude modes (modes $4, 5$), using the dynamics (2.13) and envelope illustrated in Figure 2.11(b), the controlled invariant subset $W_{hV\dot{h}}^*$ is computed and shown in Figure 6.16, and the least restrictive control

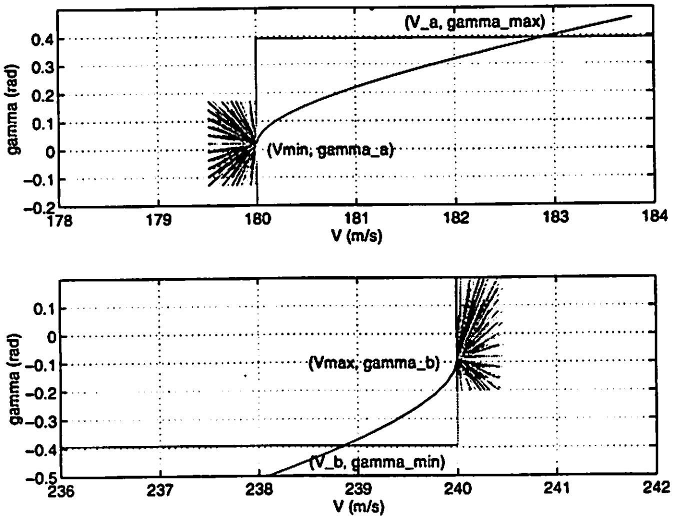


Figure 6.15: Upper left boundary and lower right boundary of $F_{V\gamma}$.

either tangent to or to the left side of the boundary $\{(V,\gamma) \mid (V = V_{max}) \wedge (\gamma_b \leq \gamma \leq \gamma_{max})\}$. At the point (V_{max}, γ_b), where $T_b(\gamma_b) = T_{min}$, T_{min} is the unique thrust which keeps the system trajectory tangent to $F_{V\gamma}$ (lower right boundary of $F_{V\gamma}$ in Figure 6.15).

Similar calculations along the upper and lower sides of $\partial F_{V\gamma}$ yield that the values of θ for which the vector field becomes tangent to $\partial F_{V\gamma}$ are $\theta_c(V)$ and $\theta_d(V)$ of equations (6.43) and (6.44). ■

In Figure 6.14, the portions of $W^*_{V\gamma}$ for which all control inputs are safe ($g(V,\gamma) = U$) are indicated with solid lines; those for which only a subset are safe ($g(V,\gamma) \subset U$) are indicated with dashed lines. The map defines the *least restrictive safe control scheme* and determines the mode switching logic. On ∂J^a and ∂J^b, the system must be in **Mode 2** or **Mode 3**. Anywhere else in $W^*_{V\gamma}$, any of the three modes is valid as long as the input constraints of equation (6.40) are satisfied. In the regions $F_{V\gamma} \backslash W^*_{V\gamma}$ (the upper left and lower right corners of $F_{V\gamma}$), no control inputs will keep the system inside of $F_{V\gamma}$.

$g(V,\gamma) = U \cap \hat{g}(V,\gamma)$, *where:*

$$\hat{g}(V,\gamma) = \{ \begin{array}{ll} \emptyset & \textit{if } (V,\gamma) \in (W^*_{V\gamma})^c \\ T \geq T_a(\gamma) & \textit{if } (V = V_{min}) \wedge (\gamma_{min} \leq \gamma \leq \gamma_a) \\ \theta = \theta_{min} \wedge T = T_{max} & \textit{if } (V,\gamma) \in \partial J^a \\ \theta \leq \theta_c(V) & \textit{if } (\gamma = \gamma_{max}) \wedge (V_a \leq V \leq V_{max}) \\ T \leq T_b(\gamma) & \textit{if } (V = V_{max}) \wedge (\gamma_b \leq \gamma \leq \gamma_{max}) \\ \theta = \theta_{max} \wedge T = T_{min} & \textit{if } (V,\gamma) \in \partial J^b \\ \theta \geq \theta_d(V) & \textit{if } (\gamma = \gamma_{min}) \wedge (V_{min} \leq V \leq V_b)\} \end{array} \tag{6.40}$$

with

$$T_a(\gamma) = a_D V_{min}^2 + Mg\sin\gamma \tag{6.41}$$

$$T_b(\gamma) = a_D V_{max}^2 + Mg\sin\gamma \tag{6.42}$$

$$\theta_c(V) = \frac{M}{a_L V c}\left(\frac{g\cos\gamma_{max}}{V} - \frac{a_L V(1 - c\gamma_{max})}{M}\right) \tag{6.43}$$

$$\theta_d(V) = \frac{M}{a_L V c}\left(\frac{g\cos\gamma_{min}}{V} - \frac{a_L V(1 - c\gamma_{min})}{M}\right) \tag{6.44}$$

Proof: Consider the set $\{(V,\gamma) \mid (V = V_{min}) \wedge (\gamma_{min} \leq \gamma \leq \gamma_a)\}$. For each (V,γ) in this set, denote by $(T_a(\gamma), \theta_a(\gamma))$ the values of (T,θ) for which the vector field $(\dot{V},\dot{\gamma})$ becomes tangent to this set. These are the (T,θ) for which $\dot{V} = 0$: setting $\dot{V} = 0$ leads to equation (6.41) for all $\theta_a(\gamma) \in [\theta_{min}, \theta_{max}]$. Thus, $\{[T_a(\gamma), T_{max}] \times [\theta_{min}, \theta_{max}]\} \subseteq U$ keeps the system either tangent to or to the right side of the boundary $\{(V,\gamma) \mid (V = V_{min}) \wedge (\gamma_{min} \leq \gamma \leq \gamma_a)\}$. At the point (V_{min}, γ_a'), where $T_a(\gamma_a') = T_{min}$ the vector field cone $(\dot{V},\dot{\gamma})$ for $(T,\theta) \in U$ points completely inside $F_{V\gamma}$. At γ_a, the cone points completely outside $F_{V\gamma}$, and $T = T_{max}$ is the unique value of throttle which keeps the system trajectory $(V(t), \gamma(t))$ tangent to $F_{V\gamma}$. This is illustrated in Figure 6.15, which shows the upper left boundary of $F_{V\gamma}$, and the cone of controls at the point (V_{min}, γ_a).

The calculation may be repeated for the set $\{(V,\gamma) \mid (V = V_{max}) \wedge (\gamma_b \leq \gamma \leq \gamma_{max})\}$. Here, denote by $(T_b(\gamma), \theta_b(\gamma))$ the values of (T,θ) for which the vector field $(\dot{V},\dot{\gamma})$ becomes tangent to this set. Setting $\dot{V} = 0$ leads to equation (6.42) for all $\theta_b(\gamma) \in [\theta_{min}, \theta_{max}]$. Therefore, $\{[T_{min}, T_b(\gamma)] \times [\theta_{min}, \theta_{max}]\} \subseteq U$ keeps the system is tangent to $\{(V,\gamma) \mid J_1^*(V,\gamma) = 0\}$. Thus the solution $(V(t),\gamma(t))$ to equations (2.14), (2.15) with $u = (T_{max}, \theta_{min})$ evolves along $J_1^*(V,\gamma) = 0$. Since, by construction, $(V,\gamma) \in \partial J^a$ satisfies equations (2.14), (2.15) with $u = (T_{max}, \theta_{min})$, then $(V,\gamma) \in \partial J^a$ satisfies $J_1^*(V,\gamma) = 0$.

Repeating this analysis for $\{(V,\gamma) \mid l_3(V,\gamma) = 0\}$, we can show that

$$\{(V,\gamma) \mid J_3^*(V,\gamma) = 0\} = \{(V,\gamma) \mid (V = V_{max}) \wedge (\gamma_b \leq \gamma \leq \gamma_{max})\} \cup \{(V,\gamma) \in \partial J^b\} \tag{6.39}$$

On the remaining boundaries, $H_2^*((V,\gamma),p) > 0$ and $H_4^*((V,\gamma),p) > 0$, so these boundaries remain unchanged under the evolution of their respective Hamilton-Jacobi equations.

It remains to prove that $W_{V\gamma}^* = \cap_{i\in\{1,2,3,4\}}\{(V,\gamma) \mid J_i^*(V,\gamma) \geq 0\}$. Clearly, any state (V,γ) for which there exists an i such that $J_i^*(V,\gamma) < 0$ must be excluded from $W_{V\gamma}^*$, since a trajectory exists which starts from this state and drives the system out of $\cap_{i\in\{1,2,3,4\}}\{(V,\gamma) \mid J_i^*(V,\gamma) \geq 0\}$. Thus $W_{V\gamma}^* \subset \cap_{i\in\{1,2,3,4\}}\{(V,\gamma) \mid J_i^*(V,\gamma) \geq 0\}$. To prove equality, we need only show that at the points of intersection of the four boundaries: $\{(V_a,\gamma_{max}),(V_{max},\gamma_{max}),(V_b,\gamma_{min}),(V_{min},\gamma_{min})\}$ there exists a control input in U which keeps the system state inside $\cap_{i\in\{1,2,3,4\}}\{(V,\gamma) \mid J_i^*(V,\gamma) \geq 0\}$. Consider the point (V_a,γ_{max}). At this point, the set of control inputs which keeps the system state inside the set $\{(V,\gamma) \mid J_1^*(V,\gamma) \geq 0\}$ is $\{(T_{max},\theta_{min})\}$, and the set of control inputs which keeps the system state inside $\{(V,\gamma) \mid J_2^*(V,\gamma) \geq 0\}$ is the set $\{(T,\theta) | T \in [T_{min}, T_{max}], \theta \in [\theta_{min}, \frac{M}{a_L V_a c}(\frac{g\cos\gamma_{max}}{V_a} - \frac{a_L V_a(1-c\gamma_{min})}{M})]\}$. Since these two sets have non-empty intersection, the intersection point $(V_a,\gamma_{max}) \in W_{V\gamma}^*$. Similar analysis holds for the remaining three intersection points. Thus $W_{V\gamma}^* = \cap_{i\in\{1,2,3,4\}}\{(V,\gamma) \mid J_i^*(V,\gamma) \geq 0\}$. ∎

Lemma 6 *The least restrictive controller that renders $W_{V\gamma}^*$ controlled invariant is* *denoted $W^*_{V\gamma}$, is the set enclosed by*

$$\partial W^*_{V\gamma} = \{(V,\gamma) \mid \begin{array}{ll} (V = V_{min}) \wedge (\gamma_{min} \leq \gamma \leq \gamma_a) & \vee \\ (V,\gamma) \in \partial J^a & \vee \\ (\gamma = \gamma_{max}) \wedge (V_a \leq V \leq V_{max}) & \vee \\ (V = V_{max}) \wedge (\gamma_b \leq \gamma \leq \gamma_{max}) & \vee \\ (V,\gamma) \in \partial J^b & \vee \\ (\gamma = \gamma_{min}) \wedge (V_{min} \leq V \leq V_b)\} & \end{array} \tag{6.33}$$

Proof: We first prove that the boundary of the set $\cap_{i\in\{1,2,3,4\}}\{(V,\gamma) \mid J_i^*(V,\gamma) \geq 0\}$ is the boundary constructed in equation (6.33). We then prove that this set is equal to $W^*_{V\gamma}$, the maximal controlled invariant set contained in $F_{V\gamma}$.

Consider first the edge $\{(V,\gamma) \mid l_1(V,\gamma) = 0\}$ in ∂F. We will show that

$$\{(V,\gamma) \mid J_1^*(V,\gamma) = 0\} = \{(V,\gamma) \mid (V = V_{min}) \wedge (\gamma_{min} \leq \gamma \leq \gamma_a)\} \cup \{(V,\gamma) \in \partial J^a\} \tag{6.34}$$

The optimal Hamiltonian $H_1^*((V,\gamma),p)$ satisfies:

$$H_1^*((V,\gamma),p) \begin{cases} < 0 & (V,\gamma) \in F_{V\gamma} \cap l_1(V,\gamma) = 0 \cap \gamma > \gamma_a \\ = 0 & (V,\gamma) \in F_{V\gamma} \cap l_1(V,\gamma) = 0 \cap \gamma = \gamma_a \\ > 0 & (V,\gamma) \in F_{V\gamma} \cap l_1(V,\gamma) = 0 \cap \gamma < \gamma_a \end{cases} \tag{6.35}$$

Thus, the set $\{(V,\gamma) \mid (V = V_{min}) \wedge (\gamma_{min} \leq \gamma \leq \gamma_a)\}$ remains unchanged under the evolution of the Hamilton-Jacobi equation (6.30), since $H_1^* > 0$ for this set. We now prove that for $(V,\gamma) \in \partial J^a$, $J_1^*(V,\gamma) = 0$. $J_1^*(V,\gamma)$ satisfies:

$$\begin{bmatrix} \frac{\partial J_1^*(V,\gamma)}{\partial V} \\ \frac{\partial J_1^*(V,\gamma)}{\partial \gamma} \end{bmatrix} [-\frac{a_D V^2}{M} - g\sin\gamma + \frac{1}{M}T_{max}, \frac{a_L V(1-c\gamma)}{M} - \frac{g\cos\gamma}{V} + \frac{a_L c V}{M}\theta_{min}] = 0 \tag{6.36}$$

Since

$$\begin{bmatrix} \frac{\partial J_1^*(V,\gamma)}{\partial V} \\ \frac{\partial J_1^*(V,\gamma)}{\partial \gamma} \end{bmatrix} \tag{6.37}$$

is the inward pointing normal to $\{(V,\gamma) \mid J_1^*(V,\gamma) = 0\}$, then for each (V,γ) in $\{(V,\gamma) \mid J_1^*(V,\gamma) = 0\}$, the vector field

$$\begin{bmatrix} -\frac{a_D V^2}{M} - g\sin\gamma + \frac{1}{M}T_{max} \\ \frac{a_L V(1-c\gamma)}{M} - \frac{g\cos\gamma}{V} + \frac{a_L c V}{M}\theta_{min} \end{bmatrix} \tag{6.38}$$

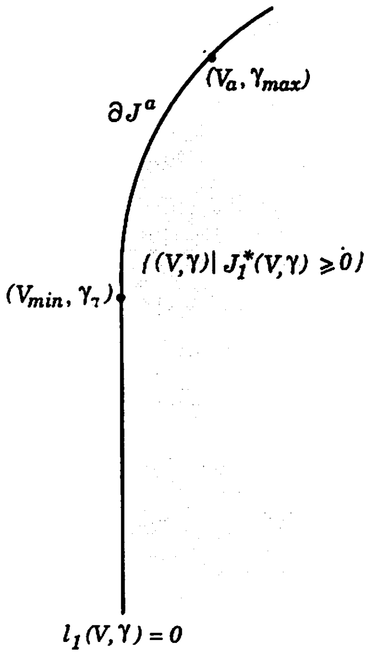


Figure 6.13: Computing the set $\{(V,\gamma) \mid J_1^*(V,\gamma) = 0\}$.

