UC Berkeley
UC Berkeley Electronic Theses and Dissertations

Title
Safety-Guaranteed Autonomy under Uncertainty

Permalink
https://escholarship.org/uc/item/8pb7v9x0

Author
Lee, Donggun

Publication Date
2022

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

Safety-Guaranteed Autonomy under Uncertainty

by

Donggun Lee

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Claire Tomlin, Co-chair
Professor Masayoshi Tomizuka, Co-chair
Professor Sosale Shankar Sastry
Professor Murat Arcak

Spring 2022

Safety-Guaranteed Autonomy under Uncertainty

Copyright 2022
by
Donggun Lee

Abstract

Safety-Guaranteed Autonomy under Uncertainty

by

Donggun Lee

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Claire Tomlin, Co-chair

Professor Masayoshi Tomizuka, Co-chair

Reachability analyzes a dynamic system's abilities to reach goals or maintain safety. This analysis plays an essential role in various safety-critical applications. Previous reachability theory characterizes the success or failure of reachability tasks. However, this does not tell us the degree to which the goal will be achieved, or the safety will be maintained. Our new reachability formulation aims to provide measures for each goal-reaching and safety.

This dissertation introduces three bodies of work. The first presents state-constrained reachability problems that provide the goal-reaching or safety metrics, and the corresponding Hamilton-Jacobi (HJ) frameworks. The HJ frameworks guarantee performance and safety metrics for general nonlinear systems with non-convex constraints. Despite this advantage, its computational complexity is exponential in the state dimension. Thus, it is not scalable for high-dimensional systems. In order to alleviate this computational complexity, the second presents efficient Hopf-Lax theory, which provides analytic solutions to HJ partial differential equations (PDEs) for state-constrained reachability problems, and the third presents reinforcement-learning approaches.

Towards Truth

Contents

List of Figures

List of Tables

Acknowledgments

I am grateful that I had two unique experiences during my Ph.D. First, I have done a lot of work on mathematical theory, but ironically, intuition was more important than rigorous proof. Making good conjectures was like throwing a stone in the dark and looking for the sound of truth. Second, I have worked with people I respect academically and humanly: Professor Claire Tomlin, Professor Masayoshi Tomizuka, Professor S. Shankar Sastry, Professor Murat Arcak, and Professor L. Craig Evans.

I am extremely thankful to my research advisor, dissertation committee members, colleagues, and friends.

I would like to give my biggest thanks to Professor Claire Tomlin as my supervisor. Her advice is always more than words. She has shown me what a good researcher, advisor, chair, and person. Also, I appreciate her incredible devotion and trust in me. Under her supervision, I could find my research paths and expertise.

I would like to express gratitude to Professor Masayoshi Tomizuka, Professor S. Shankar Sastry, and Professor Murat Arcak for their advice on the direction of my dissertation, presentation skills, teaching, mentoring, and research fields. Through interactions with them, I learned about their philosophy and passion for research and warm hearts toward students. I also appreciate Professor L. Craig Evans for his teaching and support. It was fantastic to interact with him, whose work is my research background. I appreciate his education and unbelievable advice on my research.

I would like to thank Professor Mo Chen, Alexander Keimer, Jingqi Li, Ellis Ratner, Jason Choi, Professor Alexandre Bayen, Professor Somayeh Sojoudi, and Professor Koushil Sreenath for our wonderful collaboration. Their devotion, consideration, and intellectual conversations we had are unforgettable. Also, I thank my other collaborators, Hybrid System Lab members, and all the friends I met in Berkeley for making happy moments.

Finally, I appreciate my family and fiance, Hyeonju Shin, waiting for a long time and giving me endless love.

Chapter 1

Introduction

Reachability analyzes dynamic systems' abilities to reach goals or maintain safety. This analysis has been widely used in applications, including air-traffic management [103], tracking control [29], walking robots [83], multirobot safety control [105], aircraft safety control [79, 105, 99], motor control [109], and vehicle platooning [31].

We focus on optimal control or zero-sum game frameworks to deal with reachability. Previous reachability theory deals with state-unconstrained problems. For example, consider two previous reachability problems: the viability kernel and the reach-avoid problem. First, given possibly multiple safety conditions, the viability kernel analyzes controls for the system to satisfy all conditions at all times [8, 73]. In the optimal control framework solving the viability kernel, the cost encodes the multiple conditions that must be satisfied. The optimal cost for the viability kernel characterizes the success or failure of condition satisfaction. However, this does not tell us the degree to which each condition is satisfied. For the second example, in the reach-avoid problem, we analyze control signals under which the system reaches the target while avoiding obstacles [74, 46]. Some previous literature formulates the reach-avoid problem into another state-unconstrained optimal control problem. This analysis provides an indicator to determine success or failure for the reach-avoid task. However, it does not provide the distance measure of how close the system can reach the target while avoiding obstacles.

We aim to formulate new reachability problems that do provide not only success or failure indicators but also meaningful measures. This dissertation presents two state-constrained reachability problems: state-constrained control-invariance problems (SCCIPs) relevant to the viability kernel and state-constrained reach-avoid problems (SCRAPs) relevant to the reach-avoid problem.

The SCCIP is a general version of the viability kernel [8, 73]. In SCCIP, among multiple conditions that must be satisfied, we select one condition whose measure function becomes the cost, and the other conditions are encoded in state constraints. The viability kernel does not provide a measure for the selected condition. On the other hand, SCCIP provides a quantitative measure of how much a particular condition has been satisfied or violated while the other conditions remain satisfied.

The SCRAP is a general version of the reach-avoid problem [74, 46]. In SCRAP, the cost encodes the goal-reaching measure, and the other safety conditions are encoded in state constraints. Although the reach-avoid problem does not measure how close the system can reach the target while avoiding obstacles, SCRAP provides this distance measure.

This dissertation aims to find solutions for the state-constrained reachability problems and guarantee safety and performance. Towards this goal, we first provide Hamilton-Jacobi (HJ) analysis for the state-constrained reachability problems. The proposed HJ analysis guarantees safety and performance metrics for the state-constrained problems even though the dynamics are nonlinear systems and the state constraints non-convex. Despite this advantage, its computational complexity is exponential in the state dimension. Thus, it is not scalable for high-dimensional systems. In order to alleviate this computational complexity, this dissertation presents two approaches: Hopf-Lax theory and reinforcement learning for reachability.

Hopf-Lax theory finds analytic solutions to HJ partial differential equations (PDEs). The theory assumes convex Hamiltonians in the costate space, which holds for all optimal control problems except zero-sum games. In order to solve optimal-control problems, we first find a relevant HJ PDE and then apply Hopf-Lax theory. In fact, Hopf-Lax theory provides either an optimization or optimal control problem. In other words, we can solve the Hopf-Lax formula instead of the given optimal control problem. The advantage of solving the Hopf-Lax formula is that it has better convexity conditions than the given optimal control problem. Thus, gradient-based methods can guarantee safety and performance for broader classes of problems for Hopf-Lax formulae. Previously, Hopf-Lax theory has been developed for HJ PDEs only relevant to state-unconstrained problems. This dissertation presents Hopf-Lax theory relevant to state-constrained problems.

Finally, this dissertation presents a reinforcement learning framework for reachability problems. Reinforcement learning approximately finds a solution to Bellman equations using function approximators, such as deep neural networks. Unfortunately, extensive work in reinforcement learning focuses on general sum problems but not reachability problems. We find the relevant Bellman equations and apply them in reinforcement learning frameworks based on HJ PDEs for reachability.

The contribution of this dissertation is three-fold.

1. We first deal with state-constrained reachability problems and provide the corresponding HJ PDEs. There are two difficulties of dealing with state constraints in reachability. First, gradients are not defined in infeasible state regions. Second, in the continuous-time setting, control signals are infinite-dimensional. This is different from the discrete-time setting, where control sequences are finite-dimensional. In this case, no numerical optimization theory for finite-dimensional problems work.

2. We first find Hopf-Lax theory for HJ PDEs relevant to state-constrained optimal control problems. We propose a general theorem in PDE theory that specifies sufficient conditions under which two different first-order PDEs have the same solution. Based

on the proposed theorem, we prove our Hopf-Lax formulae. We also provide numerical algorithms.

3. We present a reinforcement learning framework for reachability problems. There is a small portion of existing research dealing with reachability in reinforcement learning. This dissertation presents a general reinforcement learning framework that can be applied to other problems.

The organization of this dissertation is as follows. Chapter 2 introduces notations. Chapter 3 deals with two state-constrained reachability problems (SCCIP and SCRAP) and the corresponding HJ PDEs. Chapter 4 presents Hopf-Lax theory for the two state-constrained reachability problems. Chapter 5 presents reinforcement learning for reachability. Chapter 6 concludes the dissertation.

Chapter 2

Nomenclature and Assumptions

The nomenclature and assumptions in this chapter are used in Chapter 3 and 4 that deal with the continuous-time setting. In Chapter 5 dealing with the discrete-time setting, we use independent nomenclature.

This dissertation deals with dynamic systems' zero-sum games or optimal control problems. Since we can consider optimal control problems as an example of zero-sum games by removing a player's contribution to problems, we generally present nomenclature and assumptions for zero-sum games.

Consider a dynamical system:

$$\dot{\mathrm{x}}(s) = f(s, \mathrm{x}(s), \alpha(s), \delta(s)), s \in [t, T], \text{ and } \mathrm{x}(t) = x, \tag{2.1}$$

where $(t, x) \in [0, T] \times \mathbb{R}^n$ are the initial time and state, $\mathrm{x} : [t, T] \to \mathbb{R}^n$ is the state trajectory, $f : [0, T] \times \mathbb{R}^n \times A \times D \to \mathbb{R}^n$ is the dynamics, $A \subset \mathbb{R}^{m_a}, D \subset \mathbb{R}^{m_d}$ are the control constraints, $\alpha \in \mathcal{A}(t), \delta \in \mathcal{D}(t)$ are the control signals, in each, player A controls α and player D controls δ, and the sets of measurable control signals are

$$\begin{aligned} \mathcal{A}(t) &\coloneqq \{\alpha : [t, T] \to A \mid \|\alpha\|_{L^\infty(t,T)} < \infty\}, \\ \mathcal{D}(t) &\coloneqq \{\delta : [t, T] \to D \mid \|\delta\|_{L^\infty(t,T)} < \infty\}. \end{aligned} \tag{2.2}$$

In each zero-sum game, we consider Stackelberg games where one plays first and then the other players later at each time. For player A and D, we have two Stackelberg games. If the two Stackelberg games have the same cost, the unique Nash equilibrium exists.

For the two Stackelberg games, we use the following notations. $H(t)$ is a set of non-anticipative strategies for player D, and $Z(t)$ is a set of non-anticipative strategies for player A. The non-anticipative strategy outputs a control signal for the second player as a reaction to the first player's control signal without using the future information. The non-anticipative strategy has been introduced by Elliott and Kalton [41]:

$$\begin{aligned} H(t) \coloneqq \{&\eta : \mathcal{A}(t) \to \mathcal{D}(t) \mid \forall s \in [t, T] \text{ and } \alpha, \bar{\alpha} \in \mathcal{A}(t), \text{ if } \alpha(\tau) = \bar{\alpha}(\tau) \text{ a.e. } \tau \in [t, s], \\ &\text{then } \eta[\alpha](\tau) = \eta[\bar{\alpha}](\tau) \text{ a.e. } \tau \in [t, s]\}, \end{aligned} \tag{2.3}$$

$$\begin{aligned} Z(t) \coloneqq \{\zeta : \mathcal{D}(t) \to \mathcal{A}(t) \mid \forall s \in [t,T], \delta, \bar{\delta} \in \mathcal{D}(t), \text{ if } \delta(\tau) = \bar{\delta}(\tau) \text{ a.e. } \tau \in [t,s], \\ \text{then } \zeta[\delta](\tau) = \zeta[\bar{\delta}](\tau) \text{ a.e. } \tau \in [t,s]\}. \end{aligned} \tag{2.4}$$

In this dissertation, we use $L = L(t,x,a,d)$ for stage costs, $g = g(t,x)$ for terminal costs, $c = c(t,x)$ for state-constraint functions. For these functions, we assume the following conditions.

Assumption 1 (Lipschitz continuity and compactness)**.**

1. A and D are compact;
2. $f = f(t,x,a,d) : [0,T] \times \mathbb{R}^n \times A \times D \to \mathbb{R}^n$ is Lipschitz continuous in the state for each $(a,d) \in A \times D$:
$$\|f(t,x_1,a,d) - f(t,x_2,a,d)\| \leq L_f \|x_1 - x_2\|; \tag{2.5}$$
3. the stage cost $L = L(t,x,a,d) : [0,T] \times \mathbb{R}^n \times A \times D \to \mathbb{R}$ is Lipschitz continuous in the state for each $(a,d) \in A \times D$:
$$\|L(t,x_1,a,d) - L(t,x_2,a,d)\| \leq L_L \|x_1 - x_2\|, \tag{2.6}$$
and L is also bounded below;
4. A and D are compact and convex;
5. the terminal cost $g = g(t,x) : [0,T] \times \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous in the state:
$$\|g(t,x_1) - g(t,x_2)\| \leq L_g \|x_1 - x_2\|; \tag{2.7}$$
6. the state constraint $c = c(t,x) : [0,T] \times \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous in the state:
$$\|c(t,x_1) - c(t,x_2)\| \leq L_c \|x_1 - x_2\|; \tag{2.8}$$
7. the stage cost (L) and the terminal cost (g) are bounded below.

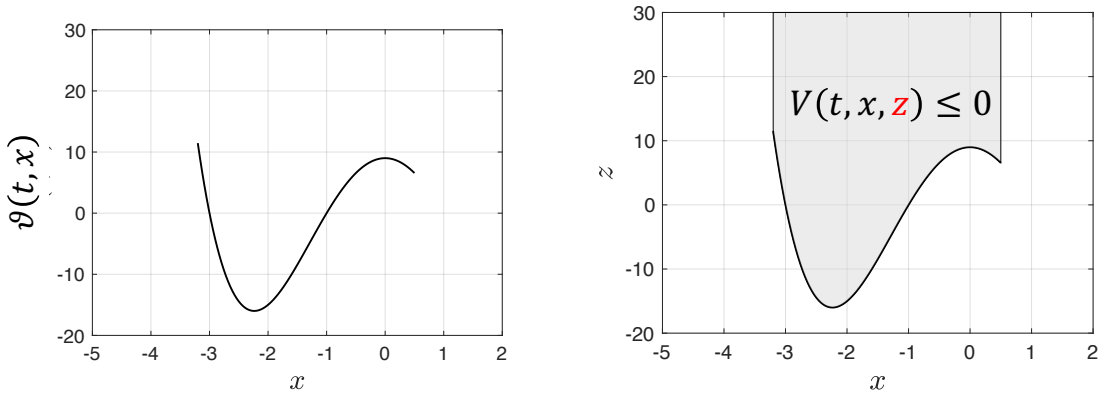
Chapter 3

Hamilton-Jacobi Analysis for the State-Constrained Reachability

The Hamilton-Jacobi (HJ) framework solves optimal control problems or zero-sum games, including reachability problems. In two-player zero-sum games, one player's control signal minimizes a cost while satisfying a state constraint, while the second player's control signal tries either to maximize the cost or to violate the state constraint. An optimal control problem may be considered a special case of the zero-sum game: a control signal that minimizes the given cost while satisfying the constraint is to be determined. HJ analysis builds on dynamic programming and viscosity theory [42]. Dynamic programming is a method to solve problems by recursively breaking them down into sub-problems; viscosity theory provides a notion of weak solutions for first-order PDEs to deal with the non-differentiability of the solution.

Various HJ methods have been developed for state-unconstrained problems and extended to state-constrained problems. Specifically, various HJ frameworks deal with state-unconstrained reachability problems, including the backward reachable tube [10, 73], viability kernel [8, 73], and the reach-avoid problem [74, 46]. For state-constrained HJ PDEs, [97, 98] first introduced a controllability assumption under which the optimal cost of the state-constrained problem is the viscosity solution to the HJ PDE in the constraint set. [59, 47, 24, 48] extended the viscosity theory in [97, 98] for variant state-constrained problems, which satisfy various controllability assumptions. Unfortunately, these controllability assumptions do not hold in practice.

Recent HJ works [4, 17, 16, 49, 36] utilize the epigraphical technique to handle general state-constrained optimal control problems without the controllability assumptions [97, 98, 59, 47, 24, 48]. The epigraphical technique aims to find the epigraph of the optimal cost for the state-constrained problem. These recent works augment a one-dimensional variable into the state and consider a state-unconstrained problem in the higher-dimensional system that encodes the state-constrained problem. Then, these works derive an HJ PDE for the augmented problem whose optimal cost's sub-zero-level set is the epigraph of the optimal cost of the state-constrained problem. The HJ PDE for the augmented problems is well defined in the entire augmented state space, even outside the constraint set; thus, standard viscosity



(a) (b)

Figure 3.1: Given initial time t, (a) shows the optimal cost ϑ for state-constrained problems for each initial state x, and (b) shows the epigraph of ϑ which is characterized by another value function V. V is defined in one-higher-dimensional space (t, x, z), and the epigraph of ϑ is the sub-zero-level set of V.

theory [42] can be applied to characterize the augmented problem as the unique viscosity solution to the HJ PDE. This dissertation presents the epigraphical technique for SCCIP and SCRAP [69, 67].

Figure 3.1 explains the epigraphical technique for state-constrained optimal control or zero-sum game problems. Suppose ϑ is the optimal cost for state-constrained problems for each initial time t. In the infeasible region in x-space, ϑ is not defined, where gradients cannot be defined. In the epigraphical technique, we aim to find continuous V in a one-higher dimensional state space (x, z) so that the sub-zero-level set of V is the epigraph of the ϑ (gray in Figure 3.1 (b)). We find an HJ PDE using the epigraphical technique for not ϑ but V. Gradients of V are well defined in (x, z)-space in a weak sense, and we find the corresponding HJ PDE for V.

The proposed HJ PDEs can deal with both time-varying and time-invariant dynamics, cost, and state constraint. Furthermore, this chapter presents additional HJ PDEs equivalent to the proposed HJ PDEs for the time-invariant case.

Organization

Chapter 3.1 presents HJ formulations for SCCIP. Chapter 3.2 presents HJ formulations for SCRAP. Chapter 3.3 concludes this chapter.

3.1 State-Constrained Control-Invariance Problems

This chapter is based on the work presented in [67], which is joint work with Claire Tomlin. I would like to thank Margaret Chapman for the inspiration for the water reservoir control problem.

This chapter presents SCCIP and provides the corresponding HJ PDE. Our analysis considers two cases. First, time-varying case assumes time-varying costs, dynamics, and state constraints. Second, time-invariant case assumes time-invariant costs, dynamics, and state constraints. Also, this chapter provides and presents analysis for the optimal control signal for each player, numerical algorithm to compute the proposed HJ PDE, as well as a practical example.

Chapter 3.1.1 presents a mathematical formulation for SCCIP. Chapter 3.1.2 presents the HJ PDEs for SCCIP in the time-varying and time-invariant cases. Chapter 3.1.3 presents analysis for an optimal control signal based on the solution to the HJ PDEs. Chapter 3.1.4 presents a numerical algorithm to compute the solution to the HJ PDEs for SCCIP. Chapter 3.1.5 provides a practical example where our HJ formulation can be utilized.

3.1.1 Problem Definition

For the dynamical system (2.1), we define SCCIP as below.

State-constrained control-invariance problem (SCCIP): for given initial time and state (t, x), solve

$$\vartheta_1^+(t,x) \coloneqq \sup_{\eta\in H(t)} \inf_{\alpha\in\mathcal{A}(t)} \max_{\tau\in[t,T]} \int_t^\tau L(s,\mathrm{x}(s),\alpha(s),\eta[\alpha](s))ds + g(\tau,\mathrm{x}(\tau)), \tag{3.1}$$

$$\text{subject to} \quad c(s,\mathrm{x}(s)) \le 0, \quad s\in[t,\tau], \tag{3.2}$$

where x solves (2.1) for $(\alpha, \eta[\alpha])$; and solve

$$\vartheta_1^-(t,x) \coloneqq \inf_{\zeta\in Z(t)} \sup_{\delta\in\mathcal{D}(t)} \max_{\tau\in[t,T]} \int_t^\tau L(s,\mathrm{x}(s),\zeta[\delta](s),\delta(s))ds + g(\tau,\mathrm{x}(\tau)), \tag{3.3}$$

$$\text{subject to} \quad c(s,\mathrm{x}(s)) \le 0, \quad s\in[t,\tau], \tag{3.4}$$

where x solves (2.1) for $(\zeta[\delta], \delta)$.

$H(t)$ is a set of non-anticipative strategies for player D, and $Z(t)$ is a set of non-anticipative strategies for player A. For notations, see Chapter 2.

The difference between ϑ_1^+ and ϑ_1^- is play order. In $\vartheta_1^+(t,x)$, at each time $s\in[t,T]$, player A first plays $\alpha(s)$, and then player D reacts by following its own strategy $\eta[\alpha](s)$. Despite this play order at each time, the choice of player D's strategy comes first since it should be chosen without information about player A's control signal. In other words, player D first chooses its strategy, and then player A chooses its control signal. In $\vartheta_1^-(t,x)$, at each time

s, player D first plays $\delta(s)$, and then player A reacts with its strategy $\zeta[\delta](s)$. Similarly to $\vartheta_1^+(t,x)$, in $\vartheta_1^-(t,x)$, player A first chooses its strategy, and then player D chooses its control signal.

SCCIP is relevant to many practical problems. For SCCIP, consider two water systems where player A controls the water level of pond 1 that is connected to pond 2. Suppose player D is precipitation. Player A needs to minimize the highest water level of pond 1 over time while satisfying constraints for water level of pond 1 and 2 under the worst precipitation assumption.

3.1.2 Hamilton-Jacobi PDEs for SCCIP

HJ PDE for SCCIP in time-varying case

In this chapter, we derive an HJ PDE for SCCIP ($\vartheta_1^\pm$). Unfortunately, for some initial time and state (t,x), there is no control α (or strategy ζ) of player A that satisfies the state constraint for all strategies η of player D (or control signal δ). In this case, $\vartheta_1^\pm(t,x)$ is infinity. Thus, $\vartheta_1^\pm$ is neither continuous nor differentiable in $(0,T)\times\mathbb{R}^n$.

To overcome this issue, we utilize an additional variable $z\in\mathbb{R}$ to define continuous value functions $V_1^\pm$ in (3.5) and (3.6) that combine the cost $\vartheta_1^\pm$ in (3.1) or (3.3), and the constraint in (3.2) or (3.4). This method is called the epigraphical technique because the epigraph of $\vartheta_1^\pm$ is the sub-zero level set of $V_1^\pm$. This method has been utilized to handle state constraints to solve other HJ problems [4, 69]. $V_1^\pm$ is well-defined in $[0,T]\times\mathbb{R}^n\times\mathbb{R}$.

$$V_1^+(t,x,z) \coloneqq \sup_{\eta\in H(t)} \inf_{\alpha\in\mathcal{A}(t)} J_1(t,x,z,\alpha,\eta[\alpha]), \tag{3.5}$$

$$V_1^-(t,x,z) \coloneqq \inf_{\zeta\in Z(t)} \sup_{\delta\in\mathcal{D}(t)} J_1(t,x,z,\zeta[\delta],\delta), \tag{3.6}$$

where cost $J_1:(t,x,z,\alpha,\delta)\to\mathbb{R}$ is defined as follows:

$$J_1(t,x,z,\alpha,\delta) \coloneqq \max_{\tau\in[t,T]} \max\left\{ \max_{s\in[t,\tau]} c(s,\mathrm{x}(s)), \int_t^\tau L(s,\mathrm{x}(s),\alpha(s),\delta(s))ds + g(\tau,\mathrm{x}(\tau)) - z \right\}, \tag{3.7}$$

where x solves (2.1). Define the auxiliary state trajectory z solving

$$\dot{\mathrm{z}}(s) = -L(s,\mathrm{x}(s),\alpha(s),\delta(s)), s\in[t,T], \text{ and } \mathrm{z}(t)=z. \tag{3.8}$$

Then, (2.1) and (3.8) are the joint ODEs whose solution is the augmented state trajectories: $(\mathrm{x},\mathrm{z}):[t,T]\to\mathbb{R}^{n+1}$

$$\begin{bmatrix}\dot{\mathrm{x}}(s)\\ \dot{\mathrm{z}}(s)\end{bmatrix} = \begin{bmatrix} f(s,\mathrm{x}(s),\alpha(s),\delta(s)) \\ -L(s,\mathrm{x}(s),\alpha(s),\delta(s))\end{bmatrix}, s\in[t,T], \begin{bmatrix}\mathrm{x}(t)\\ \mathrm{z}(t)\end{bmatrix} = \begin{bmatrix}x\\ z\end{bmatrix}. \tag{3.9}$$

Then, J_1 in (3.7) becomes

$$\begin{aligned} J_1 &= \max_{\tau\in[t,T]} \max\Big\{ \max_{s\in[t,\tau]} c(s,\mathrm{x}(s)), g(\tau,\mathrm{x}(\tau)) - \mathrm{z}(\tau)\Big\} \\ &= \max\Big\{ \max_{s\in[t,T]} c(s,\mathrm{x}(s)), \max_{\tau\in[t,T]} g(\tau,\mathrm{x}(\tau)) - \mathrm{z}(\tau)\Big\}. \end{aligned} \tag{3.10}$$

The last equality is derived by the distributive property of the maximum operations.

Lemma 1 shows that $\vartheta_1^{\pm}$ can be found if $V_1^{\pm}$ are known. For initial time and state (t,x) for which there is no control or strategy of player A such that the state constraint $(c(s,\mathrm{x}(s)) \le 0, s\in[t,T])$ is satisfied for player D's best control signal or strategy, $V_1^{\pm}(t,x,z)$ is always greater than 0 for all $z\in\mathbb{R}$. In this case, Lemma 1 implies that $\vartheta_1^{\pm}(t,x)$ is infinity.