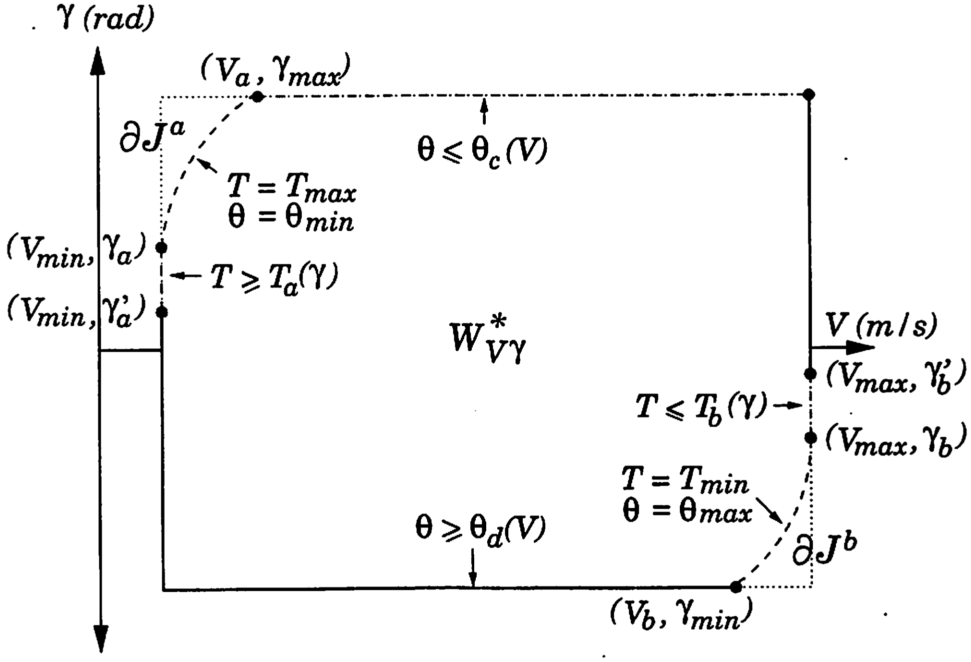


Figure 6.14: The set $W_{V\gamma}^*$ in (V,γ)-space, with control law as indicated. Values used are for a DC-8: $\gamma_{min} = -\pi/8$ rad, $\gamma_{max} = \pi/8$ rad, $V_{min} = 180$ m/s, $V_{max} = 240$ m/s, $\theta_{min} = -\pi/8$ rad, $\theta_{max} = \pi/8$ rad, $T_{min} = 40$ kN, $T_{max} = 80$ kN.

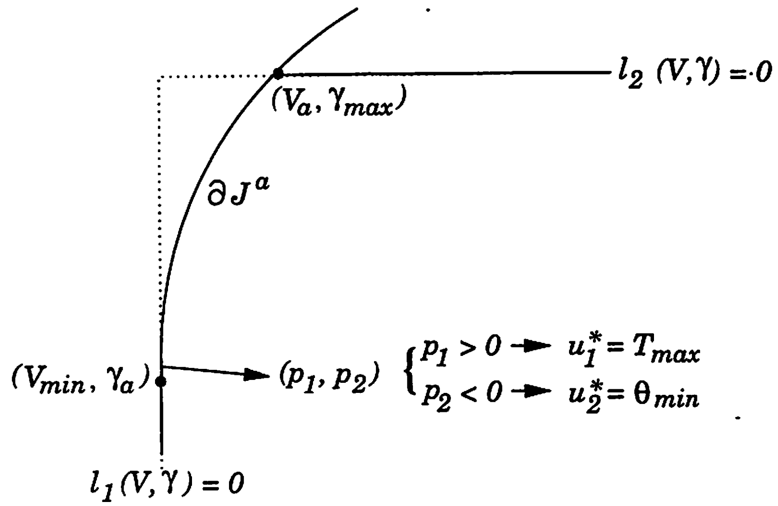


Figure 6.12: Computing the boundary ∂J^a.

$[\theta_{min}, \theta_{max}]$, the inward pointing normal to the solution $(V(t), \gamma(t))$ of the system (2.14), (2.15), starting at (V_{min}, γ_a) and proceeding backwards in time for small $t < 0$ using $u_1 = u_1^*$, is such that p_2 is negative. Thus, $u_2^* = \theta_{min}$. Denote the point of intersection of the solution of (2.14), (2.15) with $\{(V,\gamma) \mid l_2(V,\gamma) = 0\}$ as (V_a, γ_{max}), and the solution to (2.14), (2.15) between (V_{min}, γ_a) and (V_a, γ_{max}) as ∂J^a, as shown in Figure 6.12. Repeat this calculation for the remaining three boundaries. Of the remaining three, only $\{(V,\gamma) \mid l_3(V,\gamma) = 0\}$ contains a point at which the associated optimal Hamiltonian, $H_3^*((V,\gamma),p)$, becomes zero. We denote this point as (V_{max}, γ_b) where:

$$\gamma_b = \sin^{-1}(\frac{T_{min}}{Mg} - \frac{a_D V_{max}^2}{Mg}) \tag{6.32}$$

and similarly calculate ∂J^b and V_b, as shown in Figure 6.14.

Lemma 5 *For the aircraft dynamics (2.14), (2.15) with flight envelope $F_{V\gamma}$ given by (6.22), and input constraints (2.12), the maximal controlled invariant subset of $F_{V\gamma}$,* optimal cost is found by maximizing with respect to u:

$$J_1^*((V,\gamma),t) = \max_{u(\cdot)\in\mathcal{U}} J_1((V,\gamma),u(\cdot),t) \tag{6.28}$$

We seek to compute $W_1^* = \{(V,\gamma) \mid J_1^*(V,\gamma) \geq 0\}$, which are those (V,γ) for which there exists a control input which keeps the system to the right of $l_1(V,\gamma) = 0$. The optimal Hamiltonian is given by the following, where we have substituted into the dynamics the expressions for the lift L and drag D forces (2.11) (neglecting the quadratic term in D):

$$H_1^*((V,\gamma),p) = \max_{u\in U}[p_1(-\frac{a_D V^2}{M} - g\sin\gamma + \frac{1}{M}T) + p_2(\frac{a_L V(1-c\gamma)}{M} - \frac{g\cos\gamma}{V} + \frac{a_L cV}{M}\theta)] \tag{6.29}$$

where $p = (p_1, p_2) \in \mathbb{R}^2$. The Hamilton-Jacobi equation describing the evolution of $J_1^*((V,\gamma),t)$ is obtained from (4.37):

$$-\frac{\partial J_1^*(x,t)}{\partial t} = \begin{cases} H_1^*((V,\gamma), \frac{\partial J_1^*((V,\gamma),t)}{\partial(V,\gamma)}) & \text{for } \{(V,\gamma) \in X \mid J_1^*((V,\gamma),t) > 0\} \\ \min\{0, H_1^*((V,\gamma), \frac{\partial J_1^*((V,\gamma),t)}{\partial(V,\gamma)})\} & \text{for } \{(V,\gamma) \in X \mid J_1^*((V,\gamma),t) \leq 0\} \end{cases} \tag{6.30}$$

with boundary condition $J_1^*((V,\gamma),0) = l_1((V,\gamma))$.

The optimal control at $t = 0$ is computed from equation (6.29). The optimal throttle input T may be calculated directly from this equation: $u_1^*(0) = T_{max}$ (since $p_1 > 0$ for the inward pointing normal). The optimal pitch input must be calculated indirectly[1]. Define $(V_{min}, \gamma_a) = \{(V,\gamma) \mid l_1(V,\gamma) = 0 \cap H_1^*(V,\gamma) = 0\}$. Then:

$$\gamma_a = \sin^{-1}(\frac{T_{max}}{Mg} - \frac{a_D V_{min}^2}{Mg}) \tag{6.31}$$

Integrate the system dynamics (2.14), (2.15) with $(V(0),\gamma(0)) = (V_{min},\gamma_a)$, $u = (u_1^*, u_2^*)$, backwards from $t = 0$ to $t = -T$, where T is chosen to be large enough so that the solution intersects $\{(V,\gamma) \mid l_2(V,\gamma) = 0\}$. The optimal control u_2^* is required for this calculation. At the abnormal extremal (V_{min},γ_a), any $u_2 \in [\theta_{min}, \theta_{max}]$ may be used. However, as we integrate the system, we leave the abnormal extremal regardless of the choice of u_2 instantaneously, and u_2^* is uniquely determined. For all $u_2 \in$

[1] Since $H_1^*((V,\gamma),p)$ loses dependence on u_2 on the set $\{(V,\gamma) \mid l_1(V,\gamma) = 0\}$, the calculations involve computing the so-called *abnormal extremals* [68].

the envelope $F_{hV\dot{h}}$ at all times.

Speed, Flight Path Modes

In the speed and flight path modes (modes 1, 2, 3 in Section 2.4.2), V and γ are the only controlled variables, therefore we may derive the maximal controlled invariant set contained in $F_{V\gamma}$, using the (V,γ)-dynamics (2.14), (2.15):

$$\begin{aligned} \dot{V} &= -\frac{D}{M} - g\sin\gamma + \frac{T}{M}\cos\alpha \\ V\dot{\gamma} &= \frac{L}{M} - g\cos\gamma + \frac{T}{M}\sin\alpha \end{aligned}$$

where $\alpha = \theta - \gamma$. Let

$$F_{V\gamma} = \{(V,\gamma) \mid \forall i \in \{1,2,3,4\},\ l_i(V,\gamma) \geq 0\} \tag{6.22}$$

where

$$\begin{aligned} l_1(V,\gamma) &= V - V_{min} && (6.23) \\ l_2(V,\gamma) &= -\gamma + \gamma_{max} && (6.24) \\ l_3(V,\gamma) &= -V + V_{max} && (6.25) \\ l_4(V,\gamma) &= \gamma - \gamma_{min} && (6.26) \end{aligned}$$

$\partial F_{V\gamma}$ is only piecewise smooth, contradicting the assumption of existence of a differentiable function $l : (V,\gamma) \to \mathbb{R}$ such that $\partial F_{V\gamma} = \{(V,\gamma) \mid l(V,\gamma) = 0\}$. We show that, for this example, the calculation can in fact be performed one edge of the boundary at a time: we derive a Hamilton-Jacobi equation for each l_i, and prove that the intersection of the resulting sets is the maximal controlled invariant subset of $F_{V\gamma}$. The subscript i in each J_i, H_i will indicate that the calculation is for boundary l_i.