Lemma 1 (Equivalence of two value functions)**.** Suppose Assumption 1 holds. For all $(t,x)\in[0,T]\times\mathbb{R}^n$, ϑ_1^+ ((3.1) subject to (3.2)), ϑ_1^- ((3.3) subject to (3.4)), V_1^+ in (3.5), and V_1^- in (3.6) have the following relationship.

$$\vartheta_1^{\pm}(t,x) = \min z \text{ subject to } V_1^{\pm}(t,x,z) \le 0. \tag{3.11}$$

This implies that

$$\vartheta_1^+(t,x) = \sup_{\eta\in H(t)} \inf_{\alpha\in\mathcal{A}(t)} \max_{\tau\in[t,T]} \int_t^\tau L(s,\mathrm{x}(s),\alpha(s),\eta[\alpha](s))ds + g(\tau,\mathrm{x}(\tau)), \tag{3.12}$$

$$\text{subject to} \quad c(s,\mathrm{x}(s)) \le 0, \quad s\in[t,T], \tag{3.13}$$

where x solves (2.1) for $(\alpha,\eta[\alpha])$, and

$$\vartheta_1^-(t,x) = \inf_{\zeta\in Z(t)} \sup_{\delta\in\mathcal{D}(t)} \max_{\tau\in[t,T]} \int_t^\tau L(s,\mathrm{x}(s),\zeta[\delta](s),\delta(s))ds + g(\tau,\mathrm{x}(\tau)), \tag{3.14}$$

$$\text{subject to} \quad c(s,\mathrm{x}(s)) \le 0, \quad s\in[t,T], \tag{3.15}$$

where x solves (2.1) for $(\zeta[\delta],\delta)$.

Proof. See Appendix A.1.

The rest of this chapter focuses on the derivation of the corresponding HJ PDE for $V_1^{\pm}$. The HJ PDE is based on the principle of dynamic programming in Lemma 2.

Lemma 2 (Optimality condition)**.** Fix $(t,x,z)\in[0,T]\times\mathbb{R}^n\times\mathbb{R}$. Consider a small step $h>0$ such that $t+h\le T$, V_1^+ (3.5) has the following property:

$$V_1^+(t,x,z) = \sup_{\eta\in H(t)} \inf_{\alpha\in\mathcal{A}(t)} \max\Big\{ \max_{s\in[t,t+h]} c(s,\mathrm{x}(s)), \max_{s\in[t,t+h]} g(\mathrm{x}(s)) - \mathrm{z}(s), V_1^+(t+h,\mathrm{x}(t+h),\mathrm{z}(t+h))\Big\}, \tag{3.16}$$

where (x,z) solves (3.9) for $(\alpha,\eta[\alpha])$. Similarly, for V_1^- (3.6),

$$V_1^-(t,x,z) = \inf_{\zeta\in Z(t)} \sup_{\delta\in\mathcal{D}(t)} \max\Big\{ \max_{s\in[t,t+h]} c(s,\mathrm{x}(s)), \max_{s\in[t,t+h]} g(\mathrm{x}(s)) - \mathrm{z}(s), V_1^-(t+h,\mathrm{x}(t+h),\mathrm{z}(t+h))\Big\}, \tag{3.17}$$

where (x,z) solves (3.9) for $(\zeta[\delta],\delta)$.

Proof. See Appendix A.2.

Theorem 1 presents the corresponding HJ PDEs for $V_1^{\pm}$ in (3.5) and (3.6) using viscosity theory. Intuitively, the HJ PDE in Theorem 1 is derived as h in Lemma 2 converges to zero.

Theorem 1. (HJ PDE for SCCIP) For all $(t, x, z) \in [0, T] \times \mathbb{R}^n \times \mathbb{R}$, $V_1^{\pm}$ in (3.5) and (3.6) is the unique viscosity solution to the HJ PDE:

$$\max\left\{c(t,x) - V_1^{\pm}(t,x,z), g(t,x) - z - V_1^{\pm}(t,x,z), \frac{\partial V_1^{\pm}}{\partial t} - \bar{H}^{\pm}(t,x,z,\frac{\partial V_1^{\pm}}{\partial x}, \frac{\partial V_1^{\pm}}{\partial z})\right\} = 0 \tag{3.18}$$

in $(0, T) \times \mathbb{R}^n \times \mathbb{R}$, where $\bar{H}^{\pm} : [0, T] \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$

$$\bar{H}^{+}(t,x,z,p,q) \coloneqq \max_{a\in A} \min_{d\in D} -p \cdot f(t,x,a,d) + qL(t,x,a,d), \tag{3.19}$$

$$\bar{H}^{-}(t,x,z,p,q) \coloneqq \min_{d\in D} \max_{a\in A} -p \cdot f(t,x,a,d) + qL(t,x,a,d), \tag{3.20}$$

and

$$V_1^{\pm}(T,x,z) = \max\{c(T,x), g(T,x) - z\} \tag{3.21}$$

on $\{t = T\} \times \mathbb{R}^n \times \mathbb{R}$.

Proof. See Appendix A.3.

HJ PDE for SCCIP in time-invariant case

We define the problem as time-invariant if the stage cost, terminal cost, dynamics, and state constraints are all independent of time.

In this chapter, we convert $\vartheta_1^{\pm}$ ((3.1) subject to (3.2) and (3.3) subject to (3.4)) to a fixed-terminal-time problem for the time-invariant case of SCCIP, which allows to utilize methods for the fixed-terminal-time problems [4]. In the fixed-terminal-time problem, optimal control signals of players have to be determined, but the terminal time does not need to be specified but is given.

The conversion of SCCIP to a fixed-terminal-time problem by introducing a freezing control signal $\mu : [t, T] \to [0, 1]$ to the dynamics and a set of freezing control signals:

$$\dot{\mathrm{x}}(s) = f(\mathrm{x}(s), \alpha(s), \delta(s))\mu(s), s \in [t, T], \mathrm{x}(t) = x, \tag{3.22}$$

$$\mathcal{M}(t) \coloneqq \{\mu : [t, T] \to [0, 1] \mid \|\mu\|_{L^{\infty}(t,T)} < \infty\}. \tag{3.23}$$

This freezing control signal controls the contribution of the two players to the system. For example, $\mu(s) = 0$ implies that the state stops at s, and the two players do not contribute to the system. On the other hand, $\mu(s) = 1$ allows the state evolves by the control signals of the players. The maximum over τ operation in SCCIP can be replaced by the maximum over

the freezing control signal if it eliminates contribution of the two players after the maximal terminal time.

Addition to the non-anticipative strategies, the freezing control signal can be added as follows. N_A is a set of non-anticipative strategies for the freezing control to player A, and $\tilde{Z}$ is a set of non-anticipative strategies for player A to player D and the freezing control:

$$\begin{aligned}\mathrm{N}_A(t) \coloneqq \{\nu_A : \mathcal{A}(t) \to \mathcal{M}(t) \mid \forall s \in [t,T], \alpha, \bar{\alpha} \in \mathcal{A}(t), \text{ if } \alpha(\tau) = \bar{\alpha}(\tau) \text{ a.e. } \tau \in [t,s], \\ \text{then } \nu_A[\alpha](\tau) = \nu_A[\bar{\alpha}](\tau) \text{ a.e. } \tau \in [t,s]\}, \end{aligned} \tag{3.24}$$

$$\begin{aligned}\tilde{Z}(t) \coloneqq \{\tilde{\zeta} : \mathcal{D}(t) \times \mathcal{M}(t) \to \mathcal{A}(t) \mid \forall s \in [t,T], \delta, \bar{\delta} \in \mathcal{D}(t), \mu, \bar{\mu} \in \mathcal{M}(t), \\ \text{if } \delta(\tau) = \bar{\delta}(\tau), \mu(\tau) = \bar{\mu}(\tau) \text{ a.e. } \tau \in [t,s], \text{then } \tilde{\zeta}[\delta,\mu](\tau) = \tilde{\zeta}[\bar{\delta},\bar{\mu}](\tau) \text{ a.e. } \tau \in [t,s]\}. \end{aligned} \tag{3.25}$$

We present fixed-terminal-time problems as below:

$$\tilde{\vartheta}_1^+(t,x) \coloneqq \sup_{\eta \in H(t), \nu_A \in \mathrm{N}_A(t)} \inf_{\alpha \in \mathcal{A}(t)} \int_t^T L(\mathrm{x}(s), \alpha(s), \eta[\alpha](s)) \nu_A[\alpha](s) ds + g(\mathrm{x}(T)), \tag{3.26}$$

$$\text{subject to } c(\mathrm{x}(s)) \le 0, s \in [t,T], \tag{3.27}$$

where x solves (3.22) for $(\alpha, \eta[\alpha], \nu_A[\alpha])$;

$$\tilde{\vartheta}_1^-(t,x) \coloneqq \inf_{\tilde{\zeta} \in \tilde{Z}(t)} \sup_{\delta \in \mathcal{D}(t), \mu \in \mathcal{M}(t)} \int_t^T L(\mathrm{x}(s), \tilde{\zeta}[\delta,\mu](s), \delta(s)) \mu(s) ds + g(\mathrm{x}(T)), \tag{3.28}$$

$$\text{subject to } c(\mathrm{x}(s)) \le 0, s \in [t,T], \tag{3.29}$$

where x solves (3.22) for $(\tilde{\zeta}[\delta,\mu], \delta, \mu)$.

After introducing the auxiliary variable $z \in \mathbb{R}$, define cost $\tilde{J}$ by combining the cost and the constraint of $\tilde{\vartheta}_1^{\pm}$:

$$\tilde{J}(t,x,z,\alpha,\delta,\mu) \coloneqq \max\{ \max_{s \in [t,T]} c(\mathrm{x}(s)), g(\mathrm{x}(T)) - \mathrm{z}(T)\}, \tag{3.30}$$

where (x, z) solves, for $s \in [t,T]$,

$$\begin{bmatrix} \dot{\mathrm{x}}(s) \\ \dot{\mathrm{z}}(s) \end{bmatrix} = \begin{bmatrix} f(\mathrm{x}(s), \alpha(s), \delta(s)) \\ -L(\mathrm{x}(s), \alpha(s), \delta(s)) \end{bmatrix} \mu(s), \quad \begin{bmatrix} \mathrm{x}(t) \\ \mathrm{z}(t) \end{bmatrix} = \begin{bmatrix} x \\ z \end{bmatrix}. \tag{3.31}$$

Lemma 3 claims that the zero-sum games whose cost is $\tilde{J}$ are equivalent to $V_1^{\pm}$ in (3.5) and (3.6), which corresponds to $\vartheta_1^{\pm}$.

Lemma 3. Consider $V_1^{\pm}$ in (3.5) and (3.6), and $\tilde{J}$ in (3.30). For all $(t,x,z) \in [0,T] \times \mathbb{R}^n \times \mathbb{R}$,

$$V_1^+(t,x,z) = \sup_{\substack{\eta \in H(t), \\ \nu_A \in \mathrm{N}_A(t)}} \inf_{\alpha \in \mathcal{A}(t)} \tilde{J}(t,x,z,\alpha,\eta[\alpha],\nu_A[\alpha]), \tag{3.32}$$

$$V_1^-(t,x,z) = \inf_{\tilde{\zeta} \in \tilde{Z}(t)} \sup_{\substack{\delta \in \mathcal{D}(t), \\ \mu \in \mathcal{M}(t)}} \tilde{J}(t,x,z,\tilde{\zeta}[\delta,\mu],\delta,\mu). \tag{3.33}$$

Proof. See Appendix A.4.

Corollary 1. (Equivalent fixed-terminal-time game to the time-invariant SCCIP)

$$\vartheta_1^{\pm} \equiv \tilde{\vartheta}_1^{\pm}, \tag{3.34}$$

where ϑ_1^+ is (3.1) subject to (3.2), ϑ_1^- is (3.3) subject to (3.4), $\tilde{\vartheta}_1^+$ is (3.26) subject to (3.27), and $\tilde{\vartheta}_1^-$ is (3.28) subject to (3.29).

Proof. Let the right hand terms in (3.32) and (3.33) be denoted as $W_1^{\pm}$. By Corollary 5.3 in [4], $\tilde{\vartheta}_1^{\pm}(t,x) = \min z$ subject to $W_1^{\pm}(t,x,z) \leq 0$. This fact and Lemma 1 allow us to conclude (3.34). □

This corollary remarks that the free-terminal-time games ($\vartheta_1^{\pm}$) can be converted to fixed-terminal-time games ($\tilde{\vartheta}_1^{\pm}$), in which only control signals and strategies have to be specified, since the terminal time is fixed.

In Lemma 3, $V_1^{\pm}$ is converted to a fixed-terminal-time game, whose corresponding HJ PDE has been investigated in [4]. This allows us to derive an HJ PDE for the time-invariant SCCIP in Theorem 2.

Theorem 2. (HJ PDE for SCCIP in time-invariant case) Consider SCCIP for the time-invariant case. For all $(t,x,z) \in [0,T] \times \mathbb{R}^n \times \mathbb{R}$, $V_1^{\pm}$ in (3.5) and (3.6) is the unique viscosity solution to the HJ PDE:

$$\max\Big\{c(x) - V_1^{\pm}(t,x,z), \frac{\partial V_1^{\pm}}{\partial t} - \min\{0, \bar{H}^{\pm}(x,z,\frac{\partial V_1^{\pm}}{\partial x}, \frac{\partial V_1^{\pm}}{\partial z})\}\Big\} = 0 \tag{3.35}$$

in $(0,T) \times \mathbb{R}^n \times \mathbb{R}$, where $\bar{H}^+$ and $\bar{H}^-$ are defined in (3.19) and (3.20), respectively, without the time dependency, and

$$V_1^{\pm}(T,x,z) = \max\{c(x), g(x) - z\} \tag{3.36}$$

on $\{t = T\} \times \mathbb{R}^n \times \mathbb{R}$.

Proof. See Appendix A.5.
Note that the Hamiltonian $\bar{H}^{\pm}$ in (3.35) is time-invariant.

We observe that the right two terms in the HJ PDE (3.18) $\max\{g - z - V_1^{\pm}, \frac{\partial V_1^{\pm}}{\partial t} - \bar{H}^{\pm}\}$ become $\frac{\partial V_1^{\pm}}{\partial t} - \min\{0, \bar{H}^{\pm}\}$ in (3.35). Note that these two terms are not algebraically equal.

HJ PDE for SCCIP (optimal control setting)

In this chapter, we solve SCCIP in the optimal control problem setting: for given initial time and state (t,x),

$$\vartheta_1(t,x) \coloneqq \inf_{\alpha \in \mathcal{A}(t)} \max_{\tau \in [t,T]} \int_t^{\tau} L(s, \mathrm{x}(s), \alpha(s)) ds + g(\tau, \mathrm{x}(\tau)), \tag{3.37}$$

$$\text{subject to} \quad c(s, \mathrm{x}(s)) \leq 0, \quad s \in [t, \tau], \tag{3.38}$$

where x solves

$$\dot{\mathrm{x}}(s) = f(s, \mathrm{x}(s), \alpha(s)), s \in [t, T], \text{ and } \mathrm{x}(t) = x. \tag{3.39}$$

Chapters 3.1.2 and 3.1.2 present the HJ PDEs for SCCIP in the zero-sum game setting. By removing player D in the zero-sum game, we can get HJ PDEs for SCCIP in the optimal control setting. Thus, Theorem 1 and 2 imply the following remark.

Remark 1. (HJ PDE for SCCIP in optimal control setting) Let V_1 be the unique viscosity solution to the HJ PDE:

$$\max\Big\{c(t,x) - V_1(t,x,z), g(t,x) - z - V_1(t,x,z), \frac{\partial V_1}{\partial t} - \bar{H}(t,x,z,\frac{\partial V_1}{\partial x}, \frac{\partial V_1}{\partial z})\Big\} = 0 \tag{3.40}$$

in $(0,T) \times \mathbb{R}^n \times \mathbb{R}$, where $\bar{H} : [0,T] \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$

$$\bar{H}(t,x,z,p,q) \coloneqq \max_{a \in A} -p \cdot f(t,x,a) + qL(t,x,a), \tag{3.41}$$

and

$$V_1(T,x,z) = \max\{c(T,x), g(T,x) - z\} \tag{3.42}$$

on $\{t = T\} \times \mathbb{R}^n \times \mathbb{R}$. Then,

$$\vartheta_1(t,x) = \min z \text{ subject to } V_1(t,x,z) \leq 0, \tag{3.43}$$

where ϑ_1 is (3.37) subject to (3.38).

If SCCIP is time-invariant, V_1 is the unique viscosity solution to the HJ PDE:

$$\max\Big\{c(x) - V_1(t,x,z), \frac{\partial V_1}{\partial t} - \min\{0, \bar{H}(x,z,\frac{\partial V_1}{\partial x}, \frac{\partial V_1}{\partial z})\}\Big\} = 0 \tag{3.44}$$

in $(0,T) \times \mathbb{R}^n \times \mathbb{R}$, where $\bar{H}$ is defined in (3.41) with ignoring the time dependency, and

$$V_1(T,x,z) = \max\{c(x), g(x) - z\} \tag{3.45}$$

on $\{t = T\} \times \mathbb{R}^n \times \mathbb{R}$.

3.1.3 Optimal Policy and Strategy

The optimal control signal or strategy for SCCIP specified by the HJ PDEs in Chapter 3.1.2. This chapter utilizes the HJ PDEs in Theorem 1, and the method in this chapter can be simply extended for the other HJ PDEs in Theorems 2, and Remarks 1.

Recall $V_1^{\pm}$ defined in (3.5), (3.6), and suppose $V_1^{\pm}$ is computed from the HJ PDEs in Theorems 1.

Lemmas 3 imply the following remark.

Remark 2 (Find $\vartheta_1^{\pm}$ from $V_1^{\pm}$)**.** For initial time $t = 0$ and state $x \in \mathbb{R}^n$,

$$(\mathrm{x}_*(0), \mathrm{z}_*(0)) = (x, \vartheta_1^{\pm}(0, x)), \tag{3.46}$$

where $(\mathrm{x}_*, \mathrm{z}_*)$ is an optimal trajectory for $V_1^{\pm}$.

With the initial augmented state $(\mathrm{x}_*(0), \mathrm{z}_*(0))$, the optimal control and strategy can be found at $(t, \mathrm{x}_*(t), \mathrm{z}_*(t))$, and the optimal state trajectory is also updated by solving the ODE (3.9).

Define $\tilde{H}_1^{\pm} : A \times D \to \mathbb{R}$ for a fixed $(t, x, z) \in (0, T) \times \mathbb{R}^n \times \mathbb{R}$

$$\begin{aligned} \tilde{H}_1^{\pm}(a, d) \coloneqq & - D_x V_1^{\pm}(t, x, z) \cdot f(t, x, a, d) \\ & + D_z V_1^{\pm}(t, x, z) L(t, x, a, d), \end{aligned} \tag{3.47}$$

thus

$$\bar{H}^{+}(t, x, z, D_x V_1^{+}, D_z V_1^{+}) = \max_{a \in A} \min_{d \in D} \tilde{H}_1^{+}(a, d), \tag{3.48}$$

$$\bar{H}^{-}(t, x, z, D_x V_1^{-}, D_z V_1^{-}) = \min_{d \in D} \max_{a \in A} \tilde{H}_1^{-}(a, d), \tag{3.49}$$

where $\bar{H}^{+}$ and $\bar{H}^{-}$ are defined in (3.19) and (3.20), respectively. In this chapter, we omit (t, x, z) to simplify notation. Using the notation with $\tilde{H}_1^{\pm}$ (3.47), the HJ PDE (3.18) for V_1^{+} is equal to

$$\max\{c - V_1^{+}, g - z - V_1^{+}, \frac{\partial V_1}{\partial t}^{+} - \max_{a \in A} \min_{d \in D} \tilde{H}_1^{+}(a, d)\} = 0. \tag{3.50}$$

The HJ PDE implies that optimal control signal or strategy is determined by the gradient information at the current time and state (t, x, z), but the past history of the state trajectory and optimal control signals is not necessary. For example, in V_1^{+}, $\alpha_*(t) = a_*$, $\delta_*[\alpha_*](t) = d_*$ where a_* and d_* are solutions to

$$\max\{c - V_1^{+}, g - z - V_1^{+}, \frac{\partial V_1}{\partial t}^{+} - \tilde{H}_1^{+}(a, d)\} = 0. \tag{3.51}$$

at (t, x, z). In other words, it is sufficient to specify optimal controls for player A and D in $(0, T) \times \mathbb{R}^n \times \mathbb{R}$ to generate the optimal control signal or strategy. The maxmini solution (a_*, d_*) for the Hamiltonian $\bar{H}_V^{+}$ $(\max_{a \in A} \min_{d \in D} \tilde{H}_1^{+}(a, d))$ is certainly optimal, but there are more solutions. Similarly, for V_1^{-}, any pair of (a_*, d_*) satisfying the corresponding HJ PDE is optimal.

In the HJ PDE (3.50) (or (3.18)) for $V_1^{\pm}$, we have three terms: $c - V_1^{\pm}$, $g - z - V_1^{\pm}$, or $\frac{\partial V_1^{\pm}}{\partial t} - \bar{H}^{\pm}$, and, at least, one of these terms is zero. By considering which term is bigger or smaller among the three terms, all possible optimal controls for $\vartheta_1^{\pm}$ $(V_1^{\pm})$ is derived in Remark 3.

Although (a_*, d_*) satisfying the HJ PDE is optimal, we need to consider the order of players: player A plays first in V_1^+ but player D plays first in V_1^-. In V_1^+, we first find a set of optimal control for player A, and then investigate a set of optimal control for player D when player A applies its optimal control. On the other hand, in V_1^-, we first investigate a set of optimal control for player D, and then find a set of optimal control for player A after applying an optimal control of player D. Based on this argument, Remark 3 presents optimal controls for V_1^+ according to classification, and optimal controls for V_1^- can be analogously extended.

Remark 3 (Optimal controls for V_1^+)**.** Fix $(t, x, z) \in (0, T) \times \mathbb{R}^n \times \mathbb{R}$.

Optimal controls for ϑ_1^+ (V_1^+) are the following:

1. Case 1: $\max\{c - V_1^+, g - z - V_1^+\} \geq \frac{\partial V_1}{\partial t}^+ - \bar{H}^+$

$$a_* \in \{a \in A \mid \frac{\partial V_1}{\partial t}^+ - \min_{d \in D} \tilde{H}_1^+(a, d) \leq 0\}, \tag{3.52}$$

$$d_* \in D; \tag{3.53}$$

2. Case 2: $\max\{c - V_1^+, g - z - V_1^+\} < \frac{\partial V_1}{\partial t}^+ - \bar{H}^+$

$$a_* \in \arg\max_{a \in A} \min_{d \in D} \tilde{H}_1^+(a, d), \tag{3.54}$$

$$d_* \in \arg\min_{d \in D} \tilde{H}_1^+(a_*, d). \tag{3.55}$$

3.1.4 Numerical Computation for the Hamilton-Jacobi PDE

In this chapter, we present a numerical algorithm based on level set methods [78] to compute the solutions to the two HJ PDEs for SCCIP. Algorithm 1 deals with the HJ PDEs for SCCIP ((3.18), (3.35), (3.40), (3.44)). Level set methods have been utilized to solve a variety of HJ formulations [79, 46, 69].

Algorithm 1 solves the HJ PDE (3.18) in two steps. At line 6 in Algorithm 1, we first compute the HJ PDE ($\frac{\partial V_1^\pm}{\partial t} - \bar{H}^\pm(t, x, z, \frac{\partial V_1^\pm}{\partial x}, \frac{\partial V_1^\pm}{\partial z}) = 0$), and line 7 in Algorithm 1 replaces $V_1^\pm$ by one of $c(t, x)$, $g(t, x) - z$ or itself to satisfy the HJ PDE (3.18).

For solving the HJ PDE at step 1, the Lax-Friedrichs scheme [37] is utilized on the temporal discretization $\{t_0 = 0, ..., t_K = T\}$ and the spatial discretization $\{(x_0, z_0), ..., (x_N, z_N)\} \subset \mathbb{R}^n \times \mathbb{R}$:

$$V_1^\pm(t_k, x_i, z_i) = V_1^\pm(t_{k+1}, x_i, z_i) - \Delta_k \hat{\bar{H}}^\pm(\phi_x^+, \phi_x^-, \phi_z^+, \phi_z^-), \tag{3.56}$$

where $\Delta_k = t_{k+1} - t_k$, $(\phi_x^\pm, \phi_z^\pm)$ are numerical approximation for $(\frac{\partial V_1^\pm}{\partial x}, \frac{\partial V_1^\pm}{\partial z})$ (gradients with respect to (x, z) at (t_{k+1}, x_i, z_i)), and

$$\hat{\bar{H}}^\pm(\phi_x^+, \phi_x^-, \phi_z^+, \phi_z^-) = \bar{H}^\pm(t_{k+1}, x_i, z_i, \frac{\phi_x^+ + \phi_x^-}{2}, \frac{\phi_z^+ + \phi_z^-}{2}) - \alpha_x \cdot \frac{\phi_x^+ + \phi_x^-}{2} - \alpha_z \frac{\phi_z^+ + \phi_z^-}{2}, \tag{3.57}$$

where $\hat{H}^{\pm}$ are defined in (3.19) and (3.20), and $\alpha_x = (\alpha_{x_1}, ..., \alpha_{x_n})$ ($\alpha_{x_i} = \max|D_{p_i}\bar{H}^{\pm}|$) and $\alpha_z (= \max|D_q\bar{H}^{\pm}|)$ are dissipation coefficients for numerical viscosity, based on the partial derivatives of $\bar{H}^{\pm}$ [85]. The fifth-order accurate HJ WENO (weighted essentially nonoscillatory) method [85] is used for the gradient $\phi_x^{\pm}, \phi_z^{\pm}$. In (3.56), the first-order Euler method is used for the temporal partial derivative, but higher-order methods, such as third-order accurate TVD (total variation diminishing) Runge-Kutta (RK) [94] can be used. [85] provided the empirical observation that level set methods are sensitive to spatial accuracy, thus high-order scheme for spatial derivatives is desired, but high-order approximation for temporal derivatives does not significantly increase the accuracy.

For the time-invariant SCCIP, line 9 in Algorithm 1 solves the HJ PDE (3.35) whose Hamiltonian has the minimum with 0 operation. Then, line 10 in Algorithm 1 updates $V_1^{\pm}$ with the maximum between c and itself to satisfy the HJ PDE (3.35) without considering $g - z$ term.

For SCCIP in optimal control setting, Algorithm 1 also works with utilizing the correct Hamiltonian $\bar{H}$ (3.41) instead of $\bar{H}^{\pm}$ ((3.19) and (3.20)).

Algorithm 1 Computing the solution $V_1^{\pm}$ or V_1 to the HJ PDEs for SCCIP in the zero-sum game and optimal control settings. This algorithm deals with the four HJ PDEs: (3.18), (3.35), (3.40), and (3.44).

1: **Input:** the temporal discretization: $\{t_0 = 0, ..., t_K = T\}$, the spatial discretization: $\{(x_0, z_0), ..., (x_N, z_N)\}$
2: **Output:** $V_1^{\pm}$ (or V_1)
3: $V_1^{\pm}$(or V_1)$(T, x_i, z_i) \leftarrow \max\{c(T, x_i), g(T, x_i) - z_i\}, \forall i$
4: **for** $k \in \{K-1, ..., 0\}$ **do**
5: **case** solving the HJ PDEs (3.18) or (3.40)
6: $V_1^{\pm}$(or V_1)$(t_k, x_i, z_i) \leftarrow V_1^{\pm}$(or V_1)$(t_{k+1}, x_i, z_i) - \Delta_k \hat{\bar{H}}^{\pm}$(or $\hat{\bar{H}}$)$(\phi_x^+, \phi_x^-, \phi_z^+, \phi_z^-), \forall i$
7: $V_1^{\pm}$(or V_1)$(t_k, x_i, z_i) \leftarrow \max\{c(t_k, x_i), g(t_k, x_i) - z_i, V_1^{\pm}$(or V_1)$(t_k, x_i, z_i)\}, \forall i$
8: **case** solving the HJ PDEs (3.35) or (3.44)
9: $V_1^{\pm}$(or V_1)$(t_k, x_i, z_i) \leftarrow V_1^{\pm}$(or V_1)(t_{k+1}, x_i, z_i)
$-\Delta_k \min\{0, \hat{\bar{H}}^{\pm}$(or $\hat{\bar{H}}$)$(\phi_x^+, \phi_x^-, \phi_z^+, \phi_z^-)\}, \forall i$
10: $V_1^{\pm}$(or V_1)$(t_k, x_i, z_i) \leftarrow \max\{c(t_k, x_i), V_1^{\pm}$(or V_1)$(t_k, x_i, z_i)\}, \forall i$

3.1.5 Example

In this example, we solve a zero-sum game for two ponds, as shown in Figure 3.2. This example is motivated by the water system in [26].

Precipitation on pond 1 increases the water level of pond 1, and pond 1 (player A) wants to minimize the highest water level in the time horizon (1 s) by controlling amount of outflow to pond 2. We assume that the water level increasing rate on pond 1 due to the precipitation is unknown but bounded by 0 and 10 m/s. The precipitation is considered as player D. These

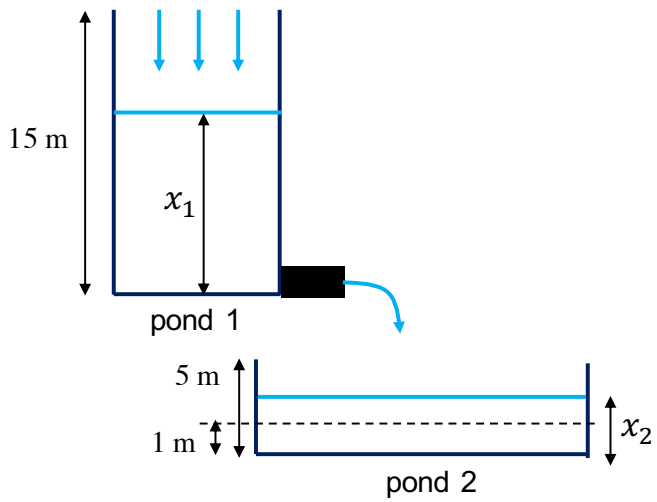


Figure 3.2: Water system with two ponds.

numbers and units can be easily changed to realistic problems. We determine an optimal control signal and strategy even in the worst behavior of player D.

In the water system, we have two states: x_1 and x_2 represent the water level of pond 1 and 2. The state trajectories are solving the following dynamics:

$$\begin{aligned} \dot{\mathrm{x}}_1(s) &= \delta(s) - \sqrt{2g\mathrm{x}_1(s)}\alpha(s), \\ \dot{\mathrm{x}}_2(s) &= 0.5\sqrt{2g\mathrm{x}_1(s)}\alpha(s) - 0.5\mathrm{x}_2(s), \end{aligned} \tag{3.58}$$

where $\alpha(s) \in A = [0, 1]$, $\delta(s) \in D = [0, 10]$, and g is the gravitational constant: 9.81 m/s^2. In the dynamics for x_1, the first term δ is by the precipitation (player D), and the second term $\sqrt{2g\mathrm{x}_1}\alpha$ is the water level decreasing rate by pond 1 (player A). The term $\sqrt{2g\mathrm{x}_1}$ is by Bernoulli's equation, and pond 1 controls the area of outflows (α) between 0 and 1. We set the bottom area of pond 2 is twice bigger than pond 1, thus the dynamics for x_2 contains $0.5\sqrt{2g\mathrm{x}_1}\alpha$. Also, we assume that pond 2's water is used for drinking water, which causes a decreasing rate $0.5\mathrm{x}_2$.

The dynamics (3.58) is not Lipschitz at $x_1 = 0$. To avoid this, we approximate $\sqrt{2gx_1}$ with a sinusoidal function $4.82\sin(1.17x_1)$ if x_1 is less than 1. This sinusoidal-approximate function has the same value and first derivative at $x_1 = 1$: $\sqrt{2g} \simeq 4.82\sin(1.17)$ and $\sqrt{g/2} \simeq 4.82 * 1.17 * \cos(1.17)$. The approximated (Lipschitz) dynamics are

$$\begin{aligned} \dot{\mathrm{x}}_1(s) &= \delta(s) - \begin{cases} \sqrt{2g\mathrm{x}_1(s)}\alpha(s), & \mathrm{x}_1(s) \geq 1, \\ 4.82\sin(1.17\mathrm{x}_1(s))\alpha(s), & \mathrm{x}_1(s) < 1, \end{cases} \\ \dot{\mathrm{x}}_2(s) &= \begin{cases} 0.5\sqrt{2g\mathrm{x}_1(s)}\alpha(s), & \mathrm{x}_1(s) \geq 1, \\ 2.41\sin(1.17\mathrm{x}_1(s))\alpha(s), & \mathrm{x}_1(s) < 1, \end{cases} - 0.5\mathrm{x}_2(s), \end{aligned} \tag{3.59}$$

We solve the two zero-sum games: the upper value function is

$$\vartheta_1^+(0,x_1,x_2), = \sup_{\eta\in H(0)}\inf_{\alpha\in\mathcal{A}(0)}\max_{\tau\in[0,1]}\mathrm{x}_1(\tau), \tag{3.60}$$

$$\text{subject to } \max\{|\mathrm{x}_1(s)-7.5|-7.5, |\mathrm{x}_2(s)-3|-2\}\le 0, \tag{3.61}$$

where $A=[0,1]$, $D=[0,10]$, $\mathcal{A}(0)=\{[0,1]\to A \mid \|\alpha\|_{L^\infty(0,1)}<\infty\}$, $\mathcal{D}(0)=\{[0,1]\to D \mid \|\delta\|_{L^\infty(0,1)}<\infty\}$, $H(0)$ is a set of non-anticipative strategies for player D (pond 2) as in (2.3), and $(\mathrm{x}_1,\mathrm{x}_2)$ solves (3.59) for $(\alpha,\eta[\alpha])$; and the lower value function is

$$\vartheta_1^-(0,x_1,x_2), = \inf_{\zeta\in Z(0)}\sup_{\delta\in\mathcal{D}(0)}\max_{\tau\in[0,1]}\mathrm{x}_1(\tau), \tag{3.62}$$

$$\text{subject to } \max\{|\mathrm{x}_1(s)-7.5|-7.5, |\mathrm{x}_2(s)-3|-2\}\le 0, \tag{3.63}$$

where $Z(0)$ is a set of non-anticipative strategies for player A (pond 1) as in (2.4), and $(\mathrm{x}_1,\mathrm{x}_2)$ solves (3.59) for $(\zeta[\delta],\delta)$. The state constraint implies that the water level of pond 1 has to be between 0 and 15 m and the one of pond 2 is between 1 and 5 m. In these games, pond 1 (player A) wants to minimize the worst water level of pond 1 in the time horizon while satisfying the state constraint for preventing flood in pond 1 and 2.

We will solve the HJ PDE (3.18) for $V_1^\pm$ corresponding to $\vartheta_1^\pm$ ((3.61) or (3.63)). We have the Hamiltonian

$$\begin{aligned}
\bar{H}^+(t,x,z,p,q) &= \max_{a\in A}\min_{d\in D} -p_1 d + 0.5p_2x_2 \\
&\quad + \begin{cases} (p_1-0.5p_2)\sqrt{2gx_1}a & \text{if } x_1\ge 1 \\ (p_1-0.5p_2)4.82\sin(1.17x_1)a & \text{if } x_1<1 \end{cases} \\
&= \begin{cases} -10p_1 & \text{if } p_1\ge 0 \\ 0 & \text{if } p_1<0 \end{cases} + 0.5p_2x_2 \\
&\quad + \begin{cases} (p_1-0.5p_2)\sqrt{2gx_1} & \text{if } \begin{array}{l} p_1-0.5p_2\ge 0 \\ x_1\ge 1 \end{array} \\ (p_1-0.5p_2)4.82\sin(1.17x_1) & \text{if } \begin{array}{l} p_1-0.5p_2\ge 0 \\ x_1<1 \end{array} \\ 0 & \text{if } p_1-0.5p_2<0 \end{cases} \\
&= \bar{H}^-(t,x,z,p,q)
\end{aligned} \tag{3.64}$$

where $x=(x_1,x_2)\in\mathbb{R}^2$ and $p=(p_1,p_2)\in\mathbb{R}^2$. (3.64) implies

$$V_1^+\equiv V_1^-\equiv V_1^\pm \text{ and } \vartheta_1^+\equiv\vartheta_1^-\equiv\vartheta_1^\pm. \tag{3.65}$$

We use $V_1^\pm$ to denote the same value functions V_1^+ and V_1^-.

The red curvature in Figure 3.3 (a) shows the zero-level set of $V_1^\pm(0,x_1,x_2)$, numerically computed by Algorithm 1. This algorithm is programmed by utilizing the `level set`

`toolbox` [78] and the `helperOC toolbox` [30] in `Matlab`, and this simulation is carried out on a laptop with a 2.8 GHz Quad-Core i7 CPU and 16 GB RAM. Each of x_1, x_2, and z axis has 81 discretization points, and the time interval $[0, 1]$ is discretized with 201 points. The computation time for $V_1^{\pm}$ is 237 s. In Figure 3.3 (a), the value of $V_1^{\pm}$ inside of the red curvature is negative, on the other hand, the value outside of the curvature is positive.

This example is time-invariant, so both HJ PDEs in (3.18) and (3.35) can be utilized. In this example, we solve the HJ PDE (3.18).

Lemma 1 describes how to compute $\vartheta_1^{\pm}$ from the zero-level set of $V_1^{\pm}$, which is illustrated in Figure 3.3. Figure 3.3 (a) illustrates the intersection of the zero-level set of $V_1^{\pm}$ and each z-level plane, and Figure 3.3 (b) shows these intersections in the state space, (x_1, x_2): the z-level sets on the zero-level set of $V_1^{\pm}$. As illustrated in Figure 3.3 (b), the lower z-level is achieved in the smaller region in (x_1, x_2). In this example, as z-level is increasing, the inner area of the z-level set on the subzero-level set of $V_1^{\pm}$ is increasing and also converging at the z-level of 15, which is the outer curvature in Figure 3.3 (b). For (x_1, x_2) outside of the outer curvature indicated with $z \geq 15$, there is no control signal or strategy for pond 1 (player A) to satisfy the state constraint, which implies that $\vartheta_1^{\pm}(0, x_1, x_2)$ is infinity. On the other hand, for (x_1, x_2) on a unique z-level set, the z-level is equal to $\vartheta_1^{\pm}$. For example, the z-level set of 6 is the only z-level set passing through $(2.6, 2)$. In this case, $\vartheta_1^{\pm}(0, 2.6, 2) = 6$. On the other hand, for (x_1, x_2) on multiple z-level sets, the minimum value of z-level is $\vartheta_1^{\pm}$. For example, $(0.05, 2)$ is on the z-level sets of any number greater than or equal to 4.5. In this case, $\vartheta_1^{\pm}(0, 0.05, 2)$ is 4.5 since 4.5 is the minimum z-level that contains the point $(0.05, 2)$.

Using the value function $V_1^{\pm}$ and $\vartheta_1^{\pm}$, the method presented in Chapter 3.1.3 provides a state trajectory and an optimal control and strategy for the two players (pond 1 and the precipitation). Among multiple solutions for optimal control and strategy presented in Remark 3, we choose

$$\begin{aligned} a_* &\in \arg\max_{a \in A} \min_{d \in D} \tilde{H}_1^+(a, d), \\ d_* &\in \arg\min_{d \in D} \tilde{H}_1^+(a_*, d), \end{aligned} \tag{3.66}$$

which satisfies (3.52) to (3.55) since $\bar{H}^+ = \bar{H}^-$ and $\tilde{H}_1^+ = \tilde{H}_1^-$, where $\tilde{H}_1^{\pm}$ is equal to $-\frac{\partial V_1^{\pm}}{\partial x} \cdot f + \frac{\partial V_1^{\pm}}{\partial z} L$ as defined in (3.47).

Figure 3.4 shows state trajectories for two different initial states: $(x_1, x_2) = (10, 4)$ and $(2, 1.1)$. As shown in Figure 3.4 (a), for the initial state $(10, 4)$, x_2 hits the boundary of the state constraint: $\mathrm{x}_2(1) = 5$, and x_1 is maximized at $t = 1$. Since the initial water levels of the two ponds are high, the precipitation (player D) tries to increase the water level of pond 1 for all time, but player A tries to balance the water levels of the two ponds. On the other hand, for the initial state $(2, 1.1)$, Figure 3.4 (b) shows that x_2 strictly satisfies the state constraint $[1, 5]$. x_1 is maximized at $t = 0.015$ and increasing for the later time. Since the initial water levels of the two ponds are low, the precipitation (player D) tries to violate the state constraint by not increasing the water level of pond 1. However, player A tries to

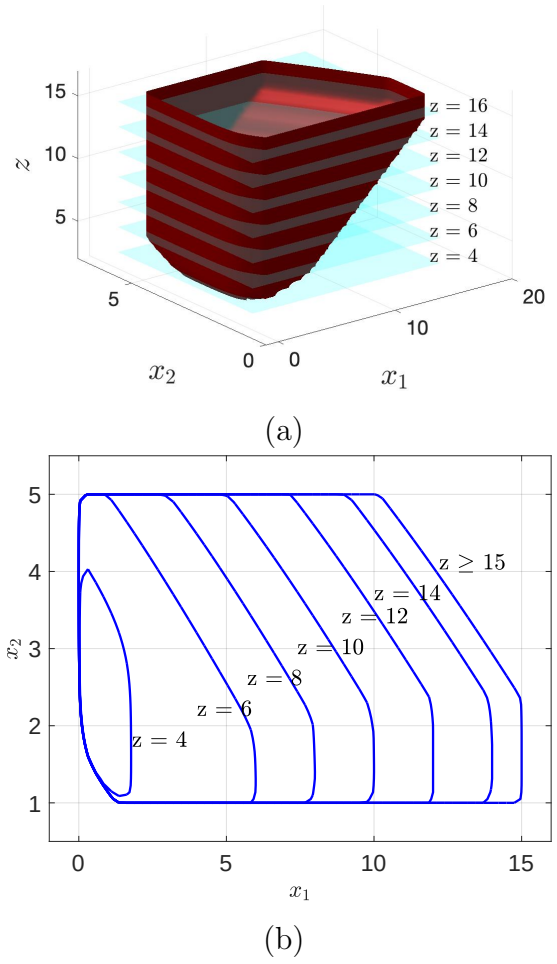


Figure 3.3: (a) The zero-level set of $V_1^{\pm}$ is shown in this figure. The value of $V_1^{\pm}$ inside of the curvature is negative, but the value is positive outside. The blue planes are z-level planes of 4, 6, 8, 10, 12, 14, and 16. (b) The z-level sets are shown in (x_1, x_2)-space. The z-level sets for $z \geq 15$ are the same (the outer curvature). The z-level sets also show $\vartheta_1^{\pm}$ by Lemma 1. For example, for $(1.60, 2.85)$ on the z-level set of 4, $\vartheta_1^{\pm}$ is 4. On the other hand, consider $(0.11, 2)$ on multiple z-level sets from 4.22 to any greather levels, for which $\vartheta_1^{\pm}$ is the minimum z-level that contains the point: $\vartheta_1^{\pm}(0.05, 2) = 4.22$.

balance the two ponds' water level so that all ponds have more water than the minimum levels.

As discussed in Chapter 3.1.4, there are some numerical issues in Algorithm 1. First, we observe that (3.66) provides a bang-bang control, thus the state trajectories are not smooth as shown in Figure 3.4. This happens due to frequent sign change of the gradient along the time horizon. Second, the numerical error on $V_1^{\pm}$ causes inaccurate $\vartheta_1^{\pm}$ by Lemma 1, which

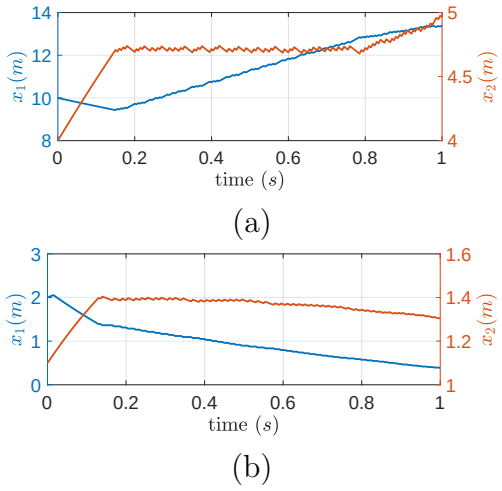


Figure 3.4: State trajectories by applying an optimal control signal and strategy for two players (pond 1 and the precipitation) where the initial states are (a) $(x_1, x_2) = (10, 4)$ and (b) $(x_1, x_2) = (2, 1.1)$.

could potentially cause unsafety even though the violation of the state constraint might be smaller for the smaller grid size. In practice, we suggest having a safety margin to the state constraint: for example, use $c(s, \mathrm{x}(s)) + \epsilon \leq 0$ for small $\epsilon > 0$ instead of $c(s, \mathrm{x}(s)) \leq 0$.

3.2 State-Constrained Reach-Avoid Problems

This chapter is based on the work presented in [69], which is joint work with Alexander Keimer, Alexandre Bayen, and Claire Tomlin.

The organization of this chapter is as follows. Chapter 3.2.1 introduces SCRAP. Chapter 3.2.2 proposes two HJ PDEs for SCRAP in time-varying and time-invariant cases. Chapter 3.2.3 presents optimal controls and strategies. Chapter 3.2.4 presents numerical algorithms to compute the solution to the HJ PDE for SCRAP. Chapter 3.2.5 demonstrates an example for SCRAP and the HJ analysis.

3.2.1 Problem Definition

State-constrained reach-avoid problem (SCRAP): for given initial time and state (t, x), solve

$$\vartheta_2^+(t,x) \coloneqq \sup_{\eta \in H(t)} \inf_{\alpha \in \mathcal{A}(t)} \min_{\tau \in [t,T]} \int_t^\tau L(s, \mathrm{x}(s), \alpha(s), \eta[\alpha](s)) ds + g(\tau, \mathrm{x}(\tau)), \tag{3.67}$$

$$\text{subject to} \quad c(s, \mathrm{x}(s)) \leq 0, \quad s \in [t, \tau], \tag{3.68}$$

where x solves (2.1) for $(\alpha, \eta[\alpha])$; and solve

$$\vartheta_2^-(t,x) \coloneqq \inf_{\zeta\in Z(t)} \sup_{\delta\in\mathcal{D}(t)} \min_{\tau\in[t,T]} \int_t^\tau L(s,\mathrm{x}(s),\zeta[\delta](s),\delta(s))ds + g(\tau,\mathrm{x}(\tau)), \tag{3.69}$$

$$\text{subject to} \quad c(s,\mathrm{x}(s)) \le 0, \quad s\in[t,\tau], \tag{3.70}$$

where x solves (2.1) for $(\zeta[\delta],\delta)$.

3.2.2 Hamilton-Jacobi PDEs for SCRAP

Chapter 3.2.2 presents two HJ PDEs for the time-varying and time-invariant cases. In addition, this chapter presents two HJ PDEs for SCRAP and the time-invariant SCRAP in the optimal control setting.

For SCRAP in the zero-sum game and optimal control settings, the corresponding HJ PDEs have been presented in [69]. This chapter first presents this previous work and then proposes HJ PDEs for the time-invariant version.

HJ PDE for SCRAP

This chapter provides an HJ formulation for SCRAP: solve ϑ_2^+ in (3.67) subject to (3.68) and ϑ_2^- in (3.69) subject to (3.70). For $(t,x,z)\in[0,T]\times\mathbb{R}^n\times\mathbb{R}$, define the augmented value functions corresponding to the upper and lower value functions $(\vartheta_2^\pm)$:

$$V_2^+(t,x,z) \coloneqq \sup_{\eta\in H(t)} \inf_{\alpha\in\mathcal{A}(t)} J_2(t,x,z,\alpha,\eta[\alpha]), \tag{3.71}$$

$$V_2^-(t,x,z) \coloneqq \inf_{\zeta\in Z(t)} \sup_{\delta\in\mathcal{D}(t)} J_2(t,x,z,\zeta[\delta],\delta), \tag{3.72}$$

where cost $J_2 : (t,x,z,\alpha,\delta)\to\mathbb{R}$ is defined as follows:

$$J_2(t,x,z,\alpha,\delta) \coloneqq \min_{\tau\in[t,T]} \max\left\{ \max_{s\in[t,\tau]} c(s,\mathrm{x}(s)), \int_t^\tau L(s,\mathrm{x}(s),\alpha(s),\delta(s))ds + g(\tau,\mathrm{x}(\tau)) - z \right\}, \tag{3.73}$$

where x solves (2.1) for (α,δ). Theorem 3 proved that, for all $(t,x)\in[t,T]\times\mathbb{R}^n$,

$$\vartheta_2^\pm(t,x) = \min z \text{ subject to } V_2^\pm(t,x,z)\le 0, \tag{3.74}$$

and $V_2^\pm$ are the unique viscosity solutions to the HJ PDEs in Theorem 4.

Theorem 3 (Equivalence of value functions)**.** Suppose Assumption 1 holds. The SCRAP costs $\vartheta_2^\pm$ and $V_2^\pm$ ((3.71) or (3.72)) satisfy for given $(t,x)\in(0,T)\times\mathbb{R}^n$

$$\vartheta_2^\pm(t,x) = \min z \quad \text{subject to} \quad V_2^\pm(t,x,z)\le 0. \tag{3.75}$$

Proof. See Appendix B.1.

For the upper value function V_2^+, we consider a trajectory $\mathrm{z}^+ : [t,T] \to \mathbb{R}$ solving the ODE

$$\begin{cases} \dot{\mathrm{z}}^+(s) = -L(s, \mathrm{x}(s), \alpha(s), \eta[\alpha](s)), \quad s \in [t,T], \\ \mathrm{z}^+(t) = z. \end{cases} \tag{3.76}$$

Then, we have

$$V_2^+(t,x,z) \coloneqq \sup_{\eta \in H(t)} \inf_{\alpha \in \mathcal{A}(t)} \min_{\tau \in [t,T]} \max \left\{ \max_{s \in [t,\tau]} c(s, \mathrm{x}(s)), g(\tau, \mathrm{x}(\tau)) - \mathrm{z}^+(\tau) \right\}. \tag{3.77}$$

For the lower value function V_2^-, we consider a trajectory $\mathrm{z}^- : [t,T] \to \mathbb{R}$ solving the ODE

$$\begin{cases} \dot{\mathrm{z}}^-(s) = -L(s, \mathrm{x}(s), \zeta[\delta](s), \delta(s)), \quad s \in [t,T], \\ \mathrm{z}^-(t) = z. \end{cases} \tag{3.78}$$

Then, we have

$$V_2^-(t,x,z) \coloneqq \inf_{\zeta \in Z(t)} \sup_{\delta \in \mathcal{D}(t)} \min_{\tau \in [t,T]} \max \left\{ \max_{s \in [t,\tau]} c(s, \mathrm{x}(s)), g(\tau, \mathrm{x}(\tau)) - \mathrm{z}^-(\tau) \right\}. \tag{3.79}$$

With the augmented state (x,z), similar to Chapter 3.1.3, we utilize [74, 46] to find the corresponding HJ PDE for $V^\pm$.

Theorem 4. (HJ PDE for SCRAP) For all $(t,x,z) \in [0,T] \times \mathbb{R}^n \times \mathbb{R}$, $V_2^\pm$ in (3.71) and (3.72) are the unique viscosity solutions to the HJ PDEs:

$$\max \left\{ c(t,x) - V_2^\pm(t,x,z), \min \left\{ g(t,x) - z - V_2^\pm(t,x,z), V_{2,t}^\pm - \bar{H}^\pm(t,x,z, D_x V_2^\pm, D_z V_2^\pm) \right\} \right\} = 0 \tag{3.80}$$

in $(0,T) \times \mathbb{R}^n \times \mathbb{R}$, where $\bar{H}^\pm$ are defined in (3.19) and (3.20), and

$$V_2^\pm(T,x,z) = \max\{c(T,x), g(T,x) - z\} \tag{3.81}$$

on $\{t = T\} \times \mathbb{R}^n \times \mathbb{R}$.

If $\bar{H}^+ \equiv \bar{H}^-$ in coincidence, $V^+ \equiv V^-$.

We observe that the difference between the two types of HJ PDEs for $V_1^\pm$ and $V_2^\pm$ is that the minimum operation in (3.80) for $V_2^\pm$ is replaced by the maximum operation in (3.18). This is from the difference between $\vartheta_1^\pm$ for SCCIP and $\vartheta_2^\pm$ for SCRAP: $\vartheta_1^\pm$ in (3.1) and (3.3) have $\max_\tau$ operation, and $\vartheta_2^\pm$ in (3.67) and (3.69) have $\min_\tau$ operation.

HJ PDE for SCRAP in time-invariant case

Through similar analysis to that in Chapter 3.1.2, this chapter derives the HJ PDEs for the time-invariant SCRAP. For SCCIP, the freezing control signal $\mu : [t, T] \to [0, 1]$ allows to convert to the fixed-terminal-time problems by replacing the maximum over τ operation in SCCIP to the supremum over the freezing control signal or strategy. Instead, SCRAP is specified in terms of the minimum over τ operation, which will be replaced by the infimum over the freezing control signal or strategy.