Starting with $l_1(V,\gamma)$, consider the system (2.14), (2.15) over the time interval $[t,0]$, where $t < 0$, with cost function

$$J_1((V,\gamma),u(\cdot),t) : \mathbb{R}^+ \times \mathbb{R} \times \mathcal{U} \times \mathbb{R}_- \to \mathbb{R} \tag{6.27}$$

such that $J_1((V,\gamma),u(\cdot),t) = l_1(V(0),\gamma(0))$. Since there are no disturbances in our model, the dynamic game of Section 4.2 reduces to an optimal control problem. The

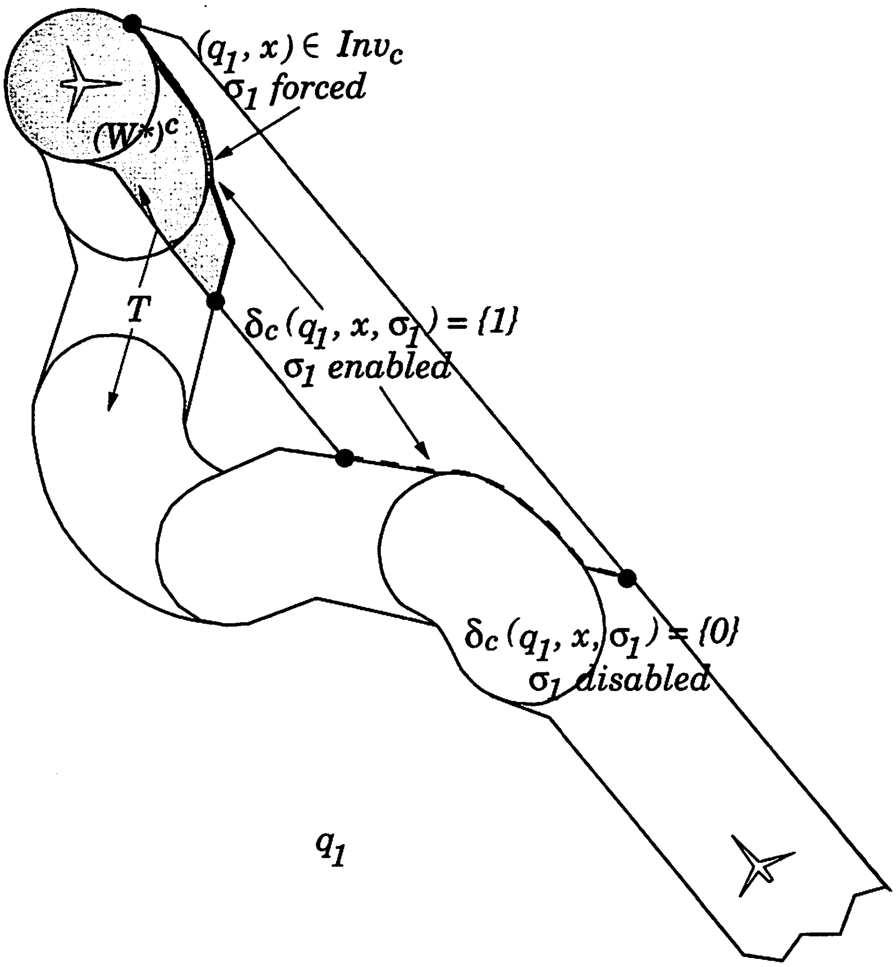


Figure 6.11: $(W^*)^c = (W^{-7})^c$ in q_1. The enabling and forcing boundaries for σ_1 are shown, and the controller $(\delta_c, Inv_c) \in H_c$ may be constructed as shown.

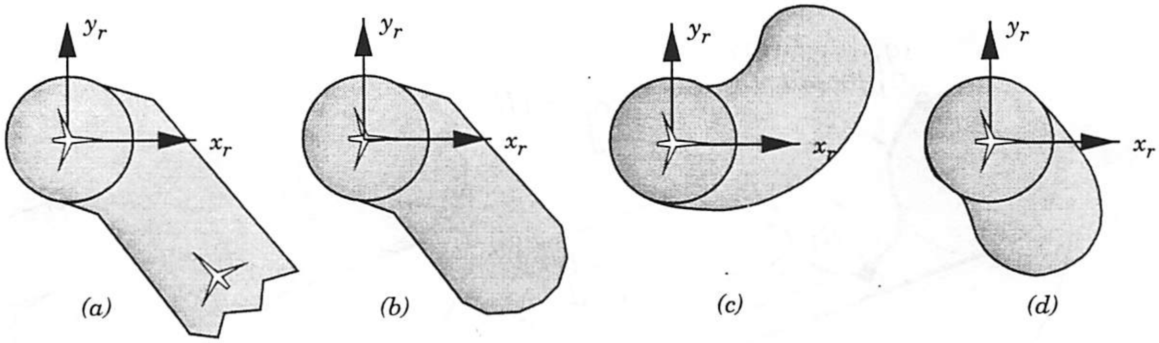


Figure 6.10: $J^*_{G_i}(x) \leq 0$ for (a) Modes 1 and 7 ($i = 1, 7$), $\omega_1 = \omega_2 = 0$ and $[\underline{v}_1, \overline{v}_1] = [2, 4]$, $[\underline{v}_2, \overline{v}_2] = [1, 5]$ (the jagged edge means the set extends infinitely); (b) Modes 3 and 5 ($i = 3, 5$), $\omega_1 = \omega_2 = 0$ and $[\underline{v}_1, \overline{v}_1] = [2, 4]$, $[\underline{v}_2, \overline{v}_2] = [1, 5]$; (c) Mode 4 ($i = 4$), $\omega_1 = \omega_2 = 1$ and $v_1 = v_2 = 5$; and (d) Modes 2 and 6 ($i = 2, 6$), $\omega_1 = \omega_2 = -1$ and $v_1 = v_2 = 5$. In all cases, $\psi_r = 2\pi/3$.

where $l(x) = x_r^2 + y_r^2 - 5^2$. As in the previous example, let G_i be the unsafe set in mode i, and let $J^*_{G_i}$ be the optimal cost function in mode i.

The sets $\{x \in X \mid J^*_{G_i} \leq 0\}$ are shown in Figure 6.10 for $i = 1, \ldots, 7$. In the straight modes, the sets are calculated using (u^*, d^*) of Section 6.1.1 (and thus show a close resemblance to the set in Figure 6.1(a)).

Figure 6.11 displays the fixed point $W^* = W^{-7}$ in q_1. The controller $(\delta_c, Inv_c) \in H_c$ is constructed as in the previous example, and is illustrated in Figure 6.11. The time spent in the straight legs of the maneuver, T, may be chosen to maximize W^*.

6.2 Mode Switching for the Longitudinal Axis Dynamics of a CTOL Aircraft

6.2.1 Continuous Dynamics

Consider the longitudinal dynamics of the CTOL aircraft (2.13) in which the state $\mathbf{x} = (x, \dot{x}, h, \dot{h})^T$ is required to stay in the envelope F, shown in Figure 2.11(a) in (V, γ)-space, and 2.11(b) in $hV\dot{h}$-space. The specification may be decoupled according to $F_{V\gamma}$ and $F_{hV\dot{h}}$: the airspeed V and flight path angle γ must remain in the envelope $F_{V\gamma}$ at all times; and the airspeed, altitude h and vertical speed $\dot{h}$ must remain in

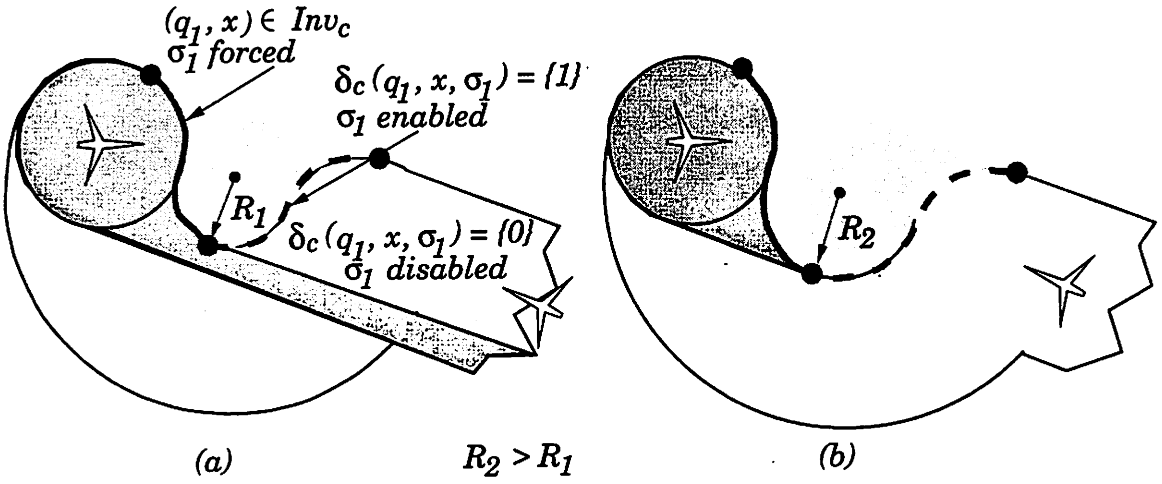


Figure 6.9: Showing the enabling and forcing boundaries for σ_1 in state q_1; and the result of increasing the radius of the turn in the avoid maneuver to increase W^*.

the aircraft are doomed to collide. Figure 6.9(b) displays the result of increasing the radius of the turn in q_2. Notice that the set W^* (the complement of the shaded region) increases as the tuıııing radius increases. This implies that the maneuver renders a larger subset of the state space safe. Figure 6.9(b) shows the critical value of the turning radius, for which the maneuver is guaranteed to be safe, provided the conflict is detected early enough.

6.1.3 Controller Synthesis for Seven-Mode Example

It is straightforward to repeat this analysis for the seven-mode example of Figure 2.9, modeled in Section 3.4.2. In this example, the input and disturbance sets U and D are the ranges of possible airspeeds of aircraft 1 and 2 in the "straight" modes, as described in Section 6.1.1. In the "circular arc" modes, these airspeeds are assumed constant. Thus, our goal is to compute the relative distance of the aircraft at which the maneuver must start, the lengths of the straight legs of the maneuver, as well as the airspeeds u^* and d^*, to ensure safety.

G is defined as:

$$G = \{q_1, \ldots, q_7\} \times \{x \in X \mid l(x) \leq 0\} \tag{6.21}$$

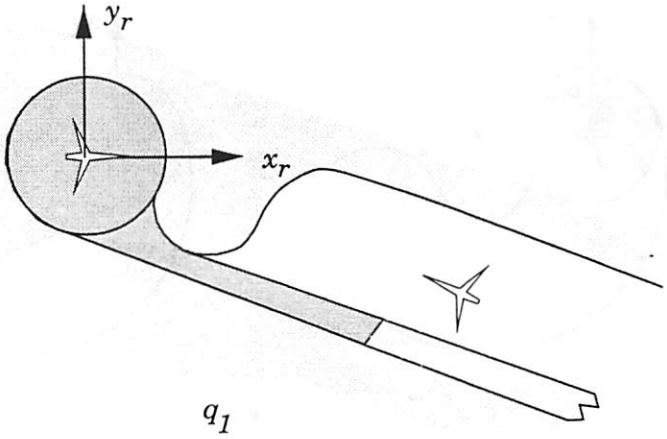


Figure 6.7: $Reach(Pre_2(W^{-2}), Pre_1(W^{-2}))$ in q_1.

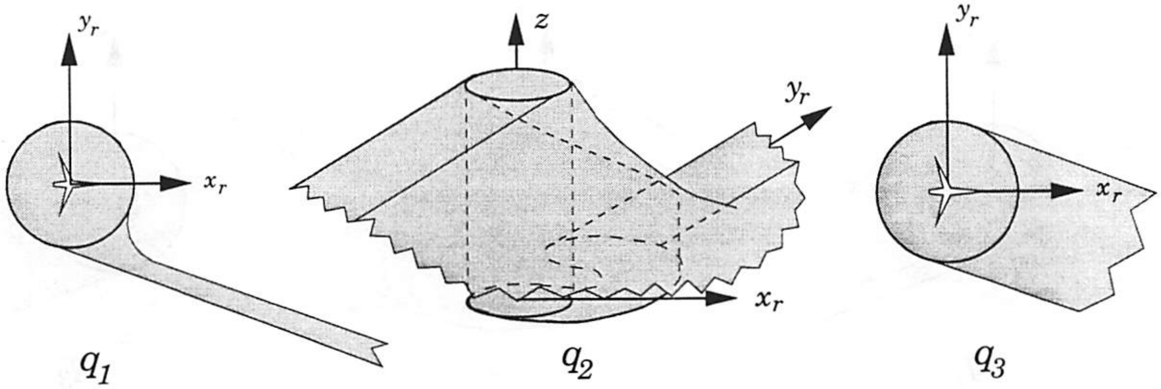


Figure 6.8: $(W^*)^c = (W^{-3})^c$.

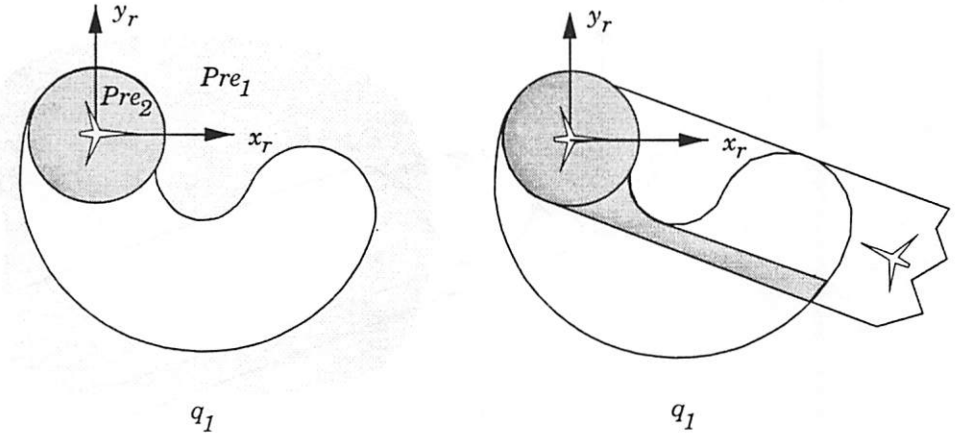


Figure 6.5: (a) $Pre_1(W^{-1})$ and $Pre_2(W^{-1})$ in q_1; (b) $Reach(Pre_2(W^{-1}), Pre_1(W^{-1}))$ in q_1.