For the notations of the freezing control signal and relevant strategies, the same notations in Chapter 3.1.2 is used. M is the set of freezing control signals

$$\mathcal{M}(t) \coloneqq \{\mu : [t, T] \to [0, 1] \mid \|\mu\|_{L^\infty(t,T)} < \infty\}. \tag{3.82}$$

N_A is a set of non-anticipative strategies for the freezing control to player A, and $\tilde{Z}$ is a set of non-anticipative strategies for player A to player D and the freezing control:

$$\begin{aligned}\mathrm{N}_A(t) \coloneqq \{\nu_A : \mathcal{A}(t) \to \mathcal{M}(t) \mid \forall s \in [t, T], \alpha, \bar{\alpha} \in \mathcal{A}(t), \text{ if } \alpha(\tau) = \bar{\alpha}(\tau) \text{ a.e. } \tau \in [t, s], \\ \text{then } \nu_A[\alpha](\tau) = \nu_A[\bar{\alpha}](\tau) \text{ a.e. } \tau \in [t, s]\},\end{aligned} \tag{3.83}$$

$$\begin{aligned}\tilde{Z}(t) \coloneqq \{\tilde{\zeta} : \mathcal{D}(t) \times \mathcal{M}(t) \to \mathcal{A}(t) \mid \forall s \in [t, T], \delta, \bar{\delta} \in \mathcal{D}(t), \mu, \bar{\mu} \in \mathcal{M}(t), \\ \text{if } \delta(\tau) = \bar{\delta}(\tau), \mu(\tau) = \bar{\mu}(\tau) \text{ a.e. } \tau \in [t, s], \text{then } \tilde{\zeta}[\delta, \mu](\tau) = \tilde{\zeta}[\bar{\delta}, \bar{\mu}](\tau) \text{ a.e. } \tau \in [t, s]\}.\end{aligned} \tag{3.84}$$

Consider two fixed-terminal-time problems:

$$\tilde{\vartheta}_2^+(t, x) \coloneqq \sup_{\tilde{\eta} \in \tilde{H}(t)} \inf_{\alpha \in \mathcal{A}(t), \mu \in \mathcal{M}(t)} \int_t^T L(\mathrm{x}(s), \alpha(s), \tilde{\eta}[\alpha, \mu](s))\mu(s)ds + g(\mathrm{x}(T)), \tag{3.85}$$

$$\text{subject to } c(\mathrm{x}(s)) \le 0, s \in [t, T], \tag{3.86}$$

where x solves (3.22) for $(\alpha, \tilde{\eta}[\alpha, \mu], \mu)$, and

$$\tilde{\vartheta}_2^-(t, x) \coloneqq \inf_{\zeta \in Z(t), \nu_B \in \mathrm{N}_B(t)} \sup_{\delta \in \mathcal{D}(t)} \int_t^T L(\mathrm{x}(s), \zeta[\delta](s), \delta(s))\nu_B[\delta](s)ds + g(\mathrm{x}(T)), \tag{3.87}$$

$$\text{subject to } c(\mathrm{x}(s)) \le 0, s \in [t, T], \tag{3.88}$$

where x solves (3.22) for $(\zeta[\delta], \delta, \nu_B[\delta])$.

Recall $\tilde{J}$ in (3.30) as below:

$$\tilde{J}(t, x, z, \alpha, \delta, \mu) \coloneqq \max\{\max_{s \in [t,T]} c(\mathrm{x}(s)), g(\mathrm{x}(T)) - \mathrm{z}(T)\}. \tag{3.89}$$

Following similar steps of the proof of Lemma 3, Lemma 4 can be proved.

Lemma 4. Consider $V_2^{\pm}$ in (3.71) and (3.72). For all $(t,x,z) \in [0,T] \times \mathbb{R}^n \times \mathbb{R}$,

$$V_2^+(t,x,z) = \sup_{\tilde{\eta} \in \tilde{H}(t)} \inf_{\substack{\alpha \in \mathcal{A}(t), \\ \mu \in \mathcal{M}(t)}} \tilde{J}(t,x,z,\alpha,\tilde{\eta}[\alpha,\mu],\mu), \tag{3.90}$$

$$V_2^-(t,x,z) = \inf_{\substack{\zeta \in Z(t), \\ \nu_B \in \mathrm{N}_B(t)}} \sup_{\delta \in \mathcal{D}(t)} \tilde{J}(t,x,z,\zeta[\delta],\delta,\nu_B[\delta]). \tag{3.91}$$

By combining the HJ formulation for the fixed-terminal-time problems [4] and Lemma 4, the HJ PDE for the time-invariant SCRAP is derived in Theorem 5. The proof for Theorem 5 is analogous to the proof for Theorem 2.

Theorem 5. (HJ PDE for SCRAP in time-invariant case) Consider SCRAP in the time-invariant case. For all $(t,x,z) \in [0,T] \times \mathbb{R}^n \times \mathbb{R}$, $V_2^{\pm}$ in (3.71) and (3.72) are the unique viscosity solutions to the HJ PDEs:

$$\max\Big\{c(x) - V_2^{\pm}(t,x,z), V_{2,t}^{\pm} - \max\left\{0, \bar{H}^{\pm}(x,z,D_x V_2^{\pm}, D_z V_2^{\pm})\right\}\Big\} = 0 \tag{3.92}$$

in $(0,T) \times \mathbb{R}^n \times \mathbb{R}$, where $\bar{H}^+$ and $\bar{H}^-$ are defined in (3.19) and (3.20), respectively, without the time dependency, and

$$V_2^{\pm}(T,x,z) = \max\{c(x), g(x) - z\} \tag{3.93}$$

on $\{t = T\} \times \mathbb{R}^n \times \mathbb{R}$.

In comparison between the HJ PDEs for SCRAP and its time-invariant version, $\min\{g - z - V_2^{\pm}, V_{2,t}^{\pm} - \bar{H}^{\pm}\}$ in (3.80) becomes $V_{2,t}^{\pm} - \max\{0, \bar{H}^{\pm}\}$ in (3.92). Note that the difference between SCCIP and SCRAP leads to the difference in HJ PDEs for the time-invariant problems: (3.35) has the term $\frac{\partial V_1^{\pm}}{\partial t} - \min\{0, \bar{H}^{\pm}\}$, but (3.92) has the term $V_{2,t}^{\pm} - \max\{0, \bar{H}^{\pm}\}$.

HJ PDE for SCRAP (optimal control setting)

In this chapter, we solve SCRAP in the optimal control setting: for given initial time and state (t,x),

$$\vartheta_2(t,x) \coloneqq \inf_{\alpha \in \mathcal{A}(t)} \min_{\tau \in [t,T]} \int_t^{\tau} L(s, \mathrm{x}(s), \alpha(s)) ds + g(\tau, \mathrm{x}(\tau)), \tag{3.94}$$

$$\text{subject to} \quad c(s, \mathrm{x}(s)) \leq 0, \quad s \in [t,\tau], \tag{3.95}$$

where x solves (3.39). By removing the contribution of player D in Theorem 4 and 5, we solve ϑ_2 using the HJ PDEs in the following remark.

Remark 4. (HJ PDE for SCRAP (optimal control setting)) Let V_2 be the unique viscosity solution to the HJ PDE:

$$\max\Big\{c(t,x) - V_2(t,x,z), \min\left\{g(t,x) - z - V_2(t,x,z), V_{2,t} - \bar{H}(t,x,z,D_xV_2,D_zV_2)\right\}\Big\} = 0 \tag{3.96}$$

in $(0,T)\times\mathbb{R}^n\times\mathbb{R}$, where $\bar{H}$ is defined in (3.41), and

$$V_2(T,x,z) = \max\{c(T,x), g(T,x) - z\} \tag{3.97}$$

on $\{t=T\}\times\mathbb{R}^n\times\mathbb{R}$. Then,

$$\vartheta_2(t,x) = \min z \text{ subject to } V_2(t,x,z) \le 0, \tag{3.98}$$

where ϑ_2 is (3.94) subject to (3.95).

If SCRAP is time-invariant, V_2 is the unique viscosity solution to the HJ PDE:

$$\max\Big\{c(x) - V_2(t,x,z), V_{2,t} - \max\left\{0, \bar{H}(x,z,D_xV_2,D_zV_2)\right\}\Big\} = 0 \tag{3.99}$$

in $(0,T)\times\mathbb{R}^n\times\mathbb{R}$, where $\bar{H}$ is defined in (3.41) without the time dependency, and

$$V_2(T,x,z) = \max\{c(x), g(x) - z\} \tag{3.100}$$

on $\{t=T\}\times\mathbb{R}^n\times\mathbb{R}$.

3.2.3 Optimal Policy and Strategy

The optimal control signal or strategy for SCRAP specified by the HJ PDEs in Chapter 3.2.2. Lemmas 4 imply the following remark.

Remark 5 (Find $\vartheta_2^{\pm}$ from $V_2^{\pm}$)**.** For initial time $t=0$ and state $x\in\mathbb{R}^n$,

$$(\mathrm{x}_*(0), \mathrm{z}_*(0)) = (x, \vartheta_2^{\pm}(0,x)), \tag{3.101}$$

where $(\mathrm{x}_*, \mathrm{z}_*)$ is an optimal trajectory for $V_2^{\pm}$.

With the initial augmented state $(\mathrm{x}_*(0), \mathrm{z}_*(0))$, the optimal control and strategy can be found at $(t, \mathrm{x}_*(t), \mathrm{z}_*(t))$, and the optimal state trajectory is also updated by solving the ODE (3.76) or (3.78).

Define $\tilde{H}_2^{\pm}: A\times D\to\mathbb{R}$ for a fixed $(t,x,z)\in(0,T)\times\mathbb{R}^n\times\mathbb{R}$

$$\tilde{H}_2^{\pm}(a,d) \coloneqq -D_xV_2^{\pm}(t,x,z)\cdot f(t,x,a,d) + D_zV_2^{\pm}(t,x,z)L(t,x,a,d), \tag{3.102}$$

thus

$$\bar{H}^{+}(t,x,z,D_xV_2^{+},D_zV_2^{+}) = \max_{a\in A}\min_{d\in D}\tilde{H}_2^{+}(a,d), \tag{3.103}$$

$$\bar{H}^-(t,x,z,D_xV_2^-,D_zV_2^-) = \min_{d\in D}\max_{a\in A}\tilde{H}_2^-(a,d), \tag{3.104}$$

where $\bar{H}^+$ and $\bar{H}^-$ are defined in (3.19) and (3.20), respectively. In this chapter, we omit (t,x,z) to simplify notation. Using the notation with $\tilde{H}_2^{\pm}$ (3.102), the HJ PDE (3.80) for V_2^+ is equal to

$$\max\{c-V_2^+, \min\{g-z-V_2^+, \frac{\partial V_2}{\partial t}^+ - \max_{a\in A}\min_{d\in D}\tilde{H}_2^+(a,d)\}\} = 0. \tag{3.105}$$

The HJ PDE implies that optimal control signal or strategy is determined by the gradient information at the current time and state (t,x,z). In V_2^+, $\alpha_*(t) = a_*$, $\delta_*[\alpha_*](t) = d_*$ where a_* and d_* are solutions to

$$\max\{c-V_2^+, \min\{g-z-V_2^+, \frac{\partial V_2}{\partial t}^+ - \tilde{H}_2^+(a,d)\}\} = 0. \tag{3.106}$$

at (t,x,z). In other words, it is sufficient to specify optimal controls for player A and D in $(0,T)\times\mathbb{R}^n\times\mathbb{R}$ to generate the optimal control signal or strategy. The maxmini solution (a_*,b_*) for the Hamiltonian $\bar{H}^+$ ($\max_{a\in A}\min_{d\in D}\tilde{H}_2^+(a,d)$) is certainly optimal, but there are more solutions.

Remark 6 (Optimal controls for V_2^+)**.** Fix $(t,x,z)\in(0,T)\times\mathbb{R}^n\times\mathbb{R}$.

Optimal controls for ϑ_2^+ (V_2^+) are the following:

1. Case 1: $c - V_2^+ \geq V_{2,t}^+ - \bar{H}^+$

$$a_* \in \{a\in A \mid V_{2,t}^+ - \min_{b\in B}\tilde{H}_2^+(a,b) \leq 0\}, \tag{3.107}$$

$$b_* \in B; \tag{3.108}$$

2. Case 2: $g - z - V_2^+ \geq V_{2,t}^+ - \bar{H}^+ \geq c - V_2^+$

$$a_* \in \arg\max_{a\in A}\min_{b\in B}\tilde{H}_2^+(a,b), \tag{3.109}$$

$$b_* \in \arg\min_{b\in B}\tilde{H}_2^+(a_*,b). \tag{3.110}$$

3. Case 3: $V_{2,t}^+ - \bar{H}^+ \geq \max\{c-V_2^+, g-z-V_2^+\}$

$$a_* \in A, \tag{3.111}$$

$$b_* \in \{b\in B \mid V_{2,t}^+ - \tilde{H}_2^+(a_*,b) \geq 0\}. \tag{3.112}$$

For ϑ_2^- and V_2^-, we can apply similar arguments to find the corresponding optimal controls and strategies.

Algorithm 2 Computing the solution $V_2^{\pm}$ or V_2 to the HJ PDEs for SCRAP in the zero-sum game and optimal control settings. This algorithm deals with the four HJ PDEs: (3.80), (3.92), (3.96), and (3.99).

Input: the temporal discretization: $\{t_0 = 0, ..., t_K = T\}$, the spatial discretization: $\{(x_0, z_0), ..., (x_N, z_N)\}$
Output: $V_2^{\pm}$ (or V_2)
$V_2^{\pm}$(or V_2)$(T, x_i, z_i) \leftarrow \max\{c(T, x_i), g(T, x_i) - z_i\}, \forall i$
for $k \in \{K-1, ..., 0\}$ **do**
    case solving the HJ PDEs (3.80) or (3.96)
        $V_2^{\pm}$(or V_2)$(t_k, x_i, z_i) \leftarrow V_2^{\pm}$(or V_2)$(t_{k+1}, x_i, z_i) - \Delta_k \hat{\hat{H}}^{\pm}$(or $\hat{\hat{H}}$)$(\phi_x^+, \phi_x^-, \phi_z^+, \phi_z^-), \forall i$
        $V_2^{\pm}$(or V_2)$(t_k, x_i, z_i) \leftarrow \max\{c(t_k, x_i), \min\{g(t_k, x_i) - z_i, V_2^{\pm}$(or V_2)$(t_k, x_i, z_i)\}\}, \forall i$
    case solving the HJ PDEs (3.92) or (3.99)
        $V_2^{\pm}$(or V_2)$(t_k, x_i, z_i) \leftarrow V_2^{\pm}$(or V_2)(t_{k+1}, x_i, z_i)
        $-\Delta_k \max\{0, \hat{\hat{H}}^{\pm}$(or $\hat{\hat{H}}$)$(\phi_x^+, \phi_x^-, \phi_z^+, \phi_z^-)\}, \forall i$
        $V_2^{\pm}$(or V_2)$(t_k, x_i, z_i) \leftarrow \max\{c(t_k, x_i), V_2^{\pm}$(or V_1)$(t_k, x_i, z_i)\}, \forall i$

3.2.4 Numerical Computation

We can find a numerical algorithm by modifying Algorithm 1 in Chapter 3.1.4. Please see Chapter 3.1.4 to find details.

3.2.5 Example

The HJ PDE in Theorem 4 can be numerically computed by Algorithm 2 This is programmed in the `level set method toolbox` [78] and the `helperOC toolbox` [30] in `Matlab`. The simulation is carried out on a desktop computer with a Core i7-5820K CPU and 128 GB RAM.

For both the optimal control problem and the game, we demonstrate the same example: vehicle lane-changing while avoiding obstacles as illustrated in Figure 3.5. In this problem, the vehicle is two degree-of-motion (DOM) in a two dimensional space and starts at the first lane. Two trucks (obstacles) are moving along the second lane to the right. Our vehicle wants to make a right turn at the end of this road, thus needs to change its lane to the third without collision with the two trucks. At the same time, our vehicle does not want to bother the traffic on the first lane. If our vehicle cannot move away from the first lane in a short time, it causes a traffic jam in the first lane. Thus, if the lane changing takes a little time, our vehicle will stop reaching to the goal and keep the lane.

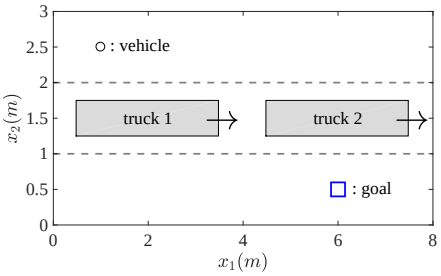
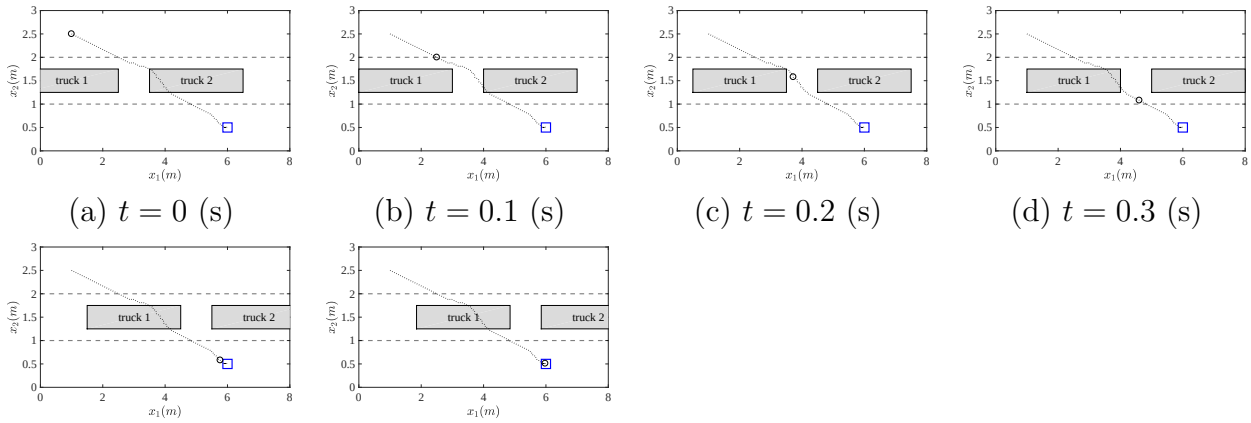


Figure 3.5: Illustration of the lane-changing problem.



(a) $t = 0$ (s) (b) $t = 0.1$ (s) (c) $t = 0.2$ (s) (d) $t = 0.3$ (s)

(e) $t = 0.4$ (s) (f) $t = 0.47$ (s)

Figure 3.6: Optimal control problem with 5 m/s obstacle velocity ($v_o = 5$). ○ represents the vehicle at each time, □ represents the goal, and the dotted line is the trajectory of the vehicle. The terminal time is determined 0.47 (s).

Optimal control problem with state constraint

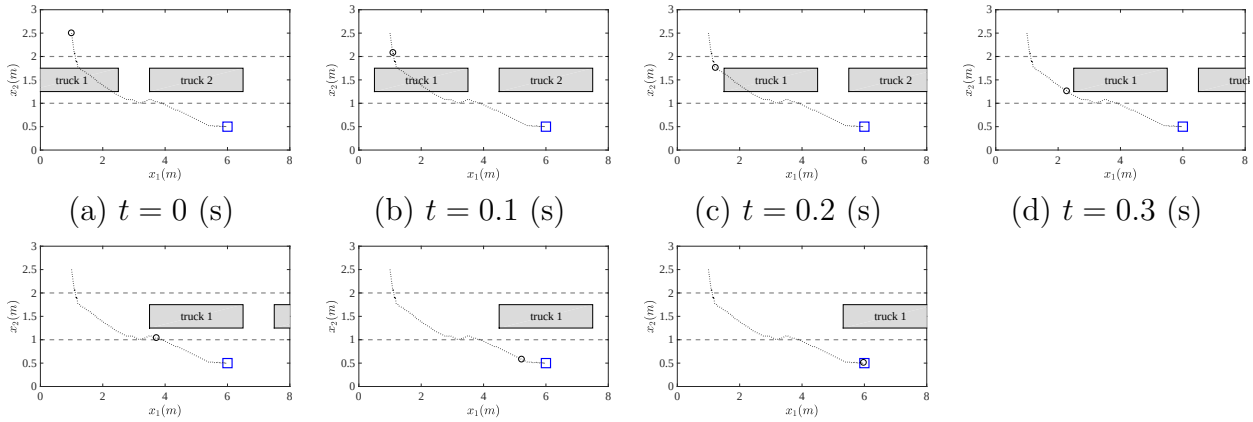
We design the state of the system $x = (x_1; x_2) \in \mathbb{R}^2$ where x_1 and x_2 are horizontal and vertical position of the vehicle. The state trajectory $\mathrm{x}(\cdot)$ follows the dynamics: for $s \in [t, T]$,

$$\dot{\mathrm{x}}(s) := \begin{pmatrix} \dot{\mathrm{x}}_1(s) \\ \dot{\mathrm{x}}_2(s) \end{pmatrix} = \alpha(s) := \begin{pmatrix} \alpha_1(s) \\ \alpha_2(s) \end{pmatrix}, \quad s \in [t, T], \tag{3.113}$$

and $\mathrm{x}(t) = x$ is given, where $\alpha \in \mathcal{A}(t) \subset \mathbb{R}^2$. Also, the two trucks are horizontally moving with the constant velocity v_o.

We design the following optimal control problem:

$$\inf_{\alpha \in \mathcal{A}(0)} \min_{\tau \in [0, 0.7]} \int_0^\tau 1 ds + 2 \|\mathrm{x}(\tau) - x_g\|_2, \tag{3.114}$$



(a) $t = 0$ (s) (b) $t = 0.1$ (s) (c) $t = 0.2$ (s) (d) $t = 0.3$ (s)

(e) $t = 0.4$ (s) (f) $t = 0.5$ (s) (g) $t = 0.582$ (s)

Figure 3.7: Optimal control problem with 10 m/s obstacle velocity ($v_o = 10$). $\circ$ represents the vehicle at each time, $\square$ represents the goal, and the dotted line is the trajectory of the vehicle. The terminal time is determined 0.582 (s).

$$\text{subject to } c(s, \mathrm{x}(s)) \leq 0, \quad s \in [0, \tau], \tag{3.115}$$

where x solves (3.113) for α, $\mathrm{x}(0) = x = (1; 2.5) \in \mathbb{R}^2$, and for $(t, x) \in [0, T] \times \mathbb{R}^n$

$$\begin{aligned} c(t,x) &= \max\{c_0(x), c_1(t,x), c_2(t,x)\}, \\ c_0(x) &= \max(-x_1, |x_2 - 1.5| - 1.5), \\ c_1(t,x) &= -\max(|x_1 - v_o t - 1| - 1.5, |x_2 - 1.5| - 0.25), \\ c_2(t,x) &= -\max(|x_1 - v_o t - 5| - 1.5, |x_2 - 1.5| - 0.25), \end{aligned} \tag{3.116}$$

c_0 constraints our vehicle on the road, c_1 and c_2 are for the safety with respect to the two trucks. The size of the trucks is 3 (m) by 0.5 (m), and the velocity is v_0. This size can be easily changed for realistic sizes. Later, we analyze the behavior of the vehicle for two cases: v_0 is 5 and 10 (m/s). In this problem, the stage cost $L(t, x, a) = 1$, the terminal cost $g(t, x) = 2\|x - x_g\|$, the time horizon $T = 0.7$, the goal position $x_g = (6; 0.5) \in \mathbb{R}^2$, and $A = [1, 15] \times [-5, 5] \subset \mathbb{R}^2$.

The Hamiltonian in (3.41) is

$$\bar{H}(t, x, z, p, q) = \begin{cases} -p_1 + 5|p_2| + q, & p_1 \geq 0, \\ -15p_1 + 5|p_2| + q, & p_1 < 0, \end{cases} \tag{3.117}$$

where $p = (p_1; p_2)$.

For numerical computations of the HJ PDE in Theorem 4, we discretize each state coordinate by 81 grid points and time by $\Delta t = 0.002$ satisfying the CFL condition [85].

Using the toolbox [78] and [30], we approximate V accordingly. Then, we numerically find $\vartheta(0,(1;2.5))$ taking advantage of Theorem 3.

By Chapter 3.2.3, the optimal state trajectory and control signal can be chosen by the gradient of V.

$$\dot{\mathrm{x}}_*(s) = \alpha_*(s), \quad \dot{\mathrm{z}}_*(s) = -1, \tag{3.118}$$

and $\mathrm{x}_*(0) = (1;2.5)$, $\mathrm{z}_*(0) = \vartheta(0,(1;2.5))$, where we denote $\alpha_*(s) = (\alpha_{1*}(s);\alpha_{2*}(s))$, and

$$\alpha_{1*}(s) = \begin{cases} 1, & \frac{\partial V_2}{\partial t}(s,\mathrm{x}_*(s),\mathrm{z}_*(s)) \geq 0, \\ 15, & \frac{\partial V_2}{\partial x_1}(s,\mathrm{x}_*(s),\mathrm{z}_*(s)) < 0, \end{cases} \tag{3.119}$$

$$\alpha_{2*}(s) = \begin{cases} -5, & \frac{\partial V_2}{\partial x_2}(s,\mathrm{x}_*(s),\mathrm{z}_*(s)) \geq 0, \\ 5, & \frac{\partial V_2}{\partial x_1}(s,\mathrm{x}_*(s),\mathrm{z}_*(s)) < 0. \end{cases} \tag{3.120}$$

Figure 3.6 and 3.7 show the results of the optimal control problem for $v_o = 5$ and $v_o = 10$, respectively. For $v_o = 5$, the vehicle successfully changes the lane through the gap between the two trucks and reaches the goal position in 0.47 s. For $v_o = 10$, the vehicle also successfully changes the lane but shows different behavior. It first waits for truck 1 moving forward and making some space behind truck 1. Then, the vehicle moves behind truck 1 and heads to the goal. The goal is reached in 0.582 s. It is observed that the vehicle cannot safely pass the gap of the two trucks when their speed is fast.

In the simulation, the grid size in the time and the state is 351 and 81^3, respectively. The computation times are 6.0 and 6.3 minutes for $v_o = 5$ and 10, respectively. Since the state-time space is discretized, the computational complexity is exponential in the dimension of the state.

Zero-sum game problem with state constraint

We modify the example in Chapter 3.2.5: we assume that truck 1 does not want our vehicle to achieve its lane change. Player A is our vehicle, and player B is truck 1. We additionally add the horizontal position of the center of truck 1, x_3, to the state: $x \coloneqq (x_1;x_2;x_3)$.

For this problem, we design the following zero-sum game problem:

$$\sup_{\beta\in\Delta(0)} \inf_{\alpha\in\mathcal{A}(0)} \min_{\tau\in[0,0.7]} \int_0^\tau 1ds + 2\left\| \begin{pmatrix} \mathrm{x}_1(\tau) \\ \mathrm{x}_2(\tau) \end{pmatrix} - x_g \right\|_2, \tag{3.121}$$

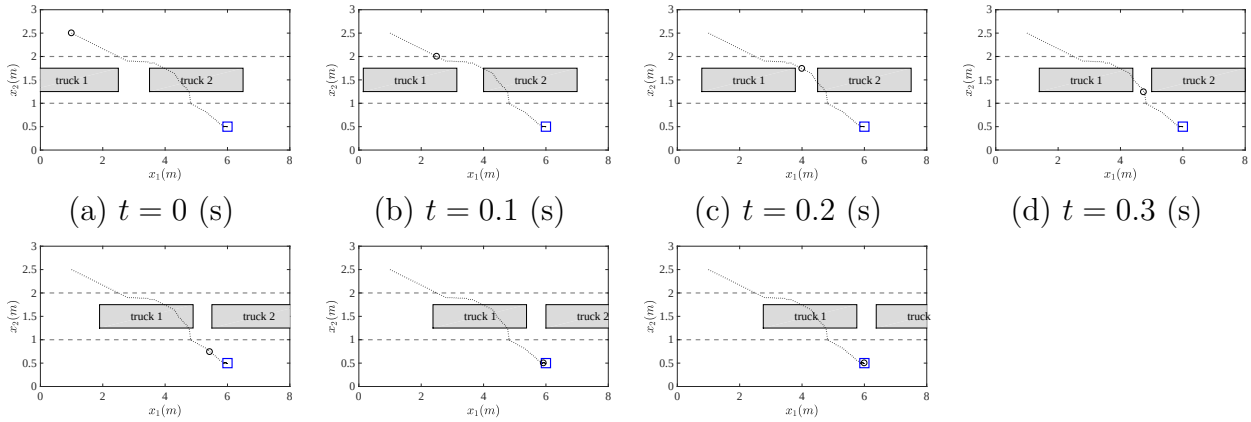
$$\text{subject to } c(s,\mathrm{x}(s)) \leq 0, \quad s \in [0,\tau], \tag{3.122}$$

where

$$\dot{\mathrm{x}}_1(s) = \alpha_1(s), \tag{3.123}$$

$$\dot{\mathrm{x}}_2(s) = \alpha_2(s), \tag{3.124}$$

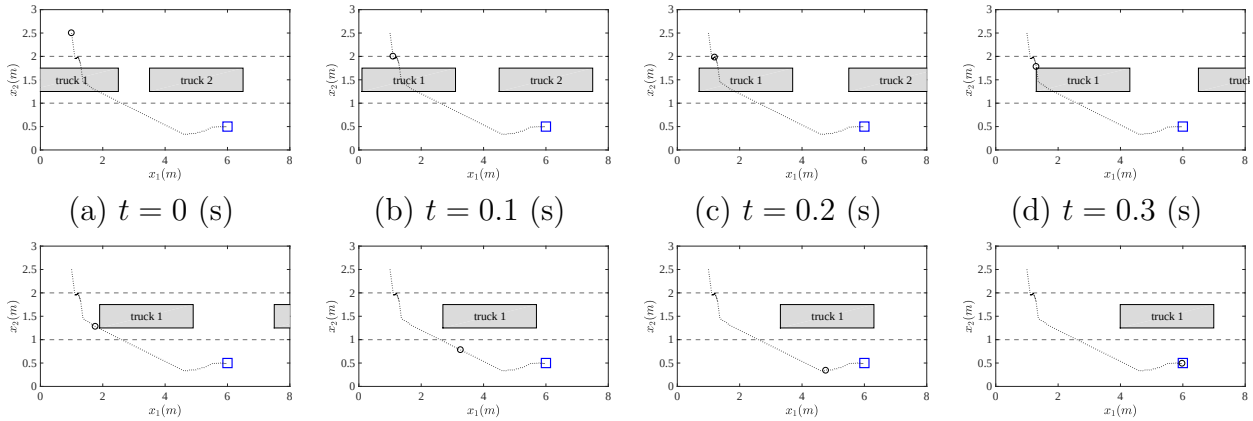
$$\dot{\mathrm{x}}_3(s) = v_o + \delta[\alpha](s), \tag{3.125}$$



(a) $t = 0$ (s) (b) $t = 0.1$ (s) (c) $t = 0.2$ (s) (d) $t = 0.3$ (s)

(e) $t = 0.4$ (s) (f) $t = 0.5$ (s) (g) $t = 0.576$ (s)

Figure 3.8: Zero-sum two-player game problem with 5 m/s obstacle velocity ($v_o = 5$). $\circ$ represents the vehicle at each time, $\square$ represents the goal, and the dotted line is the trajectory of the vehicle. The terminal time is determined 0.576 (s).



(a) $t = 0$ (s) (b) $t = 0.1$ (s) (c) $t = 0.2$ (s) (d) $t = 0.3$ (s)

(e) $t = 0.4$ (s) (f) $t = 0.5$ (s) (g) $t = 0.6$ (s) (h) $t = 0.7$ (s)

Figure 3.9: Zero-sum two-player game problem with 10 m/s obstacle velocity ($v_o = 10$). $\circ$ represents the vehicle at each time, $\square$ represents the goal, and the dotted line is the trajectory of the vehicle. The terminal time is determined 0.7 (s).

$\mathrm{x}(t) = x = (1; 2.5; 1)$, $\alpha(s) = (\alpha_1(s); \alpha_2(s))$, and for $(t, x) \in [0, T] \times \mathbb{R}^n$

$$
\begin{aligned}
c(t,x) &= \max\{c_0(x), c_1(t,x), c_2(t,x)\}, \\
c_0(x) &= \max(-x_1, |x_2 - 1.5| - 1.5), \\
c_1(x) &= -\max(|x_1 - x_3| - 1.5, |x_2 - 1.5| - 0.25), \\
c_2(t,x) &= -\max(|x_1 - v_o t - 5| - 1.5, |x_2 - 1.5| - 0.25),
\end{aligned}
\tag{3.126}
$$

c_0 constraints our vehicle on the road, c_1 and c_2 constraints the safety with respect to the two trucks. The two trucks are moving on the second lane with a constant velocity v_0 along x_1-axis. In this problem, $L(t,x,a,b) = 1, g(t,x) = 2\|x - x_g\|, T = 0.7, x_g = (6; 0.5)$, $A \in [1, 15] \times [-5, 5]$, $B \in [-5, 5]$.

In this example, the upper and lower Hamiltonians in (3.19) and (3.20) are the same: $\bar{H}^{\pm} = \bar{H}^{+} = \bar{H}^{-}$.

$$H^{\pm}(t,x,z,p,q) = \begin{cases} -p_1 + 5|p_2| - v_o p_3 + 5|p_3| + q, & p_1 \geq 0, \\ -15p_1 + 5|p_2| - v_o p_3 + 5|p_3| + q, & p_1 < 0, \end{cases} \tag{3.127}$$

where $p = (p_1; p_2; p_3)$.

For numerical computations of the HJ PDE in Theorem 4, we discretize each state coordinate by 61 grid points and time by $\Delta t = 0.002$ satisfying the CFL condition [85]. Using the toolbox [78] and [30], we approximate V accordingly. Then, we numerically find $\vartheta^{\pm}(0, (1; 2.5; 1))$ taking advantage of Theorem 3.

By Chapter 3.2.3, the strategies for each player can be chosen by the gradient of $V_2^{\pm}$. For convenience, we present the strategies for the upper value function V^{+}.

$$\dot{\mathrm{x}}_*(s) = \left(\alpha_{1*}(s);\ \alpha_{2*}(s);\ v_o + \beta_*(s)\right), \quad \dot{\mathrm{z}}_*(s) = -1, \tag{3.128}$$

and $\mathrm{x}_*(0) = (1; 2.5; 1)$, $\mathrm{z}_*(0) = \vartheta(0, (1; 2.5; 1))$, where we denote $\alpha_*(s) = (\alpha_{1*}(s); \alpha_{2*}(s))$, and

$$\alpha_{1*}(s) = \begin{cases} 1, & \frac{\partial V_2^{\pm}}{\partial x_1}(s, \mathrm{x}_*(s), \mathrm{z}_*(s)) \geq 0, \\ 15, & \frac{\partial V_2^{\pm}}{\partial x_1}(s, \mathrm{x}_*(s), \mathrm{z}_*(s)) < 0, \end{cases} \tag{3.129}$$

$$\alpha_{2*}(s) = \begin{cases} -5, & \frac{\partial V_2^{\pm}}{\partial x_2}(s, \mathrm{x}_*(s), \mathrm{z}_*(s)) \geq 0, \\ 5, & \frac{\partial V_2^{\pm}}{\partial x_2}(s, \mathrm{x}_*(s), \mathrm{z}_*(s)) < 0. \end{cases} \tag{3.130}$$

$$\delta_*[\alpha_*](s) = \begin{cases} 5, & \frac{\partial V_2^{\pm}}{\partial x_3}(s, \mathrm{x}_*(s), \mathrm{z}_*(s)) \geq 0, \\ -5, & \frac{\partial V_2^{\pm}}{\partial x_3}(s, \mathrm{x}_*(s), \mathrm{z}_*(s)) < 0. \end{cases} \tag{3.131}$$

Figure 3.8 and 3.9 show the results of the two-player zero-sum game for $v_o = 5$ and $v_o = 10$, respectively. For $v_o = 5$, the vehicle changes the lane through the gap between the two trucks and reaches its the goal position in 0.576 s. For $v_o = 10$, the vehicle waits for truck 1 moving forward and then moves behind truck 1 to the goal. The goal is reached in 0.7 s.

We compare the optimal control problems and the games for each v_0. For the both case $v_o = 5$ and $v_o = 10$, the games have a higher terminal time for the optimal control problems. This is reasonable as truck 1 bothers the goal of the vehicle in the game. In comparison to the trajectories of the vehicle, for $v_0 = 5$, the vehicle moves closer to truck 2 in the game rather than in the optimal control problem. For $v_0 = 10$, in the game, truck 1 slows down its speed to bother the vehicle at the beginning. Particularly at $t = 0.3$, the vehicle already passes behind truck 1 as shown in Figure 3.7 (d), but the vehicle still waits for truck 1 moving forward as shown in Figure 3.9 (d).

In the simulation, the grid size in the time and the state is 351 and 61^4, respectively. The computation times are 183.1 and 182.7 minutes for $v_o = 5$ and 10, respectively. This is about 29 times longer than the computation times for the optimal control problem (6.2 minutes). This ratio (29) is similar to the ratio of the grid size: $61^4/81^3 = 26.1$. Here, it is observed that the computation time is exponentially increasing in the dimension of the state. For the game problem, nearly 200 GB memory is used including RAM and swap memory. This huge memory requirement leads to the intractability of the computation for the system whose dimension is higher than four.

3.3 Computational Complexity

For both SCCIP and SCRAP, the HJ analysis deals with general nonlinear systems and non-convex state constraints. The solution of HJ PDEs is numerically computed by the level-set or fast marching methods [85], which require gridding on the state space. Because of the gridding, HJ analysis suffers from computational complexity that grows exponentially in the system dimension. In order to alleviate this issue there have been various approaches, including optimization with approximation techniques [61, 86, 88, 107], control-barrier-function-based methods [6, 33, 105, 99, 60], geometry-based formulation [87, 77, 25, 91, 15], temporal logic [71, 43], Hopf-Lax theory [64, 56, 42, 12], and learning-based approaches [32, 101, 89, 45].

In the following two chapters, this dissertation presents two efficient methods. Chapter 4 presents a new Hopf-Lax theory for state-constrained problems. Chapter 5 presents a reinforcement learning framework for reachability.

Chapter 4

Hopf-Lax Theory

Hopf-Lax theory is inverse HJ analysis: given an HJ PDE, Hopf-Lax theory provides a corresponding optimization problem called the Hopf-Lax formula [64, 56, 42, 12], as shown in Figure 4.1. This formulation is valid where the Hamiltonian of HJ PDEs is convex. This assumption does not generally hold for zero-sum games, so we deal with optimal control settings in this chapter.

Similar to solving optimal control problems, Hopf-Lax formulae can be solved by numerical optimization methods, such as convex optimization or nonlinear programming [19, 84, 51], and do not require gridding the state space [38, 106, 34, 68]. The computational complexity of Hopf-Lax formulae depends on numerical methods for convex programming, which is typically polynomial in the state dimension [38]. Although solving Hopf-Lax formulae is more efficient than solving HJ PDEs, HJ analysis provides more information than the Hopf-Lax formula. HJ analysis provides a closed-loop control, but the Hopf-Lax formula provides an open-loop control. Thus, the Hopf-Lax formula must be computed in real-time if the precomputed open-loop control drives systems to unexpected states. For real-time computation, Hopf-Lax theory can incorporate various computationally efficient convex-programming methods [19, 20] and approximation techniques, including receding horizon [39].

We can solve optimal control problems in two steps using Hopf-Lax theory. First, we derive corresponding HJ PDEs and then Hopf-Lax formulae and compute a numerical solution for the Hopf-Lax formulae. There are two advantages of solving Hopf-Lax formulae by numerical optimization methods instead of the optimal control problem. First, Hopf-Lax formulae

Figure 4.1: Hopf-Lax theory is inverse HJ analysis: given an HJ PDE, Hopf-Lax theory provides a corresponding optimization problem.

have broader convexity conditions than optimal control problems for some cases [12, 68]. Convexity is critical to guarantee optimality in the use of numerical optimization methods. Second, Hopf-Lax formulae have lower-dimensional variables than optimal control problems for some cases [12, 38, 34].

The above advantages allow Hopf-Lax theory to become an active research field to accommodate various types of optimal control problems. Hopf-Lax theory has been first developed for HJ PDEs, whose Hamiltonian has only costate-dependency, relevant to state-unconstrained optimal control problems whose dynamics only have control-input-dependency [12, 56]. Later, this theory was extended for HJ PDEs, whose Hamiltonian additionally has time-and-state-dependency, which is relevant to the time-varying nonlinear system's optimal control problem [72, 38, 106, 34]. Furthermore, Hopf-Lax theory has been extended to HJ PDEs, which are relevant to state-constrained problems [36, 68, 66]. [36] deals with HJ PDEs whose Hamiltonian has only costate-dependency. This Hopf-Lax formula [36] is a solution to a particular state-constrained problem, where the solution's epigraph is the solution to the HJ PDE corresponding the state-constrained problem. [68, 66] solve state-constrained problems for nonlinear systems by presenting relevant Hopf-Lax formulae. Particularly, [66] presents the Hopf-Lax formula for the SCRAP in a time-invariant case: cost functions, dynamics, and state constraints are time-invariant. Note that the time-varying case refers to problems whose cost functions, dynamics, and state constraints are time-varying.

Several Hopf-Lax formulae relevant to state-unconstrained problems have been rigorously proved as the viscosity solution to the corresponding HJ PDEs [12, 106]. On the other hand, no viscosity theory has been proved for the Hopf-Lax formulae relevant to state-constrained problems [36, 68, 66].

Organization

This chapter presents Hopf-Lax formulae for three state-constrained problems. Chapters 4.1, 4.2, and 4.3 provide Hopf-Lax formulae for state-constrained general sum problems, SCCIP, and SCRAP, respectively.

Notations

This paper uses the subscript of $*$ to denote optimality and superscript of $*$ to denote Legendre-Fenchel transformation. For example, α_* denotes an optimal control signal, and H^* denotes the Legendre-Fenchel transformation of a function H. $H^{**} = (H^*)^*$ denotes the Legendre-Fenchel transformation of the Legendre-Fenchel transformation of a function H. The mathematical definition of the Legendre-Fenchel transformation will be introduced where we define particular functions using the transformation.

4.1 Hopf-Lax Formula for State-Constrained General-Sum Problems

This chapter is based on the work presented in [65], which is joint work with Claire Tomlin.

This chapter presents a Hopf-Lax formula for state-constrained general-sum problems (SCGSP). Our formulation convexifies SCGSP to the control space. Thus, the Hopf-Lax formula is convex under the following conditions for the given SCGSP: (a) the dynamics are affine in the state; (b) the stage and terminal cost, as well as the state constraints, are convex in the state. Several gradient-based methods guarantee optimality for convex problems. The Hopf-Lax formula allows us to find the optimal solution for the classes of problems satisfying the above conditions even if SCGSP is non-convex.

The organization of this chapter is as follows. Chapter 4.1.1 presents SCGSP. Chapter 4.1.2 proposes the Hopf-Lax formula for SCGSP and presents a numerical algorithm to compute an optimal state trajectory and control signal. Chapter 4.1.4 presents convexity conditions for the Hopf-Lax formula. Chapter 4.1.5 provides a numerical example to demonstrate the proposed Hopf-Lax formula.

4.1.1 State-Constrained Optimal Control Problem

Consider a state trajectory solving (2.1) in optimel control settings.

$$\dot{\mathrm{x}}(s) = f(s, \mathrm{x}(s), \alpha(s)),\ s \in [t, T], \quad \mathrm{x}(t) = x, \tag{4.1}$$

where (t, x) are the initial time and state, and $\alpha(s) \in A$ for all $s \in [t, T]$. We would like to solve SCGSP for the dynamic system:

$$\vartheta(t, x) \coloneqq \lim_{\epsilon \to 0} \vartheta^{\epsilon}(t, x), \tag{4.2}$$

where

$$\vartheta^{\epsilon}(t, x) \coloneqq \inf_{\alpha} \int_t^T L(s, \mathrm{x}(s), \alpha(s)) ds + g(\mathrm{x}(T)) \tag{4.3}$$

$$\text{subject to } c(s, \mathrm{x}(s)) \leq \epsilon, \quad s \in [t, T], \tag{4.4}$$

where x solves (2.1). Here, $L : [t, T] \times \mathbb{R}^n \times A \to \mathbb{R}$ is the stage cost, $g : \mathbb{R}^n \to \mathbb{R}$ is the terminal cost, $f : [t, T] \times \mathbb{R}^n \times A \to \mathbb{R}^n$ is the system dynamics, $c : [t, T] \times \mathbb{R}^n \to \mathbb{R}$ is the state constraint, and $\alpha \in \mathcal{A}(t)$ is the control signal where $\mathcal{A}(t)$ is the set of admissible control signals:

$$\mathcal{A}(t) \coloneqq \{\alpha : [t, T] \to A \mid \|\alpha\|_{L^{\infty}(t,T)} < \infty\}, \tag{4.5}$$

and A is a compact subset in $\mathbb{R}^m$. In practice, $c(s, \cdot)$ represents unsafe regions at $s \in [t, T]$ so that $c(s, x) < 0$ for $x \in \mathbb{R}^n$ away from the unsafe regions, $c(s, x) = 0$ for x on the boundary of the unsafe regions, and $c(s, x) > 0$ for x in the unsafe regions.

Under Assumption 1, we first show the existence of $\vartheta(t, x)$ in (4.2).

Lemma 5. Suppose Assumption 1 holds, $\vartheta(t,x) : [0,T] \times \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ exists.

Proof. If $\epsilon_1 > \epsilon_2 > 0$, $\vartheta^{\epsilon_1}(t,x) \leq \vartheta^{\epsilon_2}(t,x)$. Since ϑ^ϵ is monotonically increasing as ϵ is monotonically decreasing and converging to zero, the limit of ϑ^ϵ exists in $\mathbb{R} \cup \{\infty\}$. □

4.1.2 Hopf-Lax formula and Optimal Control

Based on the Hopf-Lax formula [12], Chapter 4.1.2 proposes a convexifying formulation for SCGSP, the Hopf-Lax formula. Chapter 4.1.2 builds up mathematical background to prove the proposed formulation in Chapter 4.1.3.

The Hopf-Lax Formula for the State-Constrained Optimal Control Problem

Consider a control space transformation: for $(s,x,a) \in [t,T] \times \mathbb{R}^n \times A$,

$$b = -f(s,x,a) \in \mathbb{R}^n, \tag{4.6}$$

where f and A are the dynamics and the control constraint of the system, respectively. For (s,x), define

$$B(s,x) := \{-f(s,x,a) \mid a \in A\}. \tag{4.7}$$

$B(s,x)$ is compact for all (s,x) since A is compact and f is Lipschitz as in Assumption 1. Consider a state trajectory solving

$$\begin{aligned} &\dot{\mathrm{x}}(s) = -\beta(s),\ s \in [t,T], \quad \mathrm{x}(t) = x, \\ &\beta(s) \in \mathrm{co}(B(s,\mathrm{x})),\ s \in [t,T], \end{aligned} \tag{4.8}$$

where "co" represents a convex-hull operation.

Based on this control space transformation, Theorem 6 proposes a Lax formula for SCGSP, which is proved in Chapter 4.1.3.

Theorem 6. (The Hopf-Lax formula for SCGSP) Suppose Assumption 1 holds. For initial time and state $(t,x) \in [0,T] \times \mathbb{R}^n$, define

$$\varphi(t,x) \coloneqq \inf_{\beta} \int_t^T H^*(s,\mathrm{x}(s),\beta(s))ds + g(\mathrm{x}(T)), \tag{4.9}$$

$$\text{subject to } c(s,\mathrm{x}(s)) \leq 0, \quad s \in [t,T], \tag{4.10}$$

where x solves (4.8), $H : [0,T] \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$

$$H(t,x,p) \coloneqq \max_{a \in A} -p \cdot f(t,x,a) - L(t,x,a), \tag{4.11}$$

and H^* is the Legendre-Fenchel transformation (convex conjugate) of H with respect to the costate p:

$$H^*(t,x,b) \coloneqq \max_p [p \cdot b - H(t,x,p)]. \tag{4.12}$$

Then,

$$\vartheta(t,x) = \varphi(t,x) \quad \text{in } [t,T] \times \mathbb{R}^n. \tag{4.13}$$

Mathematical Formulation and Optimal Control Analysis

Define

$$L^b(s,x,b) := \min_{a \in A} L(s,x,a) \quad \text{s.t.} \quad f(s,x,a) = -b. \tag{4.14}$$

Lemma 6. Suppose Assumption 1 holds. L^b in (4.14) and H in (4.11) have the following properties.

$$(L^b)^*(s,x,p) = H(s,x,p) \qquad \text{in } [0,T] \times \mathbb{R}^n \times \mathbb{R}^n, \tag{4.15}$$

$$(L^b)^{**}(s,x,b) = H^*(s,x,b) \quad \text{in } [0,T] \times \mathbb{R}^n \times \mathbb{R}^n, \tag{4.16}$$

$$\mathrm{Dom}(H^*(s,x,\cdot)) = \mathrm{co}(B(s,x)). \tag{4.17}$$

$(L^b)^*$ and H^* are the Legendre-Fenchel transformations (convex conjugate) of L^b and H, respectively, with respect to p for each (s,x). $\mathrm{Dom}(H^*(s,x,\cdot))$ represents the domain of $H^*(s,x,\cdot)$, and $\mathrm{co}(B(s,x))$ represents the convex hull of $B(s,x)$.

Proof. See Appendix C.1.

(4.17) in Lemma 6 implies that the domain of $H^*(s,x,\cdot)$ contains the domain of $L^b(s,x,\cdot)$ for each (s,x). Even though $\mathrm{co}(B(s,x))$ is convex in $b \in \mathbb{R}^n$, $\mathrm{Dom}(H^*(s,\cdot,\cdot)) = \{(x,b) \mid b \in \mathrm{co}(B(s,x))\}$ is generally non-convex in $(x,b) \in \mathbb{R}^n \times \mathbb{R}^n$ for each $s \in [t,T]$.

We first state some properties regarding the Legendre-Fenchel transformations in Lemma 2.

Corollary 2 (Decomposition of control and stage cost)**.** Suppose Assumption 1 holds. For all $(s,x) \in [0,T] \times \mathbb{R}^n$ and $b \in \mathrm{co}(B(s,x))$, there exists a finite $b_i \in B(s,x)$, $a_i \in A$, $\gamma_i \in \mathbb{R}$ such that

$$H^*(s,x,b) = \sum_i \gamma_i L^b(s,x,b_i) = \sum_i \gamma_i L(s,x,a_i), \tag{4.18}$$

$$b = \sum_i \gamma_i b_i, \quad b_i = -f(s,x,a_i), \tag{4.19}$$

where $L^b(s,x,b_i) = L(s,x,a_i)$, $\sum_i \gamma_i = 1$, and $\gamma_i \geq 0$. Note that A is the control constraint, and $B(s,x)$ is defined in (4.7).

Consider any feasible control signal (β) and state (x) trajectories solving the dynamics (4.8). Corresponding to β and x, Theorem 7 proposes a corresponding control signal ($\alpha^\epsilon \in \mathcal{A}(t)$) and approximate state trajectory (x^ϵ) solving (2.1) such that $\|\mathrm{x} - \mathrm{x}^\epsilon\|_{L^\infty(t,T)} < \epsilon$, and the difference of the corresponding costs in (4.3) and (4.9) is also bounded by ϵ.

Assume that β is Riemann integrable in (t, T). For some $\delta > 0$, consider a temporal discretization: $\{t_0 = t, ...t_K = T\}$ such that $\Delta t_k := t_{k+1} - t_k < \delta, \forall k = 0, ..., K-1$. We define a control signal $\alpha^\epsilon \in \mathcal{A}(t)$: for $k = \{0, ..., K-1\}$,

$$\alpha^\epsilon(s) = a_i^k, \quad s \in \left[t_k + \sum_{j=1}^{i-1} \gamma_j^k \Delta t_k, t_k + \sum_{j=1}^{i} \gamma_j^k \Delta t_k \right), \tag{4.20}$$

where a_i^k and γ_i^k are i-th control and coefficient in Corollary 2 for $t = t_k$, $x = \mathrm{x}(t_k)$, and $b = \beta(t_k)$. We also define a state trajectory $\mathrm{x}^\epsilon : [t, T] \to \mathbb{R}^n$ solving

$$\dot{\mathrm{x}}^\epsilon(s) = f(s, \mathrm{x}^\epsilon(s), \alpha^\epsilon(s)), \quad s \in (t, T), \quad \mathrm{x}^\epsilon(t) = x. \tag{4.21}$$

Theorem 7 states that α^ϵ in (4.20) and x^ϵ in (4.21) are control signal and approximate state trajectory that satisfy (4.22) and (4.23).