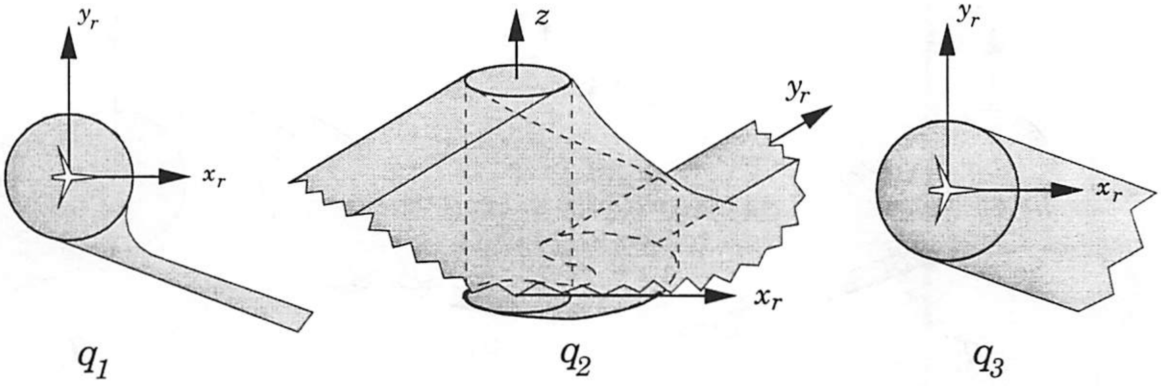


Figure 6.6: $(W^{-2})^c$.

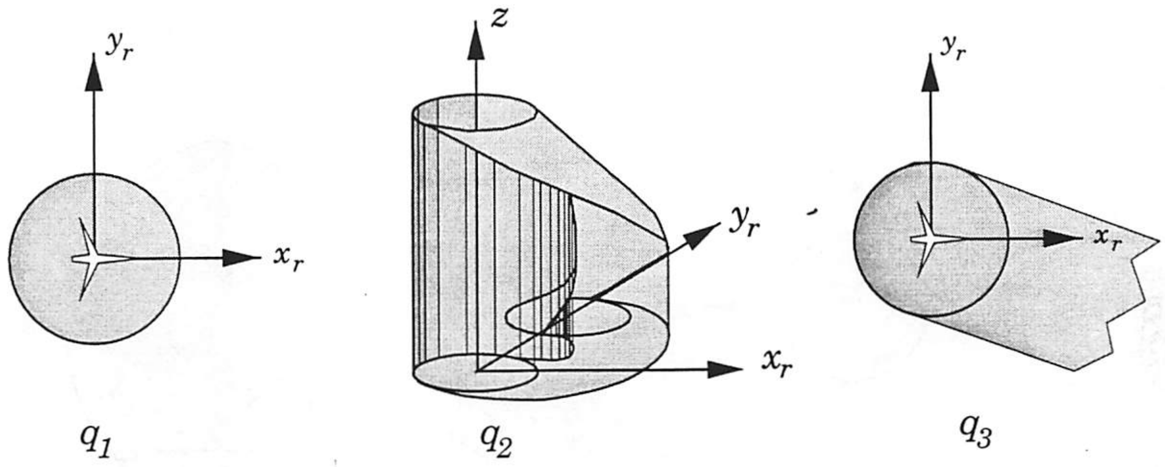


Figure 6.4: $(W^{-1})^c$. The jagged edge in q_3 means that the set extends infinitely.

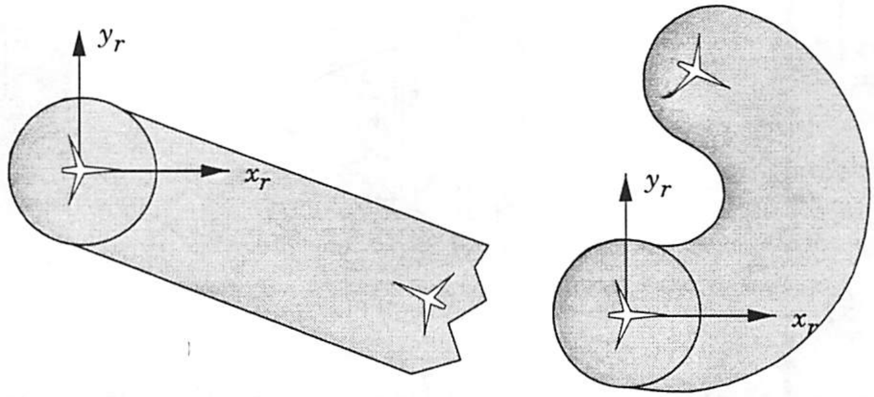
Note that $Pre_1(W^i) \subseteq \{(q_1, X)\}$ for all i, since σ_1 is only defined for transitions from q_1. The set W^{-1} (Figure 6.4) is

$$W^{-1} = W^0 \backslash Reach(Pre_2(W^0), Pre_1(W^0)) \tag{6.20}$$

The set W^{-2} involves computing $Reach(Pre_2(W^{-1}), Pre_1(W^{-1}))$, this computation is illustrated in Figure 6.5(a) and the set is shown in Figure 6.5(b) as the shaded region. Figure 6.6 illustrates the set W^{-2}, Figure 6.7 shows the computation of $Reach(Pre_2(W^{-2}), Pre_1(W^{-2}))$. Figure 6.8 illustrates the fixed point $W^* = W^{-3}$.

As we assumed in this example that the continuous control input $u = v_1$ is fixed, we need only design the discrete part of the controller $(\delta_c, Inv_c) \in H_c$ for the action $\sigma_1 \in \Sigma_c$, which specifies when the maneuver should start. The design is as illustrated in Figure 6.9(a). σ_1 must be disabled $(\delta_c(q_1, x, \sigma_1) = \{0\})$ until the relative dynamics in q_1 reach the dashed line as shown, otherwise the aircraft will lose separation with each other either during the maneuver or after the maneuver is complete. At the dashed line, σ_1 is enabled $(\delta_c(q_1, x, \sigma_1) = \{1\})$, meaning the transition from q_1 to q_2 may occur at any time. σ_1 remains enabled until the solid line (boundary of W^*), at which point it must be both enabled and forced: $\delta_c(q_1, x, \sigma_1) = \{1\}$ and $(q_1, x) \in Inv_c$, otherwise the aircraft lose separation immediately.

Note that there are states (x_r, y_r) which are not rendered safe by the maneuver. Indeed, if the initial state is in the darker shaded region shown in Figure 6.9(a), then



(a) *(b)*

Figure 6.2: $J^*_{G_i}(x) \leq 0$ for (a) Modes 1 and 3 ($i = 1, 3$), $\omega_1 = \omega_2 = 0$ (the jagged edge means the set extends infinitely), (b) Mode 2 ($i = 2$), $\omega_1 = \omega_2 = 1$. In both cases, $\psi_r = 2\pi/3$, and $v_1 = v_2 = 5$.

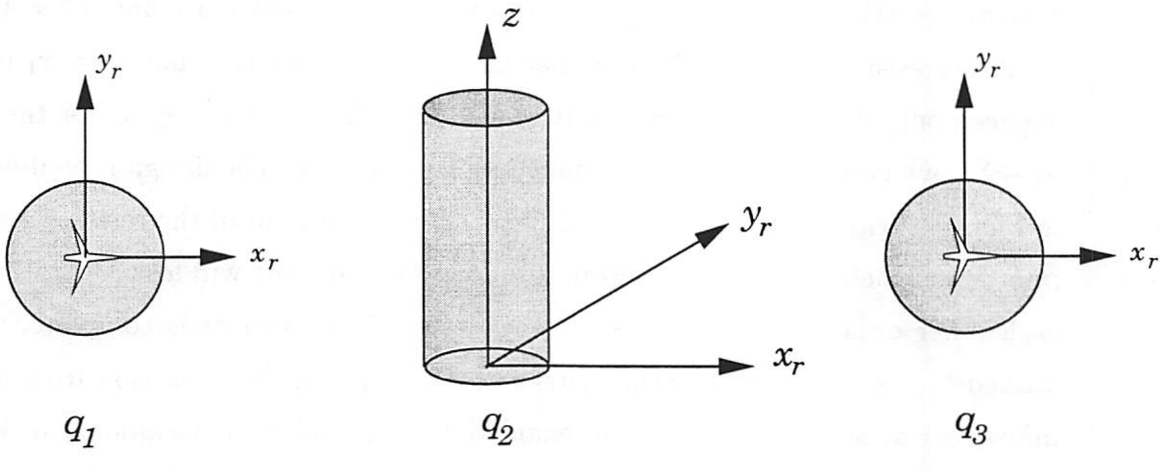


Figure 6.3: $(W^0)^c$.

in which U and D are ranges of possible speeds, is considered in the next example. Recall that our goal is to calculate the relative distance at which the system may safely switch from mode 1 to mode 2, and the minimum turning radius R in mode 2, to ensure that separation between aircraft is maintained.

The evolution of the protected zone in each mode, assuming no switches, may be computed as in the previous section using the continuous-time Hamilton-Jacobi method of Section 4.2. The unsafe set G is defined as:

$$G = \{q_1, q_2, q_3\} \times \{x \in X \mid l(x) \leq 0\} \tag{6.13}$$

where

$$l(x) = x_r^2 + y_r^2 - 5^2 \tag{6.14}$$

and let

$$G_i = (q_i, \{x \in X \mid l(x) \leq 0\}) \tag{6.15}$$

represent the unsaf^ set in mode i. Thus the set

$$\{x \in X \mid J^*_{G_i}(x) \leq 0\} \tag{6.16}$$

where $J^*_{G_i}$ is the optimal cost as defined in equation (4.30), is the backwards evolution of the protected zone in mode i, assuming no switches between modes. These sets are shown in Figure 6.2.

Now let us implement Algorithm (5.6) for this example, at each step computing the sets Pre_1, Pre_2, and $Reach(Pre_2, Pre_1)$. In the first step, $W^0 = F \triangleq G^c$, the complement of G:

$$\begin{aligned} W^0 &= ((q_1, \{x \in X \mid l(x) \leq 0\}^c \cap \{x \in X \mid z = 0\}) \cup (q_2, \{x \in X \mid l(x) \leq 0\}^c) \\ &\quad \cup (q_3, \{x \in X \mid l(x) \leq 0\}^c \cap \{x \in X \mid z = 0\})) \end{aligned} \tag{6.17}$$

as shown in Figure 6.3 (its complement is actually shown).

$$Pre_1(W^0) = (q_1, \{x \in X \mid l(x) \leq 0\}^c \cap \{x \in X \mid z = 0\}) \tag{6.18}$$

$$Pre_2(W^0) = G \tag{6.19}$$

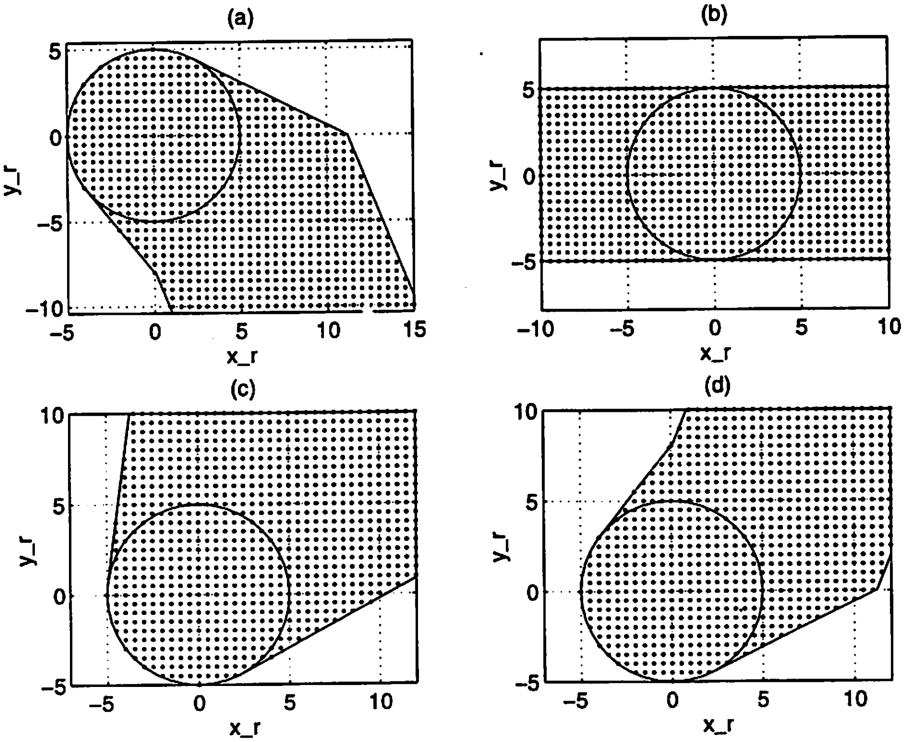


Figure 6.1: The set $\{x \in X \mid J^*(x,t) \leq 0\}$ shown in the (x_r, y_r)-plane for $[\underline{v}_1, \overline{v}_1] = [2,4]$, $[\underline{v}_2, \overline{v}_2] = [1,5]$ and (a) $\psi_r = \pi/2$, (b) $\psi_r = 0$, (c) $\psi_r = -\pi/4$, (d) $\psi_r = -\pi/2$.