Theorem 7. Suppose Assumption 1 holds. For initial time and state $(t, x) \in [0, T] \times \mathbb{R}^n$, consider any feasible control signal β and state trajectory x solving (4.8). Assume that β is Riemann integrable in $[t, T]$. Then, for any $\epsilon > 0$, there exists $\delta > 0$ such that, for any discretization $\{t_0 = t, ..., t_K = T\}$ where $|\Delta t_k| < \delta$, $k = 0, ..., K-1$: α^ϵ in (4.20) and x^ϵ in (4.21) satisfy

$$\|\mathrm{x} - \mathrm{x}^\epsilon\|_{L^\infty(t,T)} < \epsilon \tag{4.22}$$

and

$$\left| \int_t^T H^*(s, \mathrm{x}(s), \beta(s)) ds + g(\mathrm{x}(T)) - \int_t^T L(s, \mathrm{x}^\epsilon(s), \alpha^\epsilon(s)) ds - g(\mathrm{x}^\epsilon(T)) \right| < \epsilon. \tag{4.23}$$

Proof. See Appendix C.3.

Remark 7 (Optimal control signal)**.** The cost and state trajectory determined by a control signal β in the Hopf-Lax formula is the same as the ones by the control signal α^ϵ (4.20) in SCGSP, as ϵ converges to zero.

4.1.3 Proof of the Hopf-Lax Formula

This chapter proves Theorem 6 using mathematical background in Chapter 4.1.2.

Proof of Theorem 6. (i) Define functions J and $\bar{J}$ with the auxiliary variable $z \in \mathbb{R}$: for initial time $t \in [0,T]$, initial state $x \in \mathbb{R}^n$, auxiliary variable $z \in \mathbb{R}$, control signal $\alpha \in \mathcal{A}(t)$,

$$J(t,x,z,\alpha) \coloneqq \max\left\{ \max_{s\in[t,T]} c(s,\mathrm{x}(s)), \int_t^T L(s,\mathrm{x}(s),\alpha(s))ds + g(\mathrm{x}(T)) - z \right\}, \tag{4.24}$$

where x solves (2.1) for α; for initial time $t \in [0,T]$, initial state $x \in \mathbb{R}^n$, auxiliary variable $z \in \mathbb{R}$, control signal $\beta \in \{\beta \mid \beta(s) \in \mathrm{co}(B(s,\mathrm{x}(s)))\}$,

$$\bar{J}(t,x,z,\beta) \coloneqq \max\left\{ \max_{s\in[t,T]} c(s,\mathrm{x}(s)), \int_t^T H^*(s,\mathrm{x}(s),\beta(s))ds + g(\mathrm{x}(T)) - z \right\}, \tag{4.25}$$

where x solves (4.8) for β.
(ii) $\inf_\alpha J(t,x,z,\alpha) \geq \inf_\beta \bar{J}(t,x,z,\beta)$

For any feasible state (x) and control (α) trajectories solving (2.1), define a control signal (β):

$$\beta(s) = -f(s,\mathrm{x}(s),\alpha(s)) \in B(s,\mathrm{x}(s)), \quad s \in [t,T].$$

Then, x and β solve $\dot{\mathrm{x}}(s) = -\beta(s)$ for $s \in [t,T]$ and $\mathrm{x}(t) = x$. By Lemma 6, $L(s,\mathrm{x}(s),\alpha(s)) = L^b(s,\mathrm{x}(s),\beta(s)) \geq H^*(s,\mathrm{x}(s),\beta(s))$ for all $s \in [t,T]$. This implies that

$$J(t,x,z,\alpha) \geq \bar{J}(t,x,z,\beta) \geq \inf_\beta \bar{J}(t,x,z,\beta).$$

Since the above inequality holds for all any feasible x and α, we conclude $\inf_\alpha J(t,x,z,\alpha) \geq \inf_\beta \bar{J}(t,x,z,\beta)$.
(iii) $\inf_\alpha J(t,x,z,\alpha) \leq \inf_\beta \bar{J}(t,x,z,\beta)$

For any feasible state (x) and control (β) trajectories solving (4.8), by Theorem 7, there exists x^ϵ and α^ϵ solving (2.1) such that (4.21) and (4.20) hold for any $\epsilon > 0$. Then,

$$\begin{aligned} \bar{J}(t,x,z,\beta) &\geq J(t,x,z,\alpha^\epsilon) - \max\{1,L_c\}\epsilon \\ &\geq V(t,x,z) - \max\{1,L_c\}\epsilon, \end{aligned} \tag{4.26}$$

where L_c is the Lipschitz constant for c in Assumption 1. Since (4.26) holds for any x, β solving (4.8) and $\epsilon > 0$, $\inf_\beta \bar{J}(t,x,z,\beta) \geq \inf_\alpha J(t,x,z,\alpha)$.
(iv) By Theorem 3.1 in [4],

$$\vartheta(t,x) = \min z \text{ subject to } \inf_\alpha J(t,x,z,\alpha) \leq 0, \tag{4.27}$$

$$\bar{\vartheta}(t,x) = \min z \text{ subject to } \inf_\beta \bar{J}(t,x,z,\beta) \leq 0. \tag{4.28}$$

By combining this with (ii) and (iii), we conclude $\vartheta \equiv \bar{\vartheta}$. $\square$

Remark 8.

1. The state-constrained optimal control problem (4.2) can be solved by the Hopf-Lax formula in Theorem 6.

2. Corresponding to an optimal control signal β_* for the Hopf-Lax formula, Theorem 7 provides an optimal control signal to SCGSP, as ϵ converges to 0. However, the limit point of α^ϵ might not exist.

Numerical Algorithm

Algorithm 3 presents a numerical algorithm to compute an optimal state trajectory (x) and a control signal (α) for the state-constrained problem (4.2) using the Hopf-Lax formula in Theorem 6.

Algorithm 3 Computing optimal state trajectory (x) and control signal (α) for the state-constrained problem (4.2) using the Hopf-Lax formula

1: **Input:** initial time t, initial state x
2: **Output:** the optimal state (x) and control (α) trajectories
3: Solve $\bar{\vartheta}$ in (4.9) subject to (4.10), and get x_*, β_*
4: On a temporal discretization: $\{t_0 = t, ..., t_K = T\}$, find $(a_i^k, b_i^k, \gamma_i^k)$ solving (4.18) and (4.19) for $x = \mathrm{x}_*(t_k)$ and $s = t_k$
5: Additionally discretize each temporal interval:

$$[t_k, t_{k+1}) = \bigcup_i [t_k + \sum_{j=1}^{i-1} \gamma_j^k \Delta t_k, t_k + \sum_{j=1}^{i} \gamma_j^k \Delta t_k) \tag{4.29}$$

6: Design α_*^ϵ using a_i^k by (4.20) and compute x_*^ϵ by solving the ODE (4.21)

We first numerically compute an optimal state trajectory (x_*) and a control signal (β_*) for the Hopf-Lax formula in Theorem 6. Choice of methods to solve $\bar{\vartheta}$ in line 3 in Algorithm 3 is open to the users, such as gradient-based methods with temporal discretization [90] or Pontryagin minimum principle [21]. Then, Theorem 7 is utilized to get a numerical optimal state (x^ϵ) and control (α^ϵ) for the state-constrained problem (4.2), which corresponds line 4-6 in Algorithm 3. Note that, an additional temporal discretization on line 5 provides a chattering control signal unless $\beta_*(s) \in B(s, \mathrm{x}_*(s))$ for all s.

4.1.4 Convexity Analysis for the Hopf-Lax Formula

We call SCGSP convex if the stage and terminal cost are convex in the state and the control, the control constraint is convex in the control, the dynamics are affine in the state and control, and the state constraint is convex in the state. If SCGSP is convex, most temporal

Table 4.1: Convexity conditions for SCGSP and the Hopf-Lax formula

	the state-constrained optimal control problem	the Hopf-Lax formula
formulation	(4.3) subject to (4.4)	(4.9) subject to (4.10)
costs		
stage cost $L = L^x(s,x) + L^a(s,a)$	$L^x(s,\cdot)$ is convex in x $L^a(s,\cdot)$ is convex in a	$L^x(s,\cdot)$ is convex in x See Lemma 8
terminal cost g	convex in x	convex in x
constraints		
control constraint A	convex	no condition See Lemma 9
dynamics function $f(s,x,a)$	$f = M(s)x + N(s)a + C(s)$	$f = M(s)x + \varphi(s,a)$ See Lemma 8 and 9
state constraint $c(s,\cdot)$	convex in x	convex in x

discretization methods, such as Euler forward and backward discretization, higher-ordered Runge-Kutta, provide a convex problem. Thus, gradient-based methods guarantees global optimality. Pontryagin minimum principle is also sufficient for the global optimality for the convex optimal control problem [55].

This chapter investigates conditions under which the Hopf-Lax formula is convex but the given problem is non-convex.

Convexity Analysis for the Hopf-Lax Formula

For convexity analysis, we deal with the systems whose stage cost is in the following form:

$$L(s,x,a) = L^x(s,x) + L^a(s,a), \tag{4.30}$$

where L is the stage cost of SCGSP (4.2). The convexity conditions for the state-constrained problem ϑ are written in the third column of Table 4.1. The fourth column in Table 4.1 presents a sufficient convexity condition for the Hopf-Lax formula $\bar{\vartheta}$ ((4.9) subject to (4.10)).

We present Lemma 7 that will be used to derive a convexity condition of the stage cost for the Hopf-Lax formula ((4.9) subject to (4.10)) in Lemma 8.

Lemma 7. Suppose (4.30) holds. Then,

$$H^*(s,x,b) = L^x(s,x) + (H^a)^*(s,x,b), \tag{4.31}$$

where H^* is defined in (4.12),

$$H^a(s,x,p) \coloneqq \max_{a\in A} -p\cdot f(s,x,a) - L^a(s,a), \tag{4.32}$$

$$(H^a)^*(s,x,b) \coloneqq \max_p p \cdot b - H^a(s,x,p). \tag{4.33}$$

Proof. By (4.14),

$$L^b(s,x,b) = L^x(s,x) + (L^a)^b(s,x,b),$$

where $(L^a)^b(s,x,b) = \min_a L^a(s,a)$ subject to $f(s,x,a) = -b$. By $(L^b)^* \equiv H$ by Lemma 6 and the definition of H in (4.11),

$$H(s,x,p) = -L^x(s,x) + H^a(s,x,p),$$

where $H^a(s,x,p) = ((L^a)^b)^*(s,x,p)$. Then,

$$H^*(s,x,b) = L^x(s,x) + (H^a)^*(s,x,b). \quad \square$$

This shows that the stage cost of the Hopf-Lax formula (H^*) is decomposed into the control-independent (L^x) and control-dependent ($(H^a)^*$) parts similar to the stage cost of the state-constrained problem (4.2) (L) as in (4.30). One observation here is that the control-independent stage cost of the Hopf-Lax formula (L^x) is exactly the same as that of the state-constrained problem (L^x).

Corollary 3. If $L(s,x,a) = L^x(s,x)$,

$$H^*(s,x,b) = L(s,x). \tag{4.34}$$

Lemma 8 presents the convexity condition for the stage cost of the Hopf-Lax formula.

Lemma 8 (Convexity of the stage cost)**.** Suppose Assumption 1 and (4.30) hold. If $L^x(s,\cdot)$ is convex in $x \in \mathbb{R}^n$ for each $s \in [t,T]$ and the dynamics are in the following form:

$$f(s,x,a) = M(s)x + \varphi(s,a), \tag{4.35}$$

where f is the dynamics in (4.4) and $M(\cdot)$ is a time-varying linear matrix, then

$$H^*(s,x,b) = L^x(s,x) + (\bar{H}^a)^*(s, b + M(s)x), \tag{4.36}$$

where

$$\bar{H}^a(s,p) \coloneqq \max_{a \in A}[-p \cdot \varphi(s,a) - L^a(s,a)], \tag{4.37}$$

and $H^*(s,\cdot,\cdot)$ is convex in (x,b) for each $s \in [t,T]$.

Note that $\bar{H}^a(s,p)$ is independent on x and convex in p for each t.

Proof. By (4.32) and (4.33),

$$\begin{aligned} H^a(s,x,p) &= -p\cdot(M(s)x) + \bar{H}^a(s,p), \\ (H^a)^*(s,x,b) &= \max_b p\cdot(b+M(s)x) - \bar{H}^a(s,p) \\ &= (\bar{H}^a)^*(s,b+M(s)x). \end{aligned}$$

By Lemma 7,

$$H^*(s,x,b) = L^x(s,x) + (\bar{H}^a)^*(s,b+M(s)x).$$

Since $(\bar{H}^a)^*(s,\cdot)$ is convex in b and $b+M(s)x$ is affine in (x,b), $(H^a)^*(s,\cdot,\cdot)$ is convex in (x,b). Therefore, $H^*(s,\cdot,\cdot)$ is convex in (x,b) for each $s\in[t,T]$ if $L^x(s,\cdot)$ is convex in x for each $s\in[t,T]$. □

We define the control constraint of the Hopf-Lax formula in (x,b)-space: for $s\in[t,T]$,

$$\bar{B}(s) \coloneqq \{(x,b) \mid b\in \mathrm{co}(B(s,x))\}, \tag{4.38}$$

where $B(s,x)$ is defined in (4.7).

Lemma 9 (Convexity of the control constraint)**.** Suppose Assumption 1 and (4.35) holds. Then $\bar{B}(s)$ in (4.38) is convex in (x,b) for each $s\in[t,T]$.

Proof. Consider (x_1,b_1), $(x_2,b_2)\in\bar{B}(s)$ and $d\in[0,1]$. Since $b_i\in\mathrm{co}(B(s,x_i))$ for $i=1,2$, there exist a finite number of a_{ij} and $\gamma_{ij}\in[0,1]$ ($\sum_j\gamma_{ij}=1$ for each i) such that

$$b_i = -M(s)x_i - \sum_j \gamma_{ij}\varphi(s,a_{ij})$$

for each $i=1,2$. Using this, we have

$$db_1+(1-d)b_2 = -M(s)(dx_1+(1-d)x_2) - \sum_j [d\gamma_{1,j}\varphi(s,a_{1,j}) + (1-d)\gamma_{2,j}\varphi(s,a_{2,j})].$$

Since $\mathrm{co}(\{-\varphi(s,a)\mid a\in A\}$ is a convex set,

$$d(b_1+M(s)x_1)+(1-d)(b_2+M(s)x_2)\in\mathrm{co}(\{-\varphi(s,a)\mid a\in A\}).$$

and

$$db_1+(1-d)b_2\in\mathrm{co}(B(s,dx_1+(1-d)x_2)).$$

Thus, $\bar{B}(s)$ is convex in (x,b). □

Table 4.1 summarizes the convexity conditions for SCGSP ((4.3) subject to (4.4)) and the Hopf-Lax formula ((4.9) subject to (4.10)). If the state-constrained problem is convex, the Hopf-Lax formula is convex. Also, there is a class of problems in which the Hopf-Lax formula is convex even though the state-constrained problem is non-convex.

Remark 9 (Benefits of the Hopf-Lax formula)**.** The proposed formulation converts non-convexity in the control space to convex. For convex Hopf-Lax formula,

1. the control-dependent stage $L^a(s,a)$ in (4.30) is not required to be convex in $a \in A \subset \mathbb{R}^m$ for each $s \in [t,T]$,
2. the control-dependent dynamics $\varphi(s,a)$ in (4.35) is not required to be affine in $a \in A$ for each $s \in [t,T]$,
3. the control constraint A in (4.4) is not required to be convex.

4.1.5 Numerical Example

We introduce an example for the Hopf-Lax formula and illustrate the benefits of the formula. For numerical computation, a computer with a 2.8 GHz Quad-Core i7 CPU and 16 GB RAM was used.

Formation control of multiple nonlinear vehicles

We introduce a 12D nonlinear example, a formation control for multiple agents whose dynamics are nonlinear:

$$f(s,\mathrm{x}^l,\alpha^l(s)) = [\mathrm{x}^l(s,2);\ \alpha^l(s,1)\cos(\alpha^l(s,2)); \mathrm{x}^l(s,4);\ \alpha^l(s,1)\sin(\alpha^l(s,2))] \tag{4.39}$$

where $l \in \{1,2,3\}$ is the agent index, $\mathrm{x}^l(s,1)$ and $\mathrm{x}^l(s,2)$ are horizontal position and velocity in the 2D space, $\mathrm{x}^l(s,3)$ and $\mathrm{x}^l(s,4)$ are vertical position and velocity in the 2D space, and $\alpha^l(s,1)$ and $\alpha^l(s,2)$ are the magnitude of the acceleration and the angle of agent l, respectively, at time $s \in [t,T]$. For three agents, the dimension of the state is twelve, and the dimension of the control is six.

We define a state-constrained optimal control problem where three agents approach the goal point with the right-triangular-shaped formation:

$$\begin{aligned}\inf_{\alpha} \int_0^{10} \max\Bigg\{ &\left\| \begin{bmatrix}\mathrm{x}^1(s,1)\\ \mathrm{x}^1(s,3)\end{bmatrix} - \begin{bmatrix}\mathrm{x}^{1,r}(s,1)\\ \mathrm{x}^{1,r}(s,3)\end{bmatrix} \right\|_2, \left\| \begin{bmatrix}\mathrm{x}^2(s,1)\\ \mathrm{x}^2(s,3)\end{bmatrix} - \begin{bmatrix}\mathrm{x}^1(s,1)\\ \mathrm{x}^1(s,3)\end{bmatrix} - d^r \right\|_2, \\ &\left\| \begin{bmatrix}\mathrm{x}^3(s,1)\\ \mathrm{x}^3(s,3)\end{bmatrix} - h\left(\begin{bmatrix}\mathrm{x}^1(s,1)\\ \mathrm{x}^1(s,3)\end{bmatrix}, \begin{bmatrix}\mathrm{x}^2(t,1)\\ \mathrm{x}^2(t,3)\end{bmatrix} \right) \right\|_2 \Bigg\} ds \end{aligned} \tag{4.40}$$

$$\text{subject to } \begin{cases} (4.39),\ \mathrm{x}^l(0) = x^l, \\ \alpha^l(s,1) \in [-1,3], \quad \alpha^l(s,2) \in [-\frac{\pi}{6}, \frac{\pi}{6}], \end{cases} \tag{4.41}$$

for $s \in [0,T]$, $l = 1,2,3$, where $\mathrm{x}^{1,r}(s,1) = 2s$, $\mathrm{x}^{1,r}(s,3) = 0$, $d^r = [-\sqrt{3};1] \in \mathbb{R}^2$, and, for $w_1, w_2 \in \mathbb{R}^2$, $h : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$, $h(w_1,w_2) = [1/2, \sqrt{3}/2; -\sqrt{3}/2, 1/2]w_1 + [1/2, -\sqrt{3}/2; \sqrt{3}/2, 1/2]w_2$. Agent 1 is the leader that tracks the reference trajectory $\mathrm{x}^{1,r}$,

for which the first term of the state cost is designed. Agent 2 is following Agent 1 with d^r-offset, designed in the second term of the stage cost. Agent 3 is making the right-triangular formation, for which the third term of the stage cost is designed.

By Remark 9, the optimal control problem is non-convex, but the Hopf-Lax formula is convex.

By Corollary 3, the stage cost for the Hopf-Lax formula (H^*) is equal to the stage cost of the given problem (L) since the control-dependent stage cost (L^a) in (4.30) is zero. To derive the control constraint ($\mathrm{co}(B(s,\mathrm{x}(s)))$) for the Hopf-Lax formula, we use Lemma 6. Denote $\mathrm{x}(s) = [\mathrm{x}^1(s);\mathrm{x}^2(s);\mathrm{x}^3(s)]$, $\beta(s) = [\beta^1(s);\beta^2(s);\beta^3(s)]$ and $\beta^l(s) = [\beta^l(s,1);\beta^l(s,2);\beta^l(s,3);\beta^l(s,4)]$, $l = 1,2,3$. $B(s,\mathrm{x}(s))$ in (4.7) is illustrated in the grey in Figure 4.2 (a). Then, $\mathrm{co}(B(s,x))$ is derived in the last five lines in (4.43) and also illustrated in Figure 4.2, the union of the grey and the blue. Note that $B(s,\mathrm{x}(s))$ is non-convex, but $\mathrm{co}(B(s,\mathrm{x}(s)))$ is convex in b.

The Hopf-Lax formula in Theorem 6 provides the following optimal control problem:

$$\begin{aligned}\inf_{\alpha}\int_0^{10}\max\Bigg\{&\left\|\begin{bmatrix}\mathrm{x}^1(s,1)\\ \mathrm{x}^1(s,3)\end{bmatrix}-\begin{bmatrix}\mathrm{x}^{1,r}(s,1)\\ \mathrm{x}^{1,r}(s,3)\end{bmatrix}\right\|_2,\left\|\begin{bmatrix}\mathrm{x}^2(s,1)\\ \mathrm{x}^2(s,3)\end{bmatrix}-\begin{bmatrix}\mathrm{x}^1(s,1)\\ \mathrm{x}^1(s,3)\end{bmatrix}-d^r\right\|_2,\\ &\left\|\begin{bmatrix}\mathrm{x}^3(s,1)\\ \mathrm{x}^3(s,3)\end{bmatrix}-h\left(\begin{bmatrix}\mathrm{x}^1(s,1)\\ \mathrm{x}^1(s,3)\end{bmatrix},\begin{bmatrix}\mathrm{x}^2(t,1)\\ \mathrm{x}^2(t,3)\end{bmatrix}\right)\right\|_2\Bigg\}ds\end{aligned} \tag{4.42}$$

$$\text{subject to }\begin{cases}\dot{\mathrm{x}}^l(s)=-\beta^l(s),\quad \mathrm{x}^l(0)=x^l,\\ \beta^l(s,1)=-\mathrm{x}^l(s,2),\quad \beta^l(s,3)=-\mathrm{x}^l(s,4),\\ -\beta^l(s,2)-\sqrt{9-(\beta^l(s,4))^2}\le 0,\\ \beta^l(s,2)-\sqrt{1-(\beta^l(s,4))^2}\le 0,\\ \frac{1}{2\sqrt{3}}\beta^l(s,2)+\beta^l(s,4)-\frac{3}{4}\le 0,\\ \frac{1}{2\sqrt{3}}\beta^l(s,2)-\beta^l(s,4)-\frac{3}{4}\le 0,\end{cases} \tag{4.43}$$

for $s\in[0,10]$, $l=1,2,3$. For numerical optimization, we utilize Euler-forward discretization and the interior-point method [19]. We denote a numerical optimal state sequence $\mathrm{x}_*[\cdot]$ and control sequence $\beta_*[\cdot]$ for the Hopf-Lax formula. Also, $\mathrm{x}_*[k] = [\mathrm{x}_*^1[k];\mathrm{x}_*^2[k];\mathrm{x}_*^3[k]]$ and $\beta_*[k] = [\beta_*^1[k];\beta_*^2[k];\beta_*^3[k]]$.

Algorithm 3 provides the numerical optimal control signal ($\alpha_*^\epsilon : [0,10]\to\mathbb{R}^6$) in (4.20) and state trajectory ($\mathrm{x}_*^\epsilon : [0,10]\to\mathbb{R}^{12}$) in (4.21). To numerically get α_*^ϵ in (4.20), we need to find $\beta_{i*}^l[k]\in B^l(t_k,\mathrm{x}_*^l[k])$ and $\gamma_i^l[k]\in[0,1]$ ($l=1,2,3$) such that

$$\beta_*^l[k] = \gamma_1^l[k]\beta_{1*}^l[k] + \gamma_2^l[k]\beta_{2*}^l[k] \tag{4.44}$$

for $\beta_*^l[k]\in\mathrm{co}(B^l(t_k,\mathrm{x}_*^l[k]))$ as in Corollary 2. By substituting t_k for s and $\mathrm{x}_*^l[k]$ for $\mathrm{x}^l(s)$, Figure 4.2 graphically illustrates two cases to find $\beta_{i*}^l[k]$ and $\gamma_i^l[k]$: if $\beta_*^l[k]$ is in $B^l(t_k,\mathrm{x}_*^l[k])$ or not. For each case, the mathematical expression for β_{i*}^l can be found in Figure 4.2. Then, we can design

$$\alpha_*^{\epsilon l}(s) = \begin{cases}\alpha_{1*}^l[k], & s\in[t_k, t_k+\Delta t_k\gamma_1^l[k]),\\ \alpha_{2*}^l[k], & s\in[t_k+\Delta t_k\gamma_1^l[k], t_{k+1}),\end{cases} \tag{4.45}$$

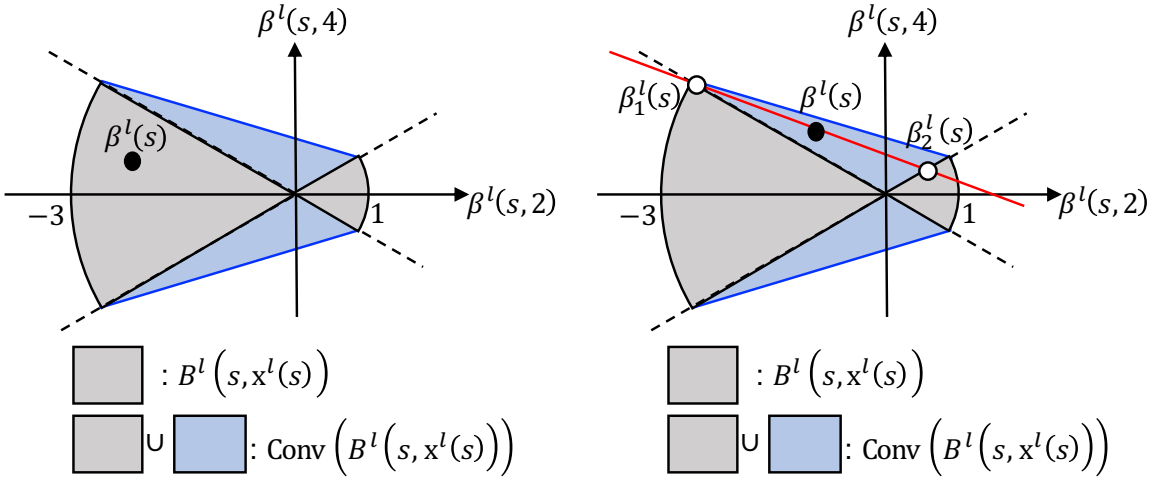


Figure 4.2: The control constraint for the 12D formation control problem in Chapter 4.1.5. l is the index of the agents in $\{1,2,3\}$. The equations of the two dotted lines for the both figures are $\sqrt{3}\beta^l(s,4) \pm \beta^l(s,2) = 0$. These figures illustrate how to find a finite $\beta_i^l(s)$ satisfying the control decomposition in Corollary 2 for $(s,\mathrm{x}^l(s))$ by dividing into two cases. (a) If $\beta^l(s) \in B^l(s,\mathrm{x}^l(s))$, we do not need the control decomposition in Corollary 2. (b) If $\beta^l(s) \notin B^l(s,\mathrm{x}^l(s))$, there are multiple choices of $\beta_{i*}^l(s)$. Among the choices, we select $\beta_1^l(s) = [-\mathrm{x}^l(s,2); -\frac{3\sqrt{3}}{2}; -\mathrm{x}^l(s,4); \pm\frac{3}{2}]$ and find $\beta_2^l(s)$: the intersection of the red line connecting $\beta^l(s)$ and $\beta_1^l(s)$, and one of the two dotted lines.

where $\alpha_{1*}^l[k]$ and $\alpha_{2*}^l[k]$ satisfy $\beta_{1*}^l[k] = -f(t_k, \mathrm{x}_*^l[k], \alpha_{1*}^l[k])$ and $\beta_{2*}^l[k] = -f(t_k, \mathrm{x}_*^l[k], \alpha_{2*}^l[k])$, and finally compute $\mathrm{x}_*^{\epsilon l}$ solving (4.39) for $\alpha_*^{\epsilon l}$ for each agent.