As can be seen from equation (6.4), the optimal airspeed u^* depends on the position of aircraft 2 relative to aircraft 1. If aircraft 2 is ahead of aircraft 1 in the relative axis frame, then u^* is at its lower limit, if aircraft 2 is behind aircraft 1 in the relative axis frame then u^* is at its upper limit. If aircraft 2 is heading towards aircraft 1, then d^* is at its upper limit, and if aircraft 2 is heading away from aircraft 1, d^* is at its lower limit. The unsafe sets of states are illustrated in Figure 6.1 for various values of ψ_r, and airspeed ranges as illustrated.

6.1.2 Controller Synthesis for Three-Mode Example

Consider the three-mode conflict resolution example pictured in Figure 2.8, and modeled in Section 3.4.1. We assume that for this example the airspeeds (v_1, v_2) of both aircraft are constant even in the straight modes, so that the input and disturbance sets are singletons ($U = v_1, D = v_2$) and $u^* = v_1, d^* = v_2$. The general case,

$$\begin{aligned}
& -x_r(t)\int_t^0 \delta v_1(s)ds - x_r(0)\int_t^0 \delta v_1(s)ds \\
& +\underline{v}_2(0-t)[x_r(t)\cos\psi_r + y_r(t)\sin\psi_r] \\
& +\underline{v}_2(0-t)[x_r(0)\cos\psi_r + y_r(0)\sin\psi_r] \\
\leq\ & x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(0-t) - x_r(0)\underline{v}_1(0-t) \\
& +\underline{v}_2(0-t)[x_r(t)\cos\psi_r + y_r(t)\sin\psi_r] \\
& +\underline{v}_2(0-t)[x_r(0)\cos\psi_r + y_r(0)\sin\psi_r] \\
=\ & J(x, u^*(\cdot), d^*(\cdot), t)
\end{aligned} \tag{6.9}$$

Similarly, let $u = u^*$ and vary d, ie. let $d = \underline{v}_2 + \delta v_2$, where $\delta v_2 \geq 0$. Then

$$\begin{aligned}
J(x, u^*(\cdot), d(\cdot), t) =\ & x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(0-t) - x_r(0)\underline{v}_1(0-t) \\
& +\underline{v}_2(0-t)[x_r(t)\cos\psi_r + y_r(t)\sin\psi_r] \\
& +\underline{v}_2(0-t)[x_r(0)\cos\psi_r + y_r(0)\sin\psi_r] \\
& +\int_t^0 \delta v_2(s)ds[x_r(t)\cos\psi_r + y_r(t)\sin\psi_r] \\
& +\int_t^0 \delta v_2(s)ds[x_r(0)\cos\psi_r + y_r(0)\sin\psi_r] \\
\geq\ & x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(0-t) - x_r(0)\underline{v}_1(0-t) \\
& +\underline{v}_2(0-t)[x_r(t)\cos\psi_r + y_r(t)\sin\psi_r] \\
& +\underline{v}_2(0-t)[x_r(0)\cos\psi_r + y_r(0)\sin\psi_r] \\
=\ & J(x, u^*(\cdot), d^*(\cdot), t)
\end{aligned} \tag{6.10}$$

Summarizing, we have shown above that in this case,

$$J(x, u(\cdot), d^*(\cdot), t) \leq J(x, u^*(\cdot), d^*(\cdot), t) \leq J(x, u^*(\cdot), d(\cdot), t) \tag{6.11}$$

Therefore, $u^* = \underline{v}_1$, $d^* = \underline{v}_2$ is the optimal solution for this case. The solutions for the three other cases can be shown in a similar manner. ■

The solution (u^*, d^*) given by equations (6.4), (6.5) is actually a *saddle solution*, meaning that it is the optimal solution regardless of whether the control or disturbance plays first:

$$\max_{u\in\mathcal{U}} \min_{d\in\mathcal{D}} J(x, u(\cdot), d(\cdot), t) = \min_{d\in\mathcal{D}} \max_{u\in\mathcal{U}} J(x, u(\cdot), d(\cdot), t) \tag{6.12}$$

Proposition 6 [(u^*, d^*) for Airspeed Control] *The optimal solution (u^*, d^*) to the system described by equations (6.1) with cost $J(x, u(\cdot), d(\cdot), t)$ given by equation (4.9) is*

$$u^* = \begin{cases} \underline{v}_1 & \textit{if } sgn(s_1) > 0 \\ \overline{v}_1 & \textit{if } sgn(s_1) < 0 \end{cases} \tag{6.4}$$

$$d^* = \begin{cases} \underline{v}_2 & \textit{if } sgn(s_2) > 0 \\ \overline{v}_2 & \textit{if } sgn(s_2) < 0 \end{cases} \tag{6.5}$$

Proof: Starting at time $t < 0$ (free) and integrating to the final time 0, the solution to equations (6.1) has $\psi_r(t) = \psi_r(0)$ and

$$\begin{aligned} x_r(0) &= x_r(t) - \textstyle\int_t^0 u(s)ds + \cos\psi_r \ \int_t^0 d(s)ds \\ y_r(0) &= y_r(t) + \sin\psi_r \textstyle\int_t^0 d(s)ds \end{aligned} \tag{6.6}$$

Substituting equations (6.6) into the cost (4.9), (4.52) (ignoring the constant 5^2) results in

$$\begin{aligned} J(x, u(\cdot), d(\cdot), t) &= x_r^2(0) + y_r^2(0) && (6.7) \\ &= x_r^2(t) + y_r^2(t) - x_r(t) \int_t^0 u(s)ds - x_r(0) \int_t^0 u(s)ds \\ &\quad + \int_t^0 d(s)ds[x_r(t)\cos\psi_r + y_r(t)\sin\psi_r] \\ &\quad + \int_t^0 d(s)ds[x_r(0)\cos\psi_r + y_r(0)\sin\psi_r] && (6.8) \end{aligned}$$

Defining the *switching functions* $s_1(t), s_2(t)$ as in equations (6.3), we consider the case in which, $\forall t \le 0$,

$$sgn(s_1(t)) > 0, \quad sgn(s_2(t)) > 0$$

We will show that in this case $u^* = \underline{v}_1$ and $d^* = \underline{v}_2$. Note that we assume that in the interval $[t, 0]$, *neither $s_1(t)$ nor $s_2(t)$ change sign.* If t is such that the switching functions do change sign on this interval, then the interval must be broken into two intervals, and the optimal solution calculated separately for each interval.

Let $d = d^*$ and vary u, ie. let $u = \underline{v}_1 + \delta v_1$, where $\delta v_1 \ge 0$. Then

$$J(x, u(\cdot), d^*(\cdot), t) = x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(0 - t) - x_r(0)\underline{v}_1(0 - t)$$

6.1 Conflict Resolution for Two Aircraft in $SE(2)$

6.1.1 Continuous Dynamics

Consider the two-aircraft relative model (2.9):

$$\begin{aligned}\dot{x}_r &= -v_1 + v_2\cos\psi_r + \omega_1 y_r \\ \dot{y}_r &= v_2\sin\psi_r - \omega_1 x_r \\ \dot{\psi}_r &= \omega_2 - \omega_1\end{aligned}$$

in which the aircraft either follow straight paths ($\omega_i = 0, i = 1,2$) or arcs of circles ($\omega_i = 1, i = 1,2$). The continuous inputs are the airspeeds of the aircraft ($u = v_1, d = v_2$). In the straight modes, the airspeeds vary over specified ranges: $u \in U = [\underline{v}_1, \overline{v}_1] \subset \mathbb{R}^+$, $d \in D = [\underline{v}_2, \overline{v}_2] \subset \mathbb{R}^+$, and model (2.9) reduces to

$$\begin{aligned}\dot{x}_r &= -u + d\cos\psi_r \\ \dot{y}_r &= d\sin\psi_r \\ \dot{\psi}_r &= 0\end{aligned} \tag{6.1}$$

In the circular arc modes, the airspeeds are fixed at constant values: $U = v_1 \in \mathbb{R}^+$, $D = v_2 \in \mathbb{R}^+$, and model (2.9) reduces to

$$\begin{aligned}\dot{x}_r &= -v_1 + v_2\cos\psi_r + y_r \\ \dot{y}_r &= v_2\sin\psi_r - x_r \\ \dot{\psi}_r &= 0\end{aligned} \tag{6.2}$$

In this section, we derive the optimal control u^* and worst disturbance d^* for the relative system in the straight modes of operation. Since equations (6.1) are linear and simple to manipulate, rather than deriving the Hamiltonian we calculate (u^*, d^*) directly, by integrating equations (6.1) for piecewise constant u and d and substituting the solutions into the cost function (4.9). Define the switching functions s_1 and s_2 as

$$\begin{aligned}s_1(t) &\triangleq x_r \\ s_2(t) &\triangleq x_r\cos\psi_r + y_r\sin\psi_r\end{aligned} \tag{6.3}$$

Chapter 6

Application to Distributed Air Traffic Management

In this chapter, we apply our controller synthesis algorithm for hybrid systems to the three air traffic examples introduced in Chapter 2. For each example, we first derive and solve the Hamilton-Jacobi equation for the continuous dynamics only, as described in Section 4.2, and then apply the controller synthesis algorithm of Section 5.1 to compute the maximal controlled invariant set and corresponding control law so that each system satisfies its specified safety requirement. For these examples, the Hamilton-Jacobi equations are simple enough, and the dimensions of the discrete and continuous state spaces small enough, to permit solutions using the method of characteristics with some help from MATLAB. For systems of many aircraft, each modeled by nonlinear equations with hundreds of modes of operation, sophisticated computation tools based on partial differential equation solvers are needed. This is the subject of the next chapter.

[70]. In practice, we are helped by the fact that we are usually interested in finite time computations, rather than computing for $t \to -\infty$ or until a fixed point is reached.

Another problem is the requirement that the controller resulting from our algorithm be *non-Zeno* (does not enforce the safety requirement by preventing time from diverging). The algorithm proposed here has no way of preventing such behavior, as will be illustrated in the third example which we solve in the next chapter. We will discuss in the next chapter a practical method of resolving the Zeno effect: adding a requirement that the system must remain in each discrete state for a non-zero amount of time.

Theorem 1 (Characterization of Reach-Avoid) *Assume that $J_G^*(x,t)$ ($J_E^*(x,t)$ respectively) satisfies the Hamilton-Jacobi equation (5.8) ((5.9) respectively), and that it converges uniformly in x as $t \rightarrow -\infty$ to a function $J_G^*(x)$ ($J_E^*(x)$ respectively). Then,*

$$Reach(G,E) = \{x \in X \mid J_G^*(x) < 0\} \tag{5.23}$$

Proof: Let $x_0 \in \{x \in X \mid J_G^*(x) < 0\}$. Therefore, by construction, for all $u(\cdot) \in \mathcal{U}$ there exists $d(\cdot) \in \mathcal{D}$ such that the state trajectory $x(\cdot)$, starting at x_0, will eventually enter G. Also, by Lemma 3, $J_E^*(x_0) > 0$. Thus $\forall u(\cdot) \in \mathcal{U}$, $\exists d(\cdot) \in \mathcal{D}$, such that the state trajectory $x(\cdot)$ starting at x_0 never enters E. Thus, $\{x \in X \mid J_G^*(x) < 0\} \subseteq Reach(G,E)$.

Now let $x_0 \in \{x \in X \mid J_G^*(x) \geq 0\}$. Assume for the sake of contradiction that for all $u(\cdot) \in \mathcal{U}$, there exists a $d(\cdot) \in \mathcal{D}$ such that the trajectory $x(\cdot)$, starting at $(x_0, 0)$, enters G. Since for all $x \in G$, $J_G^*(x) < 0$, there exists a time $t_1 > 0$ at which this trajectory crosses $\{x \in X \mid J_G^*(x) = 0\}$. However, for all x such that $J_G^*(x) = 0$, there must exist a $u \in U$ such that for all $d \in D$, $f(x,u,d)$ points outside of $\{x \in X \mid J_G^*(x) < 0\}$. This contradicts the assumption of existence of a $d(\cdot)$ which drives the system to G. Thus, $Reach(G,E) \subseteq \{x \in X \mid J_G^*(x) < 0\}$. ∎

The controller which renders W^* invariant is:

$$\begin{aligned}
h_c(q,x) &= \begin{cases} \{u \in U \mid \min_{d \in D} \frac{\partial J_G^*(x)}{\partial x} f(q,x,u,d) \geq 0\} & \text{if } x \in \partial W^* \\ U & \text{if } x \in (W^*)^\circ \end{cases} & (5.24)\\
\delta_c(q,x,\sigma_1) &= \{1\} \Leftrightarrow (q,x) \in Pre_1(W^*) & (5.25)\\
Inv_c &= (W^*)^c & (5.26)
\end{aligned}$$

5.2 Remarks

In general, one cannot expect to solve for W^* using a finite computation. The class of hybrid systems for which algorithms like the one presented here are guaranteed to terminate is known to be restricted [69]. Techniques have been proposed to resolve this problem, making use of approximation schemes to obtain estimates of the solution which implies that $J^*_E(x_0, t) = J^*_E(x_0, 0) \geq 0$ for all $t \leq 0$, which contradicts (5.16). A similar argument holds for the case in which $J^*_G(x_0, 0) \geq 0$ and $J^*_E(x_0, 0) < 0$.