Figure 4.3 demonstrates the numerical result computed by the Hopf-Lax formula and Algorithm 3. In Figure 4.3 (a), the solid and dotted lines are state trajectories, which are numerical solutions to the Hopf-Lax formula and the original state-constrained problem, respectively, using the interior-point method. The two sets of trajectories are different for x between 0 and 5, and then come together. Global optimality is guaranteed for the Hopf-Lax formula with the cost of 5.6, but solving the non-convex state-constrained problem results in the cost of 7.6. Figure 4.3 (b) shows the optimal state trajectories of each agent with the formation at each second.

The computation time for Algorithm 3 is 152 s. On the other hand, it is not realistic to numerically solve the HJ PDE in [4] using grid-based methods, such as the level-set method and the fast marching method [93], due to their exponential complexity in computation when the dimension of the state is more than five or six.

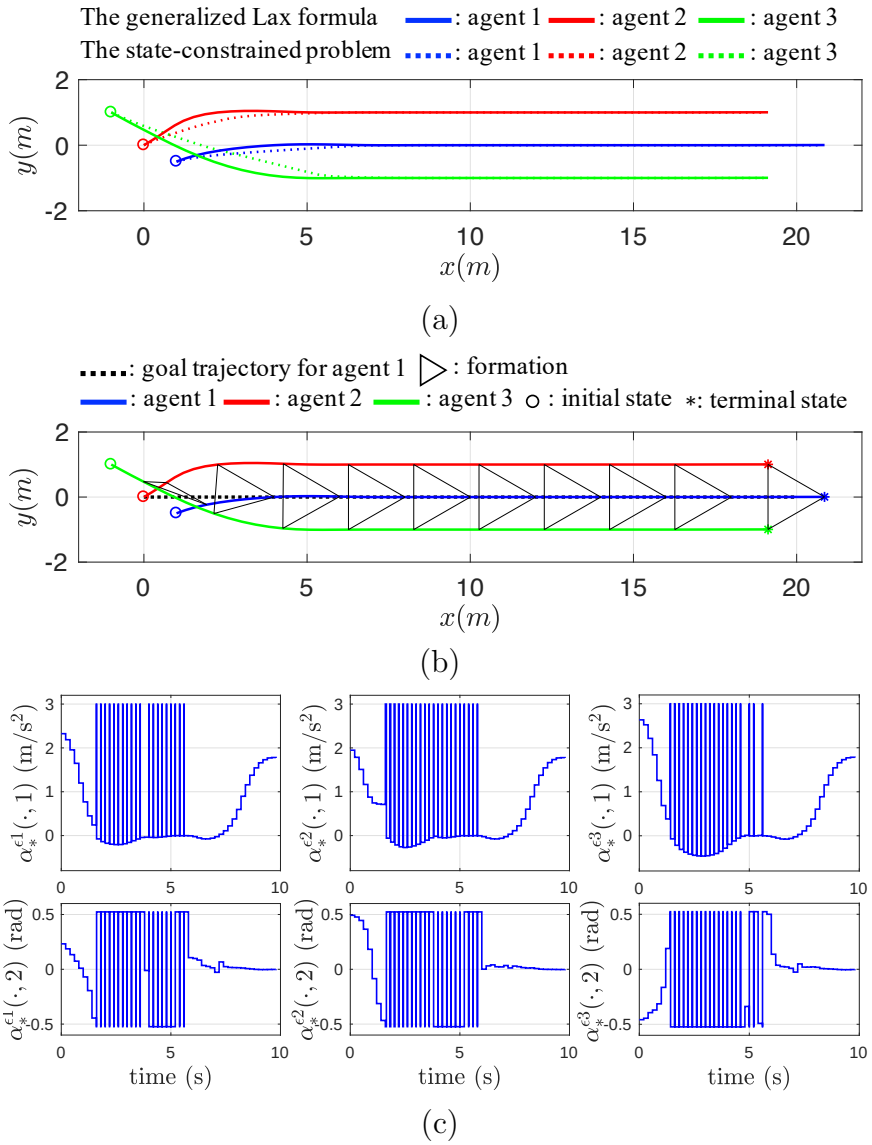


Figure 4.3: The numerical results for the 12D formation control in Chapter 4.1.5 using the Hopf-Lax formula and Algorithm 3. (a) The results of solving the state-constrained problem and its Hopf-Lax formula are shown. The solid lines are globally optimal, but the dotted lines are locally optimal. (b) The optimal state trajectories for all agents in the 2D-space are shown with triangular formation at each second. (c) The optimal control signals for all agents are shown. We observe that the control chattering is caused by the additional discretization step in Algorithm 3.

Chattering Control Signal

A disadvantage of the proposed method is the chattering in the control signal caused by the additional temporal discretization in line 6 in Algorithm 3, which is not desirable in practice.

This additional discretization is not necessary if the given optimal control problem is convex in the control. In this case, if a control signal for the Hopf-Lax formula is non-chattering, Algorithm 3 provides a non-chattering control signal. Unfortunately, the chattering is unavoidable for the non-convex problem in the control if the proposed method is utilized.

It has been observed that this chattering behavior of optimal control signals cannot be avoided for some state-constrained problems, such as Fuller's problem [23], and it is also hard to find the corresponding conditions under which the chattering is unavoidable [21, 7, 108]. To avoid this chattering control for practical issues, some approximation methods have been developed [23].

Cooperation with Closed-Loop Methods

As one of the open-loop control methods, the proposed method can cooperate with closed-loop framework, such as model predictive control (MPC) [53]. MPC allows real-time computation by considering a receding-horizon problem and accommodates estimation error, measurement noise, and unexpected disturbances in practice.

4.2 Hopf-Lax Formula for State-Constrained Control-Invariance Problems

This chapter is based on the work presented in [68], which is joint work with Claire Tomlin.

This chapter presents the Hopf-Lax formula for SCCIP. Hopf-Lax theory assumes convex Hamiltonians, and the HJ PDE's Hamiltonian relevant to the time-varying SCCIP is convex in the costate space but the one to the time-invariant SCCIP is non-convex. Thus, this chapter only presents the Hopf-Lax formula for the time-varying SCCIP. To prove the equivalence between SCCIP and the Hopf-Lax formula, this paper first proposes a new viscosity result that specifies sufficient conditions under which two different PDEs have the same solution, and then applies this to the HJ PDEs for SCCIP and the Hopf-Lax formula. This proof technique can generally apply to Hopf-Lax theory relevant to state-constrained optimal control problems. Computational advantages of the Hopf-Lax formulation are then presented. For SCCIP, the proposed Hopf-Lax formula convexifies the problem in the control-input space.

The organization of this chapter is as follows. Chapter 4.2.1 defines SCCIP. Chapter 4.2.2 presents the HJ PDE whose solution's sub-zero-level set is the epigraph of SCCIP. Our Hopf-Lax formula is presented in Chapter 4.2.3, and Chapter 4.2.4 provides proofs. Chapter 4.2.4 proposes our new theorem in the viscosity theory, which is utilized to prove the Hopf-Lax formulae for SCCIP. Chapter 4.2.5 presents convexity analysis for SCCIP and its Hopf-Lax formula. Chapter 4.2.6 presents an example to demonstrate the utility and performance of the proposed Hopf-Lax formula.

4.2.1 State-Constrained Control-Invariance Problem

This chapter recalls SCCIP and the corresponding HJ PDE presented in Chapter 3.1.

In this chapter, we consider the state trajectory ($\mathrm{x} : [t, T] \to \mathbb{R}^n$) in a time interval $[t, T]$ solving the following ordinary differential equation (ODE):

$$\dot{\mathrm{x}}(s) = f(s, \mathrm{x}(s), \alpha(s)), s \in [t, T], \text{ and } \mathrm{x}(t) = x, \tag{4.46}$$

where (t, x) are the initial time and state, s is time between t and T, $f : [t, T] \times \mathbb{R}^n \times A \to \mathbb{R}^n$ is a dynamics function, $A \subset \mathbb{R}^m$ is the control set, $\alpha \in \mathcal{A}(t)$ is the control signal, and we denote the set of measurable control signals

$$\mathcal{A}(t) \coloneqq \{\alpha : [t, T] \to A \mid \|\alpha\|_{L^\infty(t,T)} < \infty\}. \tag{4.47}$$

We assume that A is a compact subset in $\mathbb{R}^m$.

State-constrained control-invariance problem (SCCIP): for given initial time and state (t, x), solve

$$\vartheta_1(t, x) \coloneqq \inf_{\alpha \in \mathcal{A}} \max_{\tau \in [t,T]} \int_t^\tau L(s, \mathrm{x}(s), \alpha(s))ds + g(\tau, \mathrm{x}(\tau)) \tag{4.48}$$

$$\text{subject to} \quad c(s, \mathrm{x}(s)) \le 0, \quad s \in [t, T], \tag{4.49}$$

where x solves (4.46), $L : [t, T] \times \mathbb{R}^n \times A \to \mathbb{R}$ is the stage cost, $g : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ is the terminal cost, f is the system dynamics as defined above, and $c : [t, T] \times \mathbb{R}^n \to \mathbb{R}$ is the state constraint. The scalar function c can handle a number of state constraints. For example, consider state constraints $c_1, ..., c_l(s, \mathrm{x}(s)) \le 0$; $c = \max\{c_1, .., c_l\}$ encodes all state constraints into a scalar constraint function. We would like to note that $\vartheta_1(t, x) \in \mathbb{R} \cup \infty$. This means that, for initial time and state (t, x), if there does not exist a control signal α to satisfy the state constraint (4.49), the optimal cost ϑ_1 is ∞.

Assumption 1 guarantees the existence of a unique solution to the HJ PDEs, presented in Chapter 4.2.2. In (4.49), the state constraint is satisfied in $[t, T]$, but we can replace this interval with $[t, \tau]$ [67], which does not change the optimal value. Consider an optimal control signal that minimizes (4.48) subject to the state constraint (4.49) with $[t, \tau]$. If the optimal value is finite, the state constraint has to be satisfied for all time in $[t, T]$. Otherwise, the time maximizer τ will choose the time when the state constraint is not satisfied so that the optimal value becomes infinity.

4.2.2 Hamilton-Jacobi PDEs for SCCIP

This section reviews the previously presented HJ PDEs for SCCIP and SCRAP [67, 69], which will be utilized to derive Hopf-Lax formulae in Chapter 4.2.3. For each problem, this section presents two HJ PDEs, one for the time-varying case and a second for the time-invariant case. The time-varying case features time varying cost functions, dynamics, and state constraints.

Hamilton-Jacobi equation for SCCIP

[67] utilizes the epigraphical technique to derive an HJ PDE whose solution characterizes the epigraph of the optimal cost (ϑ_1) for the SCCIP. In this formulation, we first encode the cost and constraint of the SCCIP into a state-augmented value function V_1 (4.51) whose sub-zero-level set is the epigraph of the optimal cost (ϑ_1) for SCCIP:

$$\begin{aligned}\operatorname{epi}(\vartheta_1(t,\cdot)) &\coloneqq \{(x,z) \mid z \geq \vartheta_1(t,x)\} \\ &= \{(x,z) \mid V_1(t,x,z) \leq 0\}.\end{aligned} \tag{4.50}$$

Then, the dynamic programming principle is applied to derive the HJ PDE for V_1.

Define the augmented value function $V_1 = V_1(t,x,z) : [0,T] \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$

$$V_1(t,x,z) \coloneqq \inf_{\alpha \in \mathcal{A}(t)} \max\Big\{ \max_{s\in[t,T]} c(s,\mathrm{x}(s)), \max_{\tau\in[t,T]} \int_t^\tau L(s,\mathrm{x}(s),\alpha(s))\mathrm{d}s + g(\tau,\mathrm{x}(\tau)) - z \Big\}, \tag{4.51}$$

where x solves (4.46), (t,x) are initial time and states, and z is a new scalar variable that represents a value axis for the epigraph of ϑ_1. Now, we will consider (x,z) as an augmented state in V_1.

V_1 is continuous in (t,x,z)-space, and standard viscosity theory works for V_1. Theorem 8 presents HJ PDEs for V_1 and finds ϑ_1 from V_1.

Theorem 8 (HJ PDE for SCCIP)**.** Suppose Assumption 1 holds. V_1 in (4.51) is the unique viscosity solution to the HJ PDE:

$$\max\left\{ c(t,x) - V_1(t,x,z), g(t,x) - z - V_1(t,x,z), \frac{\partial V_1}{\partial t} - \bar{H}\left(t,x,z,\frac{\partial V_1}{\partial x},\frac{\partial V_1}{\partial z}\right) \right\} = 0 \tag{4.52}$$

in $(0,T)\times\mathbb{R}^n\times\mathbb{R}$, where $\bar{H} : [0,T]\times\mathbb{R}^n\times\mathbb{R}\times\mathbb{R}^n\times\mathbb{R}\to\mathbb{R}$

$$\bar{H}(t,x,z,p,q) \coloneqq \max_{a\in A}[-p\cdot f(t,x,a) + qL(t,x,a)], \tag{4.53}$$

where p and q represent the gradients $\frac{\partial V_1}{\partial x}$ and $\frac{\partial V_1}{\partial z}$, and

$$V_1(T,x,z) = \max\{c(T,x), g(T,x) - z\} \tag{4.54}$$

on $\{t=T\}\times\mathbb{R}^n\times\mathbb{R}$.

For the time-invariant case, the above HJ PDE (4.52) is simplified to

$$\max\left\{ c(x) - V_1(t,x,z), \frac{\partial V_1}{\partial t} - \bar{H}_1^{\mathrm{TI}}\left(x,z,\frac{\partial V_1}{\partial x},\frac{\partial V_1}{\partial z}\right) \right\} = 0 \tag{4.55}$$

in $(0,T)\times\mathbb{R}^n\times\mathbb{R}$, where

$$\bar{H}_1^{\mathrm{TI}}(x,z,p,q) \coloneqq \min\{0, \bar{H}(x,z,p,q)\} \tag{4.56}$$

for $(x, z, p, q) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}$, and $\bar{H}$ is defined in (4.53). In (4.56), $\bar{H}$ does not have the time dependency for the time-invariant case.

Then,

$$\vartheta_1(t, x) = \min z \text{ subject to } V_1(t, x, z) \leq 0. \tag{4.57}$$

Our Hopf-Lax formulae in Chapter 4.2.3 will assume convex Hamiltonians in the gradient space. $\bar{H}$ (4.53) is convex in (p, q), but $\bar{H}_1^{\text{TI}}$ (4.55) is not. Thus, Chapter 4.2.3 presents a Hopf-Lax formula for time-varying SCCIP.

4.2.3 Hopf-Lax Formulae for SCCIP

For SCCIP, the HJ PDEs in Theorems 8 can be numerically solved by grid-based methods, such as the level-set methods [85] and fast marching method [93]. These methods require spatial and temporal discretization, which leads to exponential computational complexity in the state's dimension. Thus, it is intractable to utilize these grid-based methods for high-dimensional systems [10]. This chapter provides more discussion about the computational complexity in Chapter 4.3.6.

This chapter presents our main results to alleviate this computational complexity: Hopf-Lax formulae for SCCIP in Chapter 4.2.3. The proof of our formulation will be presented in Chapter 4.2.4.

Hopf-Lax formula for SCCIP

Define, for initial time and state $(t, x) \in [0, T] \times \mathbb{R}^n$,

$$\varphi_1(t, x) \coloneqq \inf_{\beta} \max_{\tau \in [t,T]} \int_t^\tau H^*(s, \mathrm{x}(s), \beta(s))ds + g(\tau, \mathrm{x}(\tau)), \tag{4.58}$$

$$\text{subject to } \begin{cases} \dot{\mathrm{x}}(s) = -\beta(s), s \in [t, T] \\ \beta(s) \in \text{co}(\{-f(s, \mathrm{x}(s), a) \mid a \in A\}), s \in [t, T] \\ \mathrm{x}(t) = x, \\ c(s, \mathrm{x}(s)) \leq 0, s \in [t, T] \end{cases} \tag{4.59}$$

where $\beta : [t, T] \to \mathbb{R}^n$ is a new measurable control signal, "co" is a convex-hull operator, $H : [0, T] \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$

$$\begin{aligned} H(s, x, p) &\coloneqq \bar{H}(s, x, z, p, -1) \\ &= \max_{a \in A}[-p \cdot f(s, x, a) - L(s, x, a)], \end{aligned} \tag{4.60}$$

where p is the costate with respect to x, $\bar{H}$ is defined in (4.53), and $H^* : [0, T] \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$

$$H^*(s, x, b) \coloneqq \max_{p \in \mathbb{R}^n}[p \cdot b - H(s, x, p)], \tag{4.61}$$

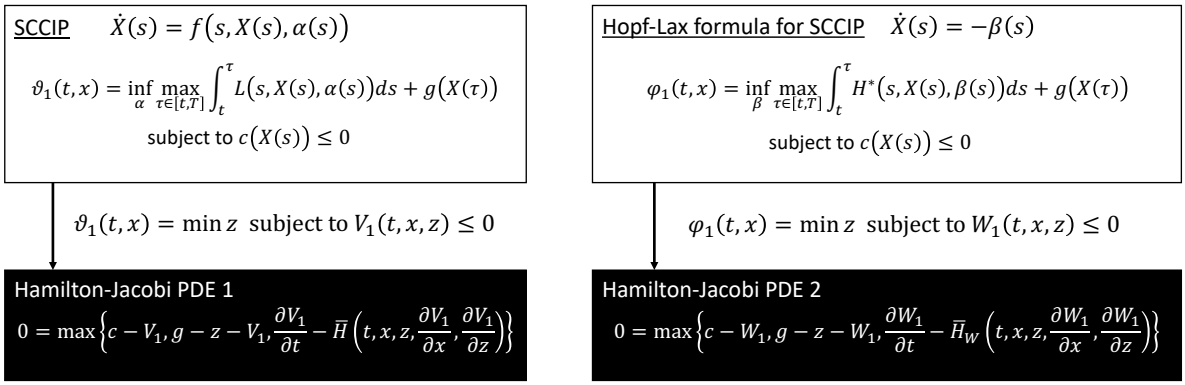


Figure 4.4: Proof overview of Hopf-Lax theory for SCCIP

where b is a new control input, and H^* is the Legendre-Fenchel transformation (the convex conjugate) of H with respect to $p \in \mathbb{R}^n$. b and p are dual variables. For each s, x, H^* is finite if $b \in \text{co}(\{-f(s, \text{x}(s), a) \mid a \in A\})$ but, otherwise, is infinite.

Theorem 9. (Hopf-Lax formula for SCCIP) For all $(t, x) \in [0, T] \times \mathbb{R}^n$,

$$\vartheta_1(t, x) = \varphi_1(t, x), \tag{4.62}$$

where ϑ_1 is the optimal cost of SCCIP ((4.48) subject to (4.49)), and φ_1 is the Hopf-Lax formula for SCCIP ((4.58) subject to (4.59)).

4.2.4 Proof of Hopf-Lax Theory for SCCIP

Figure 4.4 shows a proof overview of Hopf-Lax theory for SCCIP. In this proof, we consider two value functions: V_1 is the value function whose sub-zero-level set is the epigraph of ϑ_1, and W_1 is the one of φ_1, where W_1 will be defined in (4.73). We will prove $\vartheta_1 \equiv \varphi_1$ by showing that V_1 and W_1 are the same. This chapter later will show the two corresponding HJ PDEs for V_1 and W_1 are different but V_1 and W_1 are the same. In this chapter, we propose a general theorem in the viscosity theory that specifies sufficient conditions under which two different first-order PDEs have the same solution. Using this theorem, we will prove $V_1 \equiv W_1$, which concludes $\vartheta_1 \equiv \varphi_1$.

Viscosity theory with sufficient condition under which two different PDEs have the same solution

This chapter proposes a general theorem in viscosity theory to investigate the equivalence of two first-order PDEs. We use a more general notation in this chapter.

Consider time $t \in [0,T]$, a state that consists of $x \in \mathbb{R}^{n_x}$, $z \in \mathbb{R}^{n_z}$, and two first-order PDEs:

$$0 = F_i\Big(t,x,z,X_i(t,x,z),\frac{\partial X_i}{\partial t}(t,x,z),\frac{\partial X_i}{\partial x}(t,x,z),\frac{\partial X_i}{\partial z}(t,x,z)\Big) \tag{4.63}$$

in $(0,T)\times\mathbb{R}^{n_x}\times\mathbb{R}^{n_z}$ for $i=1,2$, and the terminal values for X_1 and X_2 are the same as $l = l(x,z) \in \mathbb{R}$:

$$X_1(T,x,z) = X_2(T,x,z) = l(x,z) \quad \forall (x,z) \in \mathbb{R}^{n_x}\times\mathbb{R}^{n_z}. \tag{4.64}$$

We say that $X_i = X_i(t,x,z)$ $(i=1,2)$ is the viscosity solution of F_i if (i) $X_i(T,x,z) = l(x,z)$ and, (ii) for each smooth function $U : (0,T)\times\mathbb{R}^{n_x}\times\mathbb{R}^{n_z}\to\mathbb{R}$,

1. if $X_i - U$ has a local maximum at a point $(t_0,x_0,z_0) \in (0,T)\times\mathbb{R}^{n_x}\times\mathbb{R}^{n_z}$ and $(X_i-U)(t_0,x_0,z_0) = 0$,

$$F_i\Big(t_0,x_0,z_0,U(t_0,x_0,z_0),\frac{\partial U}{\partial t}(t_0,x_0,z_0),\frac{\partial U}{\partial x}(t_0,x_0,z_0),\frac{\partial U}{\partial z}(t_0,x_0,z_0)\Big) \geq 0, \tag{4.65}$$

2. if $X_i - U$ has a local minimum at a point $(t_0,x_0,z_0) \in (0,T)\times\mathbb{R}^{n_x}\times\mathbb{R}^{n_z}$ and $(X_i - U)(t_0,x_0,z_0) = 0$,

$$F_i\Big(t_0,x_0,z_0,U(t_0,x_0,z_0),\frac{\partial U}{\partial t}(t_0,x_0,z_0),\frac{\partial U}{\partial x}(t_0,x_0,z_0),\frac{\partial U}{\partial z}(t_0,x_0,z_0)\Big) \leq 0. \tag{4.66}$$

For SCCIP, F_1 refers to the HJ PDE (4.52), n_x to n, n_z to 1, and X_1 to V_1. F_2 refers to the HJ PDE (4.75) that will be introduced later in Chapter 4.2.4, X_2 to W_1 (4.73) also in Chapter 4.2.4. Also, $l(x,z)$ refers to the terminal condition for V_1 and W_1: $\max\{c(T,x), g(T,x) - z\}$. For SCRAP, we have a similar notation matching rule with SCCIP.

In the notion of the viscosity theory, we present conditions under which the two different PDEs F_1 and F_2 have the same solution. Define super-differentials and sub-differentials of X_i $(i=1,2)$ with respect to z: for each $(t,x,z) \in [0,T]\times\mathbb{R}^{n_x}\times\mathbb{R}^{n_z}$, $q \in \partial_z^+ X_i(t,x,z)$ $(i=1,2)$ is a super-differential with respect to z, if

$$\partial_z^+ X_i(t,x,z) \coloneqq \Big\{q \mid \limsup_{\bar{z}\to 0}\frac{X_i(t,x,z+\bar{z}) - X_i(t,x,z) - q\cdot\bar{z}}{\|\bar{z}\|} \leq 0\Big\}, \tag{4.67}$$

and $q \in \partial_z^- X_i(t,x,z)$ $(i=1,2)$ is a sub-differential with respect to z, if

$$\partial_z^- X_i(t,x,z) \coloneqq \{q \mid \limsup_{\bar{z}\to 0}\frac{X_i(t,x,z+\bar{z}) - X_i(t,x,z) - q\cdot\bar{z}}{\|\bar{z}\|} \geq 0\}. \tag{4.68}$$

Theorem 10 states that if two different PDEs (F_1 and F_2) are the same in the super-differential or sub-differential domains in z, the two PDEs' solutions are the same.

Theorem 10. Suppose each of the two first-order PDEs in (4.63) with the terminal value (4.64) for $i = 1, 2$ has the unique solution (X_i). If, for all $(t, x, z, X, r, p) \in [0, T] \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n_x}$, $q \in \partial_z^+ X_1(t, x, z) \cup \partial_z^- X_1(t, x, z)$,

$$F_1(t, x, z, X, r, p, q) = F_2(t, x, z, X, r, p, q) \tag{4.69}$$

then $X_1 \equiv X_2$. Here, r, p, q are the costates with respect to t, x, and z, respectively.

Proof. See Appendix D.1.

Proof of the Hopf-Lax formula for time-vaying SCCIP

In this chapter, we utilize Theorem 10 to prove Theorem 9 for the time-varying SCCIP.

We first investigate the super-differentials and sub-differentials of V_i $(i = 1, 2)$ with respect to z.

Lemma 10 (Convexity of the value function in z)**.** For each $(t, x) \in [0, T] \times \mathbb{R}^n$, $V_1(t, x, \cdot)$ (4.51) is convex in $z \in \mathbb{R}$: for all $z_1, z_2 \in \mathbb{R}$ and $\theta \in [0, 1]$,

$$V_1(t, x, \theta_1 z_1 + \theta_2 z_2) \leq \theta_1 V_1(t, x, z_1) + \theta_2 V_1(t, x, z_2) \tag{4.70}$$

for $i = 1, 2$.