Thus $J^*_G(x_0, 0) \geq 0$ and $J^*_E(x_0, 0) \geq 0$. Since $J^*_G(x, t_1) < 0$, there exists $t_2 \in [t_1, 0]$ such that $J^*_G(x_0, t_2) = 0$, and for all $t \in [t_1, t_2]$, $J^*_G(x_0, t) \leq 0$. Thus for at least some interval in $[t_1, t_2]$, $J^*_E(x_0, t) > 0$ (to allow $J^*_G(x_0, t)$ to decrease in this interval) and

$$\frac{\partial J^*_E(x_0, t)}{\partial t} = 0 \tag{5.18}$$

But this contradicts the assumption that $x_0 \in E^\circ(t_1)$. A symmetric argument holds for $J^*_E(x_0, t)$.

Therefore, $G^\circ(t) \cap E^\circ(t) = \emptyset$. ■

Lemma 4 *For all $t \leq 0$,*

$$G(t) \cap E(t) = \partial G(t) \cap \partial E(t) \tag{5.19}$$

Moreover, for all $t' \leq t$,

$$G(t) \cap E(t) \subseteq \partial G(t') \cap \partial E(t') \tag{5.20}$$

Proof:

$$G(t) \cap E(t) = (G^\circ(t) \cap E^\circ(t)) \cup (\partial G(t) \cap \partial E(t)) \cup (G^\circ(t) \cap \partial E(t)) \cup (\partial G(t) \cap E^\circ(t)) \tag{5.21}$$

From Lemma 3, $(G^\circ(t) \cap E^\circ(t)) = \emptyset$.

Assume that for some $t = t_1 < 0$, $x_0 \in G^\circ(t) \cap \partial E(t)$. Therefore, $J^*_G(x_0, t_1) < 0$ and $J^*_E(x_0, t_1) = 0$. Therefore, there exists $t_2 \in [t_1, 0]$ such that $J^*_G(x_0, t_2) = 0$ and for all $t \in [t_1, t_2]$, $J^*_G(x_0, t) \leq 0$. Thus for some interval of $[t_1, t_2]$, $J^*_E(x_0, t) > 0$ and

$$\frac{\partial J^*_E(x_0, t)}{\partial t} = 0 \tag{5.22}$$

which contradicts the assumption that $x_0 \in \partial E(t_1)$. Thus $G^\circ(t) \cap \partial E(t) = \emptyset$. A symmetric argument holds for $x_0 \in \partial G(t) \cap E^\circ(t)$ for $t = t_1 < 0$, thus $\partial G(t) \cap E^\circ(t) = \emptyset$.

Therefore, $G(t) \cap E(t) = \partial G(t) \cap \partial E(t)$. That $G(t) \cap E(t) \subseteq \partial G(t') \cap \partial E(t')$ for $t' \leq t$ follows from Lemma 2. ■

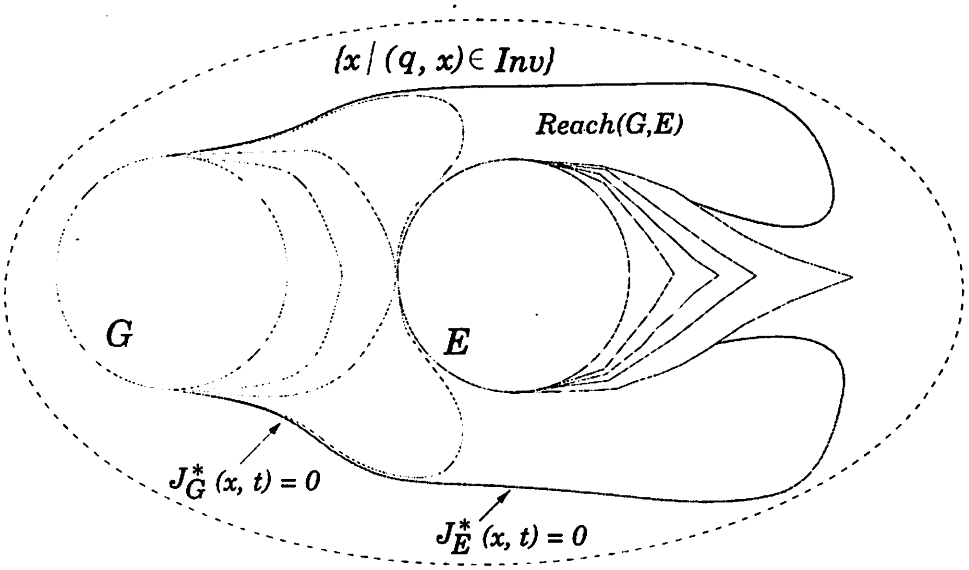


Figure 5.2: The computation of *Reach*(G, E) in a single discrete state q.

Proof: Since $\frac{\partial J_G^*}{\partial t} \geq 0$ when $J_G^*(x,t) \leq 0$ and $\frac{\partial J_E^*}{\partial t} \geq 0$ when $J_E^*(x,t) \leq 0$, both $J_G^*(x,t)$ and $J_E^*(x,t)$ are monotone non-increasing functions of $-t$ when $J_G^*(x,t) \leq 0$ and $J_E^*(x,t) \leq 0$. Thus, as t decreases, the sets $G(t)$ and $E(t)$ do not decrease in size, meaning that once a state x is inside $G(t)$ ($E(t)$ respectively) it stays inside $G(t)$ ($E(t)$ respectively) as t decreases. ■

Lemma 3 *For all* $t \leq 0$,

$$G^\circ(0) \cap E^\circ(0) = \emptyset \Rightarrow G^\circ(t) \cap E^\circ(t) = \emptyset \tag{5.15}$$

Proof: Assume for the sake of contradiction that $x_0 \in G^\circ(t) \cap E^\circ(t)$ for some $t = t_1 < 0$, i.e. that

$$J_G^*(x_0, t_1) < 0 \quad \text{and} \quad J_E^*(x_0, t_1) < 0 \tag{5.16}$$

We first show that $J_G^*(x_0, 0) \geq 0$ and $J_E^*(x_0, 0) \geq 0$ (meaning that x_0 is outside of both G and E at $t = 0$). Suppose this is not true, i.e. suppose for example that $J_G^*(x_0, 0) < 0$ and $J_E^*(x_0, 0) \geq 0$. Then for all $t \leq 0$

$$\frac{\partial J_E^*(x_0, t)}{\partial t} = 0 \tag{5.17}$$

and

$$-\frac{\partial J_E^*(x,t)}{\partial t} = \begin{cases} H_E^*(x, \frac{\partial J_E^*(x,t)}{\partial x}) & \text{for } \{x \in X \mid J_E^*(x,t) > 0\} \\ \min\{0, H_E^*(x, \frac{\partial J_E^*(x,t)}{\partial x})\} & \text{for } \{x \in X \mid J_E^*(x,t) \leq 0\} \end{cases} \tag{5.9}$$

where $J_G(x, u(\cdot), d(\cdot), t) = l_G(x(0))$ and $J_E(x, u(\cdot), d(\cdot), t) = l_E(x(0))$, and

$$H_G^*(x, \frac{\partial J_G^*}{\partial x}) = \begin{cases} 0 & \text{for } \{x \in X \mid J_E^*(x,t) \leq 0\} \\ \max_{u \in U} \min_{d \in D} \frac{\partial J_G^*}{\partial x} f(x,u,d) & \text{otherwise} \end{cases} \tag{5.10}$$

$$H_E^*(x, \frac{\partial J_E^*}{\partial x}) = \begin{cases} 0 & \text{for } \{x \in X \mid J_G^*(x,t) \leq 0\} \\ \min_{u \in U} \max_{d \in D} \frac{\partial J_E^*}{\partial x} f(x,u,d) & \text{otherwise} \end{cases} \tag{5.11}$$

Equation (5.8) describes the evolution of the set G under the Hamiltonian H_G^* (5.10). This is the "$\max_u \min_d$" game of the previous chapter, with the modification that $H_G^* = 0$ in $\{x \in X \mid J_E^*(x,t) \leq 0\}$ which ensures that the evolution of $J_G^*(x,t)$ is frozen in this set. Similarly, equation (5.9) describes the evolution of the set E under the Hamiltonian H_E^*. Here a "$\min_u \max_d$" is used, since it is assumed that the control tries to push the system into E, to escape from G. $H_E^* = 0$ in $\{x \in X \mid J_G^*(x,t) \leq 0\}$ to ensure that the evolution of $J_E^*(x,t)$ is frozen in this set. Note that in both games, the disturbance is given the advantage by assuming that the control plays first. In the following sequence of Lemmas, we prove that the resulting set $\{x \in X \mid J_G^*(x,t) < 0\}$ contains neither E nor states for which there is a control $u(\cdot) \in \mathcal{U}$ which drives the system into E; and the set $\{x \in X \mid J_E^*(x,t) < 0\}$ contains neither G nor states for which there is a disturbance input $d(\cdot) \in \mathcal{D}$ which drives the system into G. We then prove that $\{x \in X \mid J_G^*(x,t) < 0\}$ is the set *Reach*(G, E). Figure 5.2 illustrates an example.

For all $t \leq 0$, let

$$G(t) \triangleq \{x \in X \mid J_G^*(x,t) \leq 0\} \quad E(t) \triangleq \{x \in X \mid J_E^*(x,t) \leq 0\} \tag{5.12}$$

Thus $G = G(0)$ and $E = E(0)$.

Lemma 2 *For all $t_2 \leq t_1 \leq 0$,*

$$G(t_1) \subseteq G(t_2) \tag{5.13}$$

$$E(t_1) \subseteq E(t_2) \tag{5.14}$$

Consider the following algorithm.

```
Let     $W^0 = F, W^{-1} = \emptyset, i = 0.$
While   $W^i \neq W^{i-1}$ do
        $W^{i-1} = W^i \setminus Reach(Pre_2(W^i), Pre_1(W^i)))$        (5.6)
        $i = i - 1$
end
```

In the first step of this algorithm, we remove from F all states from which there is a disturbance $d(\cdot) \in \mathcal{D}$ forcing the system either outside F or to states from which an environment action $\sigma_2 \in \Sigma_2$ can force transitions outside F, without first touching the set of states from which there is a control action $\sigma_1 \in \Sigma_1$ keeping the system inside F. Since at each step, $W^{i-1} \subseteq W^i$, the set W^i decreases monotonically as i decreases. If the algorithm terminates, we denote the fixed point as W^*.

In order to implement this algorithm, we need to calculate Pre_1, Pre_2, and $Reach$. The calculation of Pre_1 and Pre_2 is done by inverting the transition relation δ. The calculation of $Reach$ can be carried out by appropriately modifying the Hamilton-Jacobi construction of Section 4.2, as we describe below.

Note that in Algorithm (5.6), $Reach(Pre_2(W^i), Pre_1(W^i))$ is computed in parallel in all discrete states in Q. In the following analysis, we describe this calculation for *one* discrete state $q \in Q$. Abusing notation, we denote the unsafe set G as

$$G = \{x \in X \mid (q, x) \in Inv \cap F^c\} \tag{5.7}$$

and implicitly assume that all subsets of states are restricted to $\{x \in X \mid (q, x) \in Inv\}$.

Let $l_G : X \to \mathbb{R}$ and $l_E : X \to \mathbb{R}$ be differentiable functions such that $G \triangleq \{x \in X : l_G(x) \leq 0\}$ and $E \triangleq \{x \in X : l_E(x) \leq 0\}$ (in general G and E may be expressed as the intersection of a set of differentiable functions, as discussed in Chapter 7). Consider the following system of interconnected Hamilton-Jacobi equations:

$$-\frac{\partial J_G^*(x,t)}{\partial t} = \begin{cases} H_G^*(x, \frac{\partial J_G^*(x,t)}{\partial x}) & \text{for } \{x \in X \mid J_G^*(x,t) > 0\} \\ \min\{0, H_G^*(x, \frac{\partial J_G^*(x,t)}{\partial x})\} & \text{for } \{x \in X \mid J_G^*(x,t) \leq 0\} \end{cases} \tag{5.8}$$

5.1 Algorithm

Consider the nonlinear hybrid automaton (3.6) with trajectory acceptance condition $\Omega = \Box F$, with $F \subseteq Q \times X$. We seek to construct the largest set of states for which the control $(u(\cdot), \sigma_1[\cdot])$ can guarantee that the acceptance condition is met despite the action of the disturbance $(d(\cdot), \sigma_2[\cdot])$. For any set $K \subseteq Q \times X$, we define the *controllable predecessor* $Pre_1(K)$ and the *uncontrollable predecessor* $Pre_2(K)$ by

$$\begin{aligned} Pre_1(K) &= \{(q,x) \in Q \times X | \exists \sigma_1 \in \Sigma_1 \ \forall \sigma_2 \in \Sigma_2 \ \delta(q,x,\sigma_1,\sigma_2) \subseteq K\} \cap K \\ Pre_2(K) &= \{(q,x) \in Q \times X | \forall \sigma_1 \in \Sigma_1 \ \exists \sigma_2 \in \Sigma_2 \ \delta(q,x,\sigma_1,\sigma_2) \cap K^c \neq \emptyset\} \cup K^c \end{aligned} \tag{5.4}$$

Therefore $Pre_1(K)$ contains all states in K for which a controllable action σ_1 can force the state to remain in K for at least one step in the discrete evolution. The intersection with K in the equation for $Pre_1(K)$ excludes states which are outside of K and have a transition into K. $Pre_2(K)$, on the other hand, contains all states in K^c, the complement of K, as well as all states from which an uncontrollable action σ_2 may be able to force the state outside of K.