Proof. See Appendix D.2.

Lemma 11. For all $(t, x, z) \in [0, T] \times \mathbb{R}^n \times \mathbb{R}$ and $i = 1, 2$,

$$\partial_z^- V_1(t, x, z) \subset [-1, 0], \tag{4.71}$$

and if $\partial_z^+ V_1(t, x, z)$ is not the empty set, the set of super-differentials with respect to z is a singleton

$$\partial_z^+ V_1(t, x, z) = \left\{ \frac{\partial V_1}{\partial z}(t, x, z) \right\} \subset [-1, 0]. \tag{4.72}$$

Note that V_1 is defined in (4.51), respectively.

Proof. See Appendix D.3.

By applying the HJ analysis in Chapter 4.2.2, we have following results for the Hopf-Lax formula for SCCIP (φ_1 (4.58) subject to (4.59)). Combining (4.58) and (4.59), define a value function in the augmented state space, $W_1 : [0, T] \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$:

$$W_1(t, x, z) = \inf_{\beta} \max \left\{ \max_{s \in [t,T]} c(s, \mathrm{x}(s)), \max_{\tau \in [t,T]} \int_t^\tau H^*(s, \mathrm{x}(s), \beta(s)) ds + g(\tau, \mathrm{x}(\tau)) - z \right\}, \tag{4.73}$$

where $\beta(s) \in \text{co}(\{-f(s, \text{x}(s), a) \mid a \in A\})$, and "co" is the convex-hull operator. As shown in [65],

$$\begin{aligned} \text{Dom}(H^*(s, \text{x}(s), \cdot)) &= \{b \mid H^*(s, \text{x}(s), b) < \infty\} \\ &= \text{co}(\{-f(s, \text{x}(s), a) \mid a \in A\}). \end{aligned} \tag{4.74}$$

Thus, it is not necessary to add the control constraint in the infimum operation in (4.73) since H^* becomes the infinity out of the control constraint (4.74).

By Theorem 8, W_1 is the unique viscosity solution to

$$\max\left\{c(t,x) - W_1(t,x,z), g(t,x) - z - W_1(t,x,z), \frac{\partial W_1}{\partial t} - \bar{H}_W(t,x,z,\frac{\partial W_1}{\partial x}, \frac{\partial W_1}{\partial z})\right\} = 0 \tag{4.75}$$

in $(0,T) \times \mathbb{R}^n \times \mathbb{R}$, where $\bar{H}_W : [0,T] \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$

$$\bar{H}_W(t,x,z,p,q) \coloneqq \max_b [p \cdot b + qH^*(t,x,b)], \tag{4.76}$$

where p and q represent the gradients $\frac{\partial W_1}{\partial x}$ and $\frac{\partial W_1}{\partial z}$, H^* is defined in (4.61), and

$$W_1(T,x,z) = \max\{c(T,x), g(T,x) - z\} \tag{4.77}$$

on $\{t = T\} \times \mathbb{R}^n \times \mathbb{R}$. Then,

$$\varphi(t,x) = \min z \text{ subject to } W_1(t,x,z) \leq 0. \tag{4.78}$$

Now, it is sufficient to prove $V_1 \equiv W_1$.

We will utilize Theorem 10 to prove $V_1 \equiv W_1$. F_1 and F_2 in Chapter 4.2.4 refer to (4.52) and (4.75), respectively. Also, Lemma 12 analyzes the relationship between the two Hamiltonians: $\bar{H}$ (4.53) for V_1 and $\bar{H}_W$ (4.76) for W_1.

Lemma 12. For $(t,x,z,p,q) \in [0,T] \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}$,

$$\bar{H}(t,x,z,p,q) = \bar{H}_W(t,x,z,p,q) \quad \text{ if } q \leq 0, \tag{4.79}$$

where $\bar{H}$ and $\bar{H}_W$ are defined in (4.53) and (4.76), respectively.

Proof. See Appendix D.4.

Now, we are ready to conclude the proof of Theorem 9.

Proof of Theorem 9. Lemma 11 states that the sub-differentials and super-differentials of $V_1(t,x,z)$ with respect to z are less than or equal to 0 for all (t,x,z). Thus, by combining Theorem 10, Lemma 11 for $i = 1$, and Lemma 12, we prove $V_1 \equiv W_1$. By (4.57) and (4.78), we conclude $\vartheta_1 \equiv \varphi_1$. $\square$

The proof of Theorem 12 is very similar to the above argument in this chapter.

4.2.5 Convexity analysis for SCCIP and its Hopf-Lax formula

We can solve SCCIP and Hopf-Lax formula by utilizing gradient-based methods and temporal discretization. This chapter investigates convexity conditions under which gradient-based methods guarantee optimality. For the temporal discretization, we can choose, for example, the backward Euler method, Crank-Nicolson method, and high-order Runge-Kutta methods [22]. For any discretization method, the convexity analysis in this Chapter 4.2.5 is valid. This chapter utilizes the first-order forward Euler method.

We discretize the time interval to $\{t_0 = 0, ..., t_K = T\}$ where $\Delta_k \coloneqq t_{k+1} - t_k$. In the use of the first-order Euler method, discretization error for optimal control problems becomes smaller for smaller Δ_k [18, 3]. For notation, the state at t_k is $\mathrm{x}[k]$, and $\alpha[k]$ and $\beta[k]$ are control inputs at t_k. In this paper, we use x to denote both a state trajectory in the continuous-time setting ($\mathrm{x}(\cdot)$) and a state sequence in the discrete-time setting ($\mathrm{x}[\cdot]$). We apply the same notation rule for α, β.

Consider the temporally discretized SCCIP:

$$\vartheta_1(0,x) \simeq \min_{\mathrm{x}[\cdot],\alpha[\cdot]} \max_{k' \in \{0,...,K\}} \sum_{k=0}^{k'} L(t_k, \mathrm{x}[k], \alpha[k])\Delta_k + g(t_{k'}, \mathrm{x}[k']), \tag{4.80}$$

$$\text{subject to } \begin{cases} \mathrm{x}[k+1] - \mathrm{x}[k] = \Delta_k f(t_k, \mathrm{x}[k], \alpha[k]), & k \in \{0, ..., K-1\}, \\ \alpha[k] \in A, & k \in \{0, ..., K-1\}, \\ \mathrm{x}[0] = x, & \\ c(t_k, \mathrm{x}[k]) \leq 0, & k \in \{0, ..., K\}. \end{cases} \tag{4.81}$$

The temporal discretized SCCIP ((4.80)) subject to (4.81)) is convex in $(\mathrm{x}[\cdot], \alpha[\cdot])$-space if Assumption 2 holds. Since a pointwise maximum of convex functions is convex, the cost in (4.80) is convex if $\sum_{k=0}^{k'} L(t_k, \mathrm{x}[k], \alpha[k])\Delta_k + g(t_{k'}, \mathrm{x}[k'])$ is convex for each k'. Thus, the first two conditions in Assumption 2 are sufficient to have convex cost in (4.80). The other three conditions are for convex constraints (4.81).

Assumption 2. (Convexity conditions for the temporally discretized SCCIP)

1. $L(t, x, a)$ is convex in (x, a) for all $t \in [0, T]$,
2. $g(t, x)$ is convex in x for all $t \in [0, T]$,
3. $c(t, x)$ is convex in x for all $t \in [0, T]$,
4. $f(t, x, a)$ is affine in (x, a) for all $t \in [0, T]$,
5. A is convex.

We discretize the Hopf-Lax formula for SCCIP (φ_1 in (4.58) subject to (4.59)) as following:

$$\varphi_1(0,x) \simeq \min_{\mathrm{x}[\cdot],\beta[\cdot]} \max_{k' \in \{0,\dots,K\}} \sum_{k=0}^{k'} H^*(t_k, \mathrm{x}[k], \beta[k])\Delta_k + g(t_{k'}, \mathrm{x}[k']), \tag{4.82}$$

$$\text{subject to } \begin{cases} \mathrm{x}[k+1] - \mathrm{x}[k] = -\Delta_k \beta[k], & k \in \{0,\dots,K-1\}, \\ \beta[k] \in \mathrm{co}(\{-f(t_k, \mathrm{x}[k], a) \mid a \in A\}), & k \in \{0,\dots,K-1\}, \\ \mathrm{x}[0] = x, & \\ c(t_k, \mathrm{x}[k]) \le 0, & k \in \{0,\dots,K\}. \end{cases} \tag{4.83}$$

Based on the above convexity analysis for the temporally discretized SCCIP in $(\mathrm{x}[\cdot], \beta[\cdot])$-space, it is sufficient to verify convexity conditions for $H^*(t,x,b)$ and $\{(x,b) \mid b \in \mathrm{co}(\{-f(t,x,a) \mid a \in A\})\}$ in (x,b). [65] provides sufficient convexity conditions for this, as written below.

Lemma 13. Suppose 1) $L(t,x,a) = L^x(t,x) + L^a(t,a)$, and L^x is convex in x for each $t \in [0,T]$, 2) $f(t,x,a) = M(t)x + L^a(t,a)$ for some matrix $M(t) \in \mathbb{R}^n \times \mathbb{R}^n$. Then, $H^*(t,x,b)$ and $\{(x,b) \mid b \in \mathrm{co}(\{-f(t,x,a) \mid a \in A\})\}$ are convex in (x,b).[65]

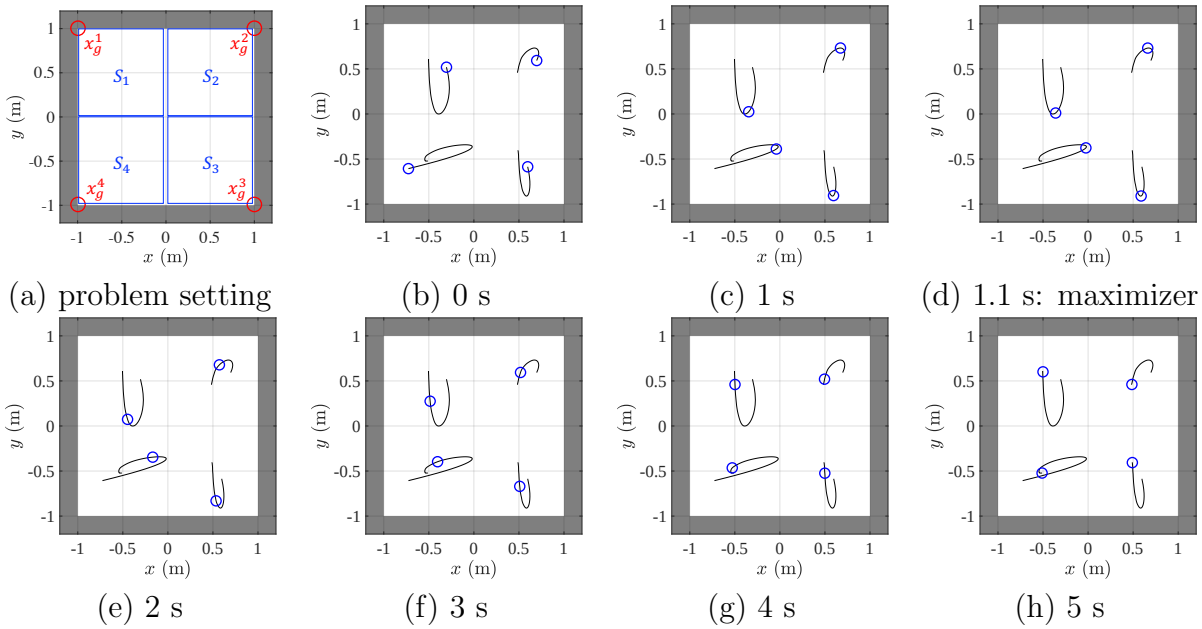
Combining Lemma 13 and the convexity analysis argument for SCCIP, we conclude sufficient convexity conditions for the temporally discretized Hopf-Lax formula for SCCIP in Assumption 3.

Assumption 3. (Convexity condition for the temporally discretized Hopf-Lax formula for SCCIP)

1. $L(t,x,a) = L^x(t,x) + L^a(t,a)$ for some L^x and L^a, and L^x is convex in x for all $t \in [0,T]$,
2. $g(t,x)$ is convex in x for all $t \in [0,T]$,
3. $c(t,x)$ is convex in x for all $t \in [0,T]$,
4. $f(t,x,a) = M(t)x + f^a(t,a)$ for some M and f^a for all $t \in [0,T]$.

Assumption 3 does not require 1) convex stage cost L in the control input a, 2) affine dynamics f in the control input a, and 3) convex control set A. This is because the Legendre-Fenchel transformation of Hamiltonian (H^*) is always convex in the control input a, and the convex-hull operator (co) also convexifies control-input space. We would like to note that the HJ analysis in Chapter 4.2.2 still needs to assume A is convex, as in Assumption 1.

Remark 10. The Hopf-Lax formula for SCCIP convexifies SCCIP in the control-input space.



(a) problem setting (b) 0 s (c) 1 s (d) 1.1 s: maximizer

(e) 2 s (f) 3 s (g) 4 s (h) 5 s

Figure 4.5: (a) shows a SCCIP problem setting that describes the target positions x_g^r and constrained regions S_r, $r = 1, ..., 4$. (b)-(h) show optimal trajectories for the SCCIP problem at each time, where the blue circles are the vehicles' positions, and the black curves are optimal state trajectories. The SCCIP cost is maximized at 1.1 s.

4.2.6 Numerical Example: robust formation control problem

This chapter provides one numerical example to demonstrate the Hopf-Lax formula for SCCIP. The numerical algorithm for SCGSP in Chapter 4.1 can be used for SCCIP.

Consider a sixteen-dimensional system which consists of four four-dimensional vehicles: $\mathrm{x}^r(s) = (\mathrm{x}_1^r(s), \mathrm{x}_2^r(s), \mathrm{x}_3^r(s), \mathrm{x}_4^r(s)) \in \mathbb{R}^4$ refers to the r-th vehicle's four-dimensional state $(r = 1, ..., 4)$ at time s, and $\mathrm{x}(s) = (\mathrm{x}^1(s), ..., \mathrm{x}^4(s)) \in \mathbb{R}^{16}$ refers to the system state at s. Each vehicle has two control inputs: $\alpha^r(s) = (\alpha_1^r(s), \alpha_2^r(s)) \in \mathbb{R}^2$. The system dynamics is

$$\begin{aligned} \dot{\mathrm{x}}_1^r(s) &= \mathrm{x}_2^r(s), \quad \dot{\mathrm{x}}_2^r(s) = \alpha_1^r(s)\cos\alpha_2^r(s), \\ \dot{\mathrm{x}}_3^r(s) &= \mathrm{x}_4^r(s), \quad \dot{\mathrm{x}}_4^r(s) = \alpha_1^r(s)\sin\alpha_2^r(s), \end{aligned} \tag{4.84}$$

where x_1^r and x_2^r (x_3^r and x_4^r) are horizontal (vertical) position and velocity of the r-th vehicle, α_1^r is the magnitude of acceleration, and α_2^r is the angle of the acceleration for $r = 1, ..., 4$.

As shown in Figure 4.5 (a), we would like to control the multi-vehicle system where the r-th vehicle stays close to the point (x_g^r) at the interval $s \in [0, 5]$ while maneuvering in the r-th squared-shaped constrained region (S_r). We control the accelerations of the system (4.84). Due to initial velocities, it is challenging to control the r-th vehicle to stay in S_r and

close to x_g^r. For this problem, we solve

$$\inf_{\alpha} \max_{\tau\in[0,5]} \max_{r=1,...,4} \|(\mathrm{x}_1^r(\tau), \mathrm{x}_3^r(\tau)) - x_g^r\|_2 \tag{4.85}$$

$$\text{subject to} \begin{cases} (4.84),\ \alpha^r(s) \in [-1,1] \times [-\frac{\pi}{6}, \frac{\pi}{6}], \\ \mathrm{x}(0) = x, \\ (\mathrm{x}_1^r(s), \mathrm{x}_3^r(s)) \in S_r, \quad r = 1,...,4, s \in [0,5], \end{cases} \tag{4.86}$$

where the target positions are following: $x_g^1 = (-1,1), x_g^2 = (1,1), x_g^3 = (1,-1), x_g^4 = (-1,-1)$. The stage cost L is zero, and the terminal cost g is $\max_{r=1,...,4}\|(\mathrm{x}_1^r(\tau), \mathrm{x}_3^r(\tau)) - x_g^r\|_2$.

For this SCCIP, the Hamiltonian H in (4.60) becomes

$$H(s,x,p) = \sum_{r=1}^{4} -p_1^r x_2^r - p_3^r x_4^r + \max\left\{ \|(p_2^r, p_4^r)\|_2, |p_2^r|\frac{\sqrt{3}}{2} + |p_4^r|\frac{1}{2} \right\}, \tag{4.87}$$

where x_i^r is the name of the variable for which we substitute $\mathrm{x}_i^r(s)$ $(r = 1,...,4)$, p_i^r is the costate with respect to x_i^r, $p^r = (p_1^r, p_2^r, p_3^r, p_4^r) \in \mathbb{R}^4$, and $p = (p^1, ..., p^4) \in \mathbb{R}^{16}$. H is convex in p, and any supporting hyperplane for H can be written as $b \cdot p = 0$ for some normal vector $b \in \mathbb{R}^{16}$. Since the supporting hyperplane $b \cdot p = 0$ passes through the origin for any b, the Legendre-Fenchel transformation of the Hamiltonian becomes

$$H^*(s,x,b) = \begin{cases} 0, & b_1^r = -x_2^r, b_3^r = -x_4^r, \|(b_2^r, b_4^r)\|_2 \le 1, |b_r^4| \le \frac{1}{2}, \\ \infty, & \text{otherwise}, \end{cases} \tag{4.88}$$

where $b = (b_1^1, b_2^1, ..., b_4^4) \in \mathbb{R}^{16}$, and $\mathrm{Dom}(H^*(s,x,\cdot))$ is analytically derived by (4.74). In general, if the stage cost L is zero, H^* becomes zero, which has been investigated in [65].

The Hopf-Lax formula for SCCIP in Theorem 8 is

$$\inf_{\beta} \max_{\tau\in[0,5]} \max_{r=1,...,4} \|(\mathrm{x}_1^r(\tau), \mathrm{x}_3^r(\tau)) - x_g^r\|_2 \tag{4.89}$$

$$\text{subject to} \begin{cases} \dot{\mathrm{x}}^r(s) = -\beta^r(s), \\ \beta_1^r(s) = -\mathrm{x}_2^r(s), \beta_3^r(s) = -\mathrm{x}_4^r(s), \\ \|(\beta_2^r(s), \beta_4^r(s))\|_2 \le 1, |\beta_4^r(s)| \le \frac{1}{2}, \\ \mathrm{x}(0) = x, \\ (\mathrm{x}_1^r(s), \mathrm{x}_3^r(s)) \in S_r, \quad r = 1,...,4, s \in [0,5], \end{cases} \tag{4.90}$$

where we get the second and third lines in (4.90) by substituting $\mathrm{x}(s)$ into x and $\beta(s) = (\beta_1^1(s), ..., \beta_4^4(s))$ into b in (4.88).

Consider temporally discretized SCCIP and Hopf-Lax formula for SCCIP on any temporal discretization $\{t_0 = 0, ..., t_K = 5\}$ as in Chapter 4.2.5. The temporally discretized SCCIP is non-convex, but the temporally discretized Hopf-Lax formula is convex since Assumption

3 is satisfied. Thus, gradient-based methods provide an optimal solution for the proposed Hopf-Lax formula for SCCIP.

For numerical computation of the Hopf-Lax formula, we discretize the temporal space to $\{t_0 = 0, ..., t_K = 5\}$ with $\Delta_k = 0.1$ (51 time steps). We utilize the interior-point method to solve the temporally discretized Hopf-Lax formula for SCCIP [19]. The computation time is 108.5 s. This system is 16 dimensional with four vehicles, for which it is intractable to utilize grid-based methods (such as the level-set method [85]) to solve the HJ PDEs (4.52).

As shown in Figure 4.5 (b)-(h), the four vehicles successfully stay in the constrained regions and stay near their target positions (x_g^r). Within the time interval, the maximum distance between x_g^r and the four vehicles is maximized at 1.1 s as shown in Figure 4.5 (d). At this time, each distance between x_g^r and the r-th vehicle is 1.18, 0.43 0.42, and 1.16 m, and the first vehicle shows the farthest distance among the four vehicles. Thus, the optimal cost for SCCIP is 1.18. This means that the r-th vehicle stays near x_g^r with maintaining less than 1.18 m distance within the time horizon while staying in S_r for all $r = 1, ..., 4$. The first vehicle's trajectory shown in Figure 4.5 (b)-(h) initially moves downward, with a velocity in the -y-direction. Thus, in the solution found by SCCIP, the optimal control first decreases its vertical speed to ensure that it remains in S_1. Then, it heads towards x_g^1, in order to stay close to x_g^1, as incentivized by the cost function.

This example shows an optimal-control analysis for SCCIP, and this can be additionally utilized in decision-making for hardware specification. For example, consider designing multiple mobile manipulators for which we attach a manipulator to each mobile robot (vehicle). Each manipulator aims to perform some tasks at its target position x_g^r for all time in $[0, 5]$. Our SCCIP analysis provides a guideline for choosing the manipulator's workspace that has to cover more than 1.18 m since all vehicles can be controlled to stay within 1.18 m-radius regions from x_g^r for all time in $[0, 5]$.

4.3 Hopf-Lax Formula for State-Constrained Reach-Avoid Problems

This chapter is based on the work presented in [68], which is joint work with Claire Tomlin.

In this chapter, we consider SCRAP for time-varying and time-invariant cases. In Chapter 3.2, we presented the HJ PDEs for SCRAP. Based on these HJ PDEs, we propose two Hopf-Lax formulae for SCRAP into two cases: time-varying and time-invariant ones.

4.3.1 State-Constrained Reach-Avoid Problems

This chapter recalls SCRAP and the corresponding HJ PDEs presented in Chapter 3.2.

In this chapter, we consider the state trajectory ($\mathrm{x} : [t, T] \to \mathbb{R}^n$) in a time interval $[t, T]$ solving the following ordinary differential equation (ODE):

$$\dot{\mathrm{x}}(s) = f(s, \mathrm{x}(s), \alpha(s)), s \in [t, T], \text{ and } \mathrm{x}(t) = x, \tag{4.91}$$

where (t, x) are the initial time and state, s is time between t and T, $f : [t, T] \times \mathbb{R}^n \times A \to \mathbb{R}^n$ is a dynamics function, $A \subset \mathbb{R}^m$ is the control set, $\alpha \in \mathcal{A}(t)$ is the control signal, and we denote the set of measurable control signals

$$\mathcal{A}(t) \coloneqq \{\alpha : [t, T] \to A \mid \|\alpha\|_{L^\infty(t,T)} < \infty\}. \tag{4.92}$$

We assume that A is a compact subset in $\mathbb{R}^m$.

State-constrained reach-avoid problem (SCRAP): for given initial time and state (t, x), solve

$$\vartheta_2(t, x) \coloneqq \inf_{\alpha\in\mathcal{A}} \min_{\tau\in[t,T]} \int_t^\tau L(s, \mathrm{x}(s), \alpha(s))ds + g(\tau, \mathrm{x}(\tau)) \tag{4.93}$$

$$\text{subject to} \quad c(s, \mathrm{x}(s)) \le 0, \quad s \in [t, \tau], \tag{4.94}$$

where x solves (4.91). The description for L, g, f, c is the same as SCCIP. Also, for each (t, x), $\vartheta_2(t, x) \in \mathbb{R} \cup \infty$. This means that if there is no feasible control signal to satisfy the state constraint (4.94), then $\vartheta_2(t, x)$ is the infinity.

For SCRAP, we also assume Assumption 1 holds, which guarantees the existence of the unique solution to the HJ PDEs, presented in Chapter 4.3.2.

4.3.2 Hamilton-Jacobi Equations for SCRAP

This chapter addresses the previously presented HJ PDEs for SCRAP [69], which will be utilized to derive Hopf-Lax formulae in Chapter 4.3.3. This chapter presents two HJ PDEs for the time-varying (general) and time-invariant cases. In the time-varying case, cost functions, dynamics, and state constraints are time-varying; in the time-invariant case, those are time-invariant.

Hamilton-Jacobi equation for SCRAP

Similar to Chapter 4.2, the epigraphical techinique is utilized. We first define a state-augmented value function V_2 (4.95) that combines the cost (4.93) and the constraint (4.94) of ϑ_2 so that V_2's sub-zero-level set is the epigraph of the optimal cost ϑ_2 for the SCRAP. For $(t, x, z) \in [0, T] \times \mathbb{R}^n \times \mathbb{R}$,

$$V_2(t, x, z) \coloneqq \inf_{\alpha\in\mathcal{A}(t)} \min_{\tau\in[t,T]} \max\Big\{ \max_{s\in[t,\tau]} c(s, \mathrm{x}(s)), \int_t^\tau L(s, \mathrm{x}(s), \alpha(s))ds + g(\tau, \mathrm{x}(\tau)) - z \Big\}, \tag{4.95}$$

where x solves (4.91), (t, x) are initial time and states, and z is again the new variable that represents a value axis for the epigraph of ϑ_2.

Theorem 11 presents HJ PDEs for V_2 and finds ϑ_2 from V_2.

Theorem 11 (HJ PDE for SCRAP)**.** Suppose Assumption 1 holds. V_2 in (4.95) is the unique viscosity solution to the HJ PDE:

$$\max\left\{c(t,x)-V_2(t,x,z), \min\left\{g(t,x)-z-V_2(t,x,z), \frac{\partial V_2}{\partial t}-\bar{H}(t,x,z,\frac{\partial V_2}{\partial x},\frac{\partial V_2}{\partial z})\right\}\right\}=0 \tag{4.96}$$

in $(0,T)\times\mathbb{R}^n\times\mathbb{R}$, where $\bar{H}:[0,T]\times\mathbb{R}^n\times\mathbb{R}\times\mathbb{R}^n\times\mathbb{R}\to\mathbb{R}$

$$\bar{H}(t,x,z,p,q) \coloneqq \max_{a\in A}[-p\cdot f(t,x,a)+qL(t,x,a)], \tag{4.97}$$

where p and q represent the gradients $\frac{\partial V_2}{\partial x}$ and $\frac{\partial V_2}{\partial z}$, and

$$V_2(T,x,z)=\max\{c(T,x), g(T,x)-z\} \tag{4.98}$$

on $\{t=T\}\times\mathbb{R}^n\times\mathbb{R}$.

For the time-invariant case, V_2 is also the unique