Proposition 5 $Pre_1(K) \cap Pre_2(K) = \emptyset$.

Proof: Suppose $(q,x) \in Pre_1(K) \cap Pre_2(K)$. Since $(q,x) \in Pre_1(K)$, there exists a $\sigma_1 \in \Sigma_1$ (call it σ_1^*) such that for all $\sigma_2 \in \Sigma_2$, $\delta(q,x,\sigma_1^*,\sigma_2) \subseteq K$. Since $(q,x) \in Pre_2(K)$, for all $\sigma_1 \in \Sigma_1$, there exists a $\sigma_2 \in \Sigma_2$ such that $\delta(q,x,\sigma_1,\sigma_2) \cap K^c \neq \emptyset$. But this contradicts the existence of σ_1^*. ■

In order to construct the backwards iteration we need the "reach-avoid" operator:

Definition 7 (Reach-Avoid) *Consider two subsets $G \subseteq Q \times X$ and $E \subseteq Q \times X$ such that $G \cap E = \emptyset$. The reach-avoid operator is defined as*

$$\begin{aligned} Reach(G,E) = \ & \{(q,x) \in Q \times X \mid \forall u \in \mathcal{U} \ \exists d \in \mathcal{D} \text{ and } t \geq 0 \text{ such that} \\ & (q(t),x(t)) \in G \text{ and } (q(s),x(s)) \in Inv \setminus E \text{ for } s \in [0,t]\} \end{aligned} \tag{5.5}$$

where $(q(s),x(s))$ is the continuous state trajectory of $\dot{x} = f(q(s),x(s),u(s),d(s))$ starting at (q,x). The set $Reach(G,E)$ describes those states from which, for all $u(\cdot) \in \mathcal{U}$, there exists a $d(\cdot) \in \mathcal{D}$, such that the state trajectory $(q(s),x(s))$ can be driven to G while avoiding an "escape" set E.

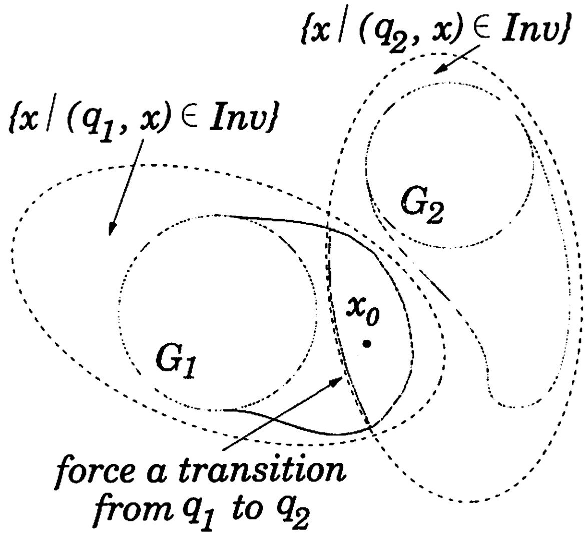

$\{x \,/\, (q_2, x) \in Inv\}$
$\{x \,/\, (q_1, x) \in Inv\}$
G_2
x_0
G_1
force a transition
from q_1 to q_2

Figure 5.1: In q_1, the portion of the unsafe set which intersects the invariant of q_2 may be made "safe" by switching from q_1 to q_2.

ous chapter separately for each q_i, and derived sets

$$\{x \in X \mid (q_1, x) \in Inv \text{ and } J^*_{G_1}(x) < 0\} \tag{5.2}$$

$$\{x \in X \mid (q_2, x) \in Inv \text{ and } J^*_{G_2}(x) < 0\} \tag{5.3}$$

If there exists an x_0 in the intersection of the two invariants, as shown in Figure 5.1, where $J^*_{G_1}(x_0) < 0$ but $J^*_{G_2}(x_0) > 0$, then hybrid system trajectories which start at (q_1, x_0) and stay in q_1 are unsafe, whereas trajectories which switch to q_2 are safe. Therefore, it makes sense to construct a discrete transition so that, if possible, the system automatically switches from q_1 to q_2 on the boundary of the intersection of the invariants. In such a way, the designer of a control scheme chooses not only the continuous control law $u(\cdot)$ in each discrete state, but chooses $\sigma_1[\cdot]$ so that states which would have evolved into the unsafe set are made safe by discrete transitions.

In this chapter, we first derive an expression for the largest controlled invariant set $W^* \in F$ for the nonlinear hybrid automaton (3.6) with $\Omega = \Box F$. We then describe the process for generating the control law $(u(\cdot), \sigma_1[\cdot])$ which guarantees that the system remains in W^*. In the next chapter, we apply this controller synthesis algorithm to our three air traffic examples.

Chapter 5

Controller Synthesis for Nonlinear Hybrid Systems

In the previous chapter we derived expressions for the largest controlled invariant set W^*, for both finite state automata and continuous differential equations. We also derived the least restrictive control laws to ensure that the state trajectories of these systems remain in W^*. For these two systems, if the system state strays from W^*, then there is no way to guarantee safety of the system.

Now consider the corresponding problem of synthesizing a control law $(u(\cdot), \sigma_1[\cdot])$, in the presence of environmental disturbances $(d(\cdot), \sigma_2[\cdot])$, for the nonlinear hybrid automaton (3.6)

$$H = (Q \times X, U \times D, \Sigma_1 \times \Sigma_2, f, \delta, Inv, I, Y, h, \Box F)$$

for $F \subseteq Q \times X$. Associated to each discrete state $q_i \in Q$ is a subset of the continuous state space $\{x \in X \mid (q_i, x) \in Inv\}$ in which the system may evolve when in q_i. As an example, consider two discrete states q_1 and q_2, with invariants as illustrated in Figure 5.1. The unsafe set $G \subseteq Q \times X$ may be written as the union of two subsets of X:

$$G = (q_1, G_1) \cup (q_2, G_2) \tag{5.1}$$

Suppose we performed the continuous-time Hamilton-Jacobi calculation of the previ-

T (throttle) and the aircraft pitch θ (through the elevators), thus there are two continuous control inputs $(u_1, u_2) = (T, \theta)$. Physical considerations impose constraints on the inputs:

$$u \in [T_{min}, T_{max}] \times [\theta_{min}, \theta_{max}] \tag{2.12}$$

The longitudinal dynamics may be modeled by the Newton-Euler equations:

$$M \begin{bmatrix} \ddot{x} \\ \ddot{h} \end{bmatrix} = R(\theta) \left[R^T(\alpha) \begin{bmatrix} -D \\ L \end{bmatrix} + \begin{bmatrix} T \\ 0 \end{bmatrix} \right] + \begin{bmatrix} 0 \\ -Mg \end{bmatrix} \tag{2.13}$$

where $R(\alpha)$ and $R(\theta)$ are standard rotation matrices, M is the mass of the aircraft, and g is gravitational acceleration. The state of the system is $\mathbf{x} = (x, \dot{x}, h, \dot{h})^T$.

The airspeed of the aircraft is defined as $V = \sqrt{\dot{x}^2 + \dot{h}^2}$. The simplified FMS studied in this dissertation uses control inputs T and θ to control combinations of the airspeed V, flight path angle γ, and altitude h. The linear and angular accelerations $(\dot{V}, V\dot{\gamma})$ may be derived directly from (2.13):

$$\dot{V} = -\frac{D}{M} - g \sin\gamma + \frac{T}{M}\cos\alpha \tag{2.14}$$

$$V\dot{\gamma} = \frac{L}{M} - g\cos\gamma + \frac{T}{M}\sin\alpha \tag{2.15}$$

Note that these dynamics are expressed solely in terms of (V, γ) and inputs (T, θ), where $\alpha = \theta - \gamma$; thus equations (2.14), (2.15) are a convenient way to represent the dynamics for modes in which h is not a controlled variable.

Safety regulations for the aircraft dictate that V, γ, and h must remain within specified limits:

$$\begin{aligned} V_{min} \le V \le V_{max} \\ \gamma_{min} \le \gamma \le \gamma_{max} \\ h_{min} \le h \le h_{max} \end{aligned} \tag{2.16}$$

where $V_{min}, V_{max}, \gamma_{min}, \gamma_{max}, h_{min}, h_{max}$ are functions of such factors as airspace regulations, type of aircraft, and weather. For aircraft flying in en-route airspace, we assume that these limits are constants, and thus the aerodynamic flight envelope F is as illustrated in Figure 2.11, in (V, γ)-space and $(h, V, \dot{h})$-space, where $\dot{h} = V \sin\gamma$. The state trajectory must remain within F at all times during en-route flight. We

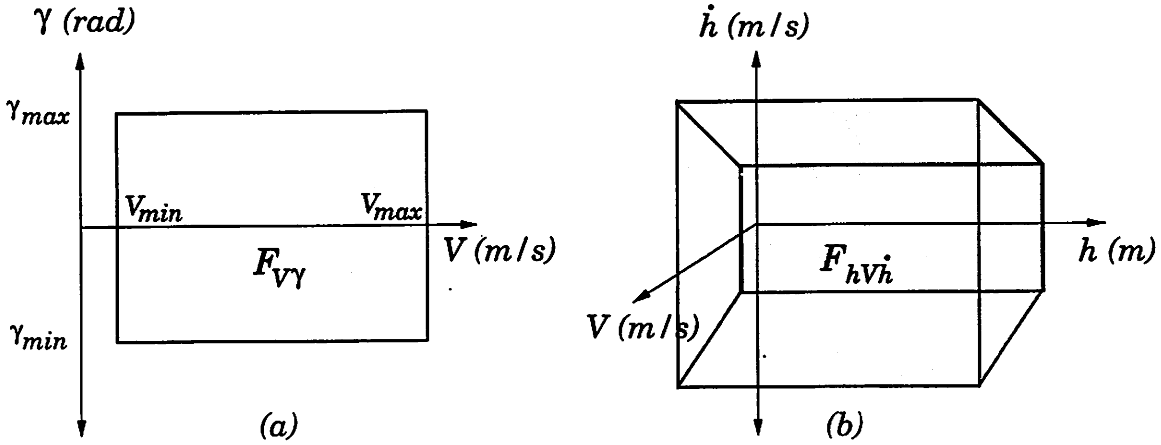


Figure 2.11: (a) Simplified Aerodynamic Flight Envelope in (V, γ)-space: axes are airspeed V, flight path angle γ; (b) Simplified Aerodynamic Flight Envelope in $(h, V, \dot{h})$-space: axes are altitude h, airspeed V, vertical speed $\dot{h}$.

also impose a *secondary criterion*, that the state trajectory must satisfy constraints on the linear and angular acceleration:

$$|\dot{V}| \leq 0.1g, \quad |V\dot{\gamma}| \leq 0.1g \tag{2.17}$$

imposed for passenger comfort.

The system may be discretized into five flight modes, depending on the state variables being controlled:

- **Mode 1:** (Speed, Flight Path), in which the thrust T is between its specified operating limits ($T_{min} < T < T_{max}$), the control inputs are T and θ, and the controlled outputs are the speed and the flight path angle of the aircraft $y = (V, \gamma)^T$;

- **Mode 2:** (Speed), in which the thrust saturates ($T = T_{min} \vee T = T_{max}$) and thus it is no longer available as a control input; the only input is θ, and the only controlled output is V;

- **Mode 3:** (Flight Path), in which the thrust saturates ($T = T_{min} \vee T = T_{max}$); the input is again θ, and the controlled output is γ;

- **Mode 4:** (Speed, Altitude), in which the thrust T is between its specified operating limits ($T_{min} < T < T_{max}$), the control inputs are T and θ, and the controlled outputs are the speed and the vertical position of the aircraft $y = (V, h)^T$;
- **Mode 5:** (Altitude), in which the thrust saturates $T = T_{min} \vee T = T_{max}$; the input is θ, and the controlled output is h.

In our calculations we use the following parameter values, which correspond to a DC-8 at cruising speed: M = 85000kg, b = 0.01, c = 6, a_L = 30, a_D = 2, T_{min} = 40000 N, T_{max} = 80000 N, $\theta_{min} = -22.5°$, $\theta_{max} = 22.5°$, V_{min} = 180 m/s, V_{max} = 240 m/s, $\gamma_{min} = -22.5°$ and $\gamma_{max} = 22.5°$. The bounds on the pitch angle θ and the flight path angle γ are chosen to be symmetric about zero for ease of computation. In actual flight systems, the positive bound on these angles is greater than the negative bound. Also, the angles chosen for this example are greater than what are considered acceptable for passenger flight (±10°). Since we are interested in en route flight, the limits on the altitudes are: $h_{min} = 15,000$ feet, $h_{max} = 51,000$ feet.

Problem statement: Generate the mode switching logic for this flight management system, as well as the continuous control inputs (T, θ) to use in each flight mode, so that envelope protection is guaranteed.

Chapter 3

Nonlinear Hybrid System Model

Our goal is to develop a mathematical representation of such systems as described in the previous chapter. The representation should be compact, yet rich enough to describe both the evolution of continuous aircraft dynamics as well as the hundreds of discrete maneuvers and flight modes. Since the model is to be used to verify safety properties of and synthesize controllers for real-life safety critical systems, we would like it to be capable of modeling uncertainty in both the continuous and discrete variables. Finally, the model should be fairly easy to program into a computer, so that controller synthesis may be done automatically.

In this section we present a model for a *nonlinear hybrid automaton* which has all of these properties. The model is called *hybrid* because it combines nonlinear continuous dynamics with the dynamics of discrete event systems. Along with *control variables* through which the controller has access to the hybrid automaton, we incorporate in the model *environment variables* which cannot be controlled and whose values are uncertain. Also, we show in subsequent chapters how existing controller synthesis techniques for purely discrete and purely continuous systems may be combined in a clever way to produce a computationally feasible controller synthesis method for nonlinear hybrid automata. Our model is based on the hybrid system model of [20], developed further in [26] and [31].

As background, we first present a model for a discrete event system, and then one for a purely continuous nonlinear system. We describe the state and input spaces, the control and environment variables, system trajectories and safety properties. We then present a model for a nonlinear hybrid automaton. We describe compositions of hybrid automata, define a special hybrid automaton called a controller, and present an interconnection of two hybrid automata: plant and controller. Finally, we describe how the three air traffic examples introduced in Chapter 2 are modeled as nonlinear hybrid automata.

3.1 Background

3.1.1 Finite Automata

We describe a variant of a finite state automaton, whose actions are partitioned into those of two "players", the controller and the environment. The controller's actions may be used to achieve a desired goal, whereas the actions of the environment are uncontrollable, uncertain, and could possibly oppose the controller's actions. The automaton is represented as

$$(Q, \Sigma, \delta, Q_0, \Omega) \tag{3.1}$$

where $Q = \{q_1, q_2, \dots, q_m\}$ is a finite set of *discrete states*; $\Sigma = \Sigma_1 \times \Sigma_2$ is a finite set of *actions*, $\delta : Q \times \Sigma_1 \times \Sigma_2 \rightarrow 2^Q$ is a *partial transition relation*, $Q_0 \subseteq Q$ is a set of *initial states*, and Ω is a *trajectory acceptance condition*. Σ_1 contains the actions of the controller and Σ_2 contains the actions of the environment, so that each transition between states depends on a joint action (σ_1, σ_2). Note that the behavior of the finite state automaton is *non-deterministic*: the transition function $\delta(q, \sigma_1, \sigma_2)$ represents a set of possible new states, rather than a single unique state. Transitions are prevented, or blocked, from occurring at state q by setting $\delta(q, \sigma_1, \sigma_2) = \emptyset$.

A *system trajectory* $(q[\cdot], \sigma_1[\cdot], \sigma_2[\cdot]) \in Q^\omega \times \Sigma_1^\omega \times \Sigma_2^\omega$ is a finite or infinite sequence of states and actions which satisfies, for $i \in \mathbb{Z}$,

$$q[0] \in Q_0 \text{ and } q[i+1] \in \delta(q[i], \sigma_1[i], \sigma_2[i]) \tag{3.2}$$

The trajectory acceptance condition describes a desired goal that the system should achieve, which is expressed as a specification on the state trajectory. For safety specifications, in which the state trajectories must remain within a safe subset $F \subseteq Q$, the trajectory acceptance condition is written as $\Omega = \Box F$, meaning that $\forall i, q[i] \in F$. The controller wins the game if the trajectory satisfies $\Box F$, otherwise the environment wins.

3.1.2 Nonlinear Continuous-Time Dynamics

As in the discrete case we consider two players, controller and environment, competing over nonlinear continuous-time systems of the form

$$\dot{x}(t) = f(x(t), u(t), d(t)), \quad x(0) \in X_0 \tag{3.3}$$

where $x \in X$ is the finite-dimensional *state* in an n-manifold (frequently $X = \mathbb{R}^n$), $u \in U \subseteq \mathbb{R}^u$ is the *control input* which models the actions of the controller, $d \in D \subseteq \mathbb{R}^d$ is the *disturbance input* which models the actions of the environment, f is a smooth vector field over $\mathbb{R}^n$, and $X_0 \subseteq X$ is a set of *initial conditions*. The input set $U \times D$ is the continuous-time analog of the partition $\Sigma_1 \times \Sigma_2$. The spaces of acceptable control and disturbance trajectories are denoted by the spaces of piecewise continuous functions $\mathcal{U} = \{u(\cdot) \in PC^0 \mid u(t) \in U, \ \forall t \in \mathbb{R}, U \subseteq \mathbb{R}^u\}$, $\mathcal{D} = \{d(\cdot) \in PC^0 \mid d(t) \in D, \ \forall t \in \mathbb{R}, D \subseteq \mathbb{R}^d\}$.

A *system trajectory* over an interval $[\tau, \tau'] \subseteq \mathbb{R}$ is a map:

$$(x(\cdot), u(\cdot), d(\cdot)) : [\tau, \tau'] \rightarrow X \times U \times D \tag{3.4}$$

such that $u(\cdot) \in \mathcal{U}$, $d(\cdot) \in \mathcal{D}$, $x(\cdot)$ is continuous, and for all $t \in [\tau, \tau']$ where $u(\cdot)$ and $d(\cdot)$ are continuous, $\dot{x}(t) = f(x(t), u(t), d(t))$. We assume that the function f is *globally Lipschitz* in x and continuous in u and d. Then, by the existence and uniqueness theorem of solutions for ordinary differential equations, given an interval $[\tau, \tau']$, the value of $x(t)$ for some $t \in [\tau, \tau']$ and input and disturbance trajectories $u(\cdot), d(\cdot)$ over $[\tau, \tau']$ there exists a unique solution $(x(\cdot), u(\cdot), d(\cdot))$ to (3.3).

The safety specification considered here corresponds to the specification in a class of zero-sum dynamic games known as *pursuit-evasion games*. The controller wins if it can keep the state trajectory from entering a "bad" subset of the state space, called the "capture set" and defined as the interior of a region G, denoted G° and defined as

$$G^\circ = \{x \in X \mid l(x) < 0\} \tag{3.5}$$

with boundary $\partial G = \{x \in X \mid l(x) = 0\}$ where $l : X \to \mathbb{R}$ is a differentiable function of x with $\frac{\partial l(x)}{\partial x} \neq 0$ on ∂G. Equivalently, the specification may be written in terms of a safe set $F \triangleq (G^\circ)^c$, the complement of G° in X. Our convention throughout this dissertation is that safe sets F are closed sets, whereas the capture sets G° are open.

3.2 Nonlinear Hybrid Automata

In this section we combine the finite automaton and nonlinear continuous dynamics into a *nonlinear hybrid automaton* which models both discrete and continuous behavior. The control and environment inputs have continuous and discrete components, and so they may affect the system both continuously and through discrete actions.

Definition 2 (Nonlinear Hybrid Automaton) *We define a* **nonlinear hybrid automaton** *as*

$$H = (Q \times X, U \times D, \Sigma_1 \times \Sigma_2, f, \delta, Inv, I, Y, h, \Omega) \tag{3.6}$$

such that

- **State space.** *$Q \times X$ is the state space, with $Q = \{q_1, q_2, \ldots, q_m\}$ a finite set of discrete states, and X an n-manifold; the state of the system is a pair $(q_i, x) \in Q \times X$;*

- **Continuous control inputs and disturbances.** *$U \times D$ is the product of the set of continuous control inputs and the set of continuous environment inputs, known as disturbances; the space of acceptable control and disturbance trajectories are denoted by $\mathcal{U} = \{u(\cdot) \in PC^0 | u(\tau) \in U, \ \forall \tau \in \mathbb{R}\}$, $\mathcal{D} = \{d(\cdot) \in PC^0 | d(\tau) \in D, \ \forall \tau \in \mathbb{R}\}$;*

- **Discrete control and disturbance actions.** $\Sigma_1 \times \Sigma_2$ *is the product of the finite set of discrete control inputs, or control actions, and the finite set of discrete environment inputs, or disturbance actions;*

- **Continuous map.** $f : Q \times X \times U \times D \to TX$ *is the continuous map which associates with each discrete state* $q \in Q$ *a control system* $f(q, x, u, d)$*;*

- **Discrete transition function.** $\delta : Q \times X \times \Sigma_1 \times \Sigma_2 \to 2^{Q \times X}$ *is the discrete transition function;*

- **Invariants.** $Inv \subseteq Q \times X$ *is the invariant associated with each discrete state, meaning that the system evolves according to* $\dot{x} = f(q, x, u, d)$ *only if* $(q, x) \in Inv$*;*

- **Initial states.** $I \subseteq Q \times X$ *is the set of initial states;*

- **Continuous outputs.** Y *is the set of continuous outputs;*

- **Output map.** $h : Q \times X \to 2^Y$ *is the output map;*

- **Trajectory acceptance condition.** Ω *is the trajectory acceptance condition (here* $\Omega = \Box F$ *for* $F \subseteq Q \times X$*).*

State trajectories of a hybrid system evolve continuously as well as in discrete jumps: the concept of a hybrid time trajectory is therefore needed.

Definition 3 (Hybrid Time Trajectory) *A* **hybrid time trajectory** *is a sequence of intervals of the real line*

$$\tau = [0, \tau_0'][\tau_1, \tau_1'][\tau_2, \tau_2'] \ldots [\tau_k, \tau_k') \tag{3.7}$$

such that $\tau_{i-1}' = \tau_i$ *and* $\tau_i \leq \tau_i'$*. The index* k *may be finite or infinite.*

For $t \in \mathbb{R}$, we use $t \in \tau$ to represent $t \in [\tau_i, \tau_i']$ for some i. Denote by $\mathcal{T}$ the set of all hybrid time trajectories. Let $(q[\cdot], \sigma_1[\cdot], \sigma_2[\cdot]) \in Q^\omega \times \Sigma_1^\omega \times \Sigma_2^\omega$. We may extend this

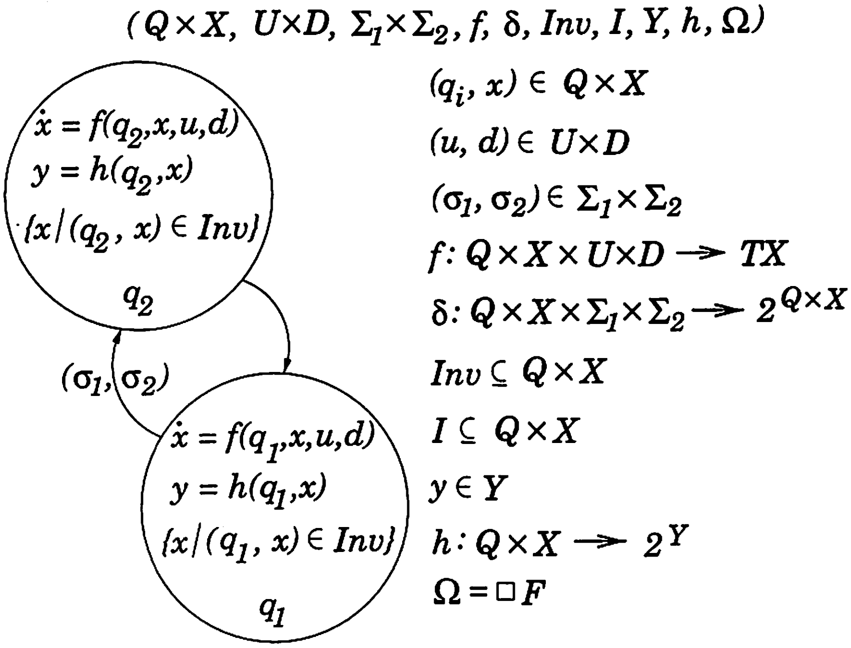


Figure 3.1: Nonlinear Hybrid Automaton.

sequence of states and actions to a function over τ by defining, for all for $t \in [\tau_i, \tau_i']$:

$$\begin{aligned} q(t) &\triangleq q[i] \\ \sigma_1(t) &\triangleq \sigma_1[i] \\ \sigma_2(t) &\triangleq \sigma_2[i] \end{aligned} \tag{3.8}$$

Definition 4 (Hybrid System Trajectory) *A* **hybrid system trajectory** *is defined as*

$$(\tau, q[\cdot], x(\cdot), u(\cdot), d(\cdot), \sigma_1[\cdot], \sigma_2[\cdot]) \tag{3.9}$$

where $\tau \in \mathcal{T}$, $q[\cdot] \in Q^\omega$, $x(\cdot) : \tau \to X$, $u(\cdot) : \tau \to U$, $d(\cdot) : \tau \to D$, $\sigma_1[\cdot] \in \Sigma_1^\omega$, and $\sigma_2[\cdot] \in \Sigma_2^\omega$. The initial condition satisfies $(q[0], x(0)) \in I$; the discrete evolution satisfies

$$(q[i+1], x(\tau_{i+1})) \in \delta(q[i], x(\tau_i'), \sigma_1[i], \sigma_2[i])$$

for all i and τ; the continuous evolution satisfies $(q[i], x(t)) \in Inv$ and

$$\dot{x}(t) = f(q[i], x(t), u(t), d(t))$$

for $t \in [\tau_i, \tau_i']$; and the output evolution satisfies $y(t) \in h(q[i], x(t))$, for $t \in [\tau_i, \tau_i']$. The state trajectory is the $(q[\cdot], x(\cdot))$ such that the above conditions are satisfied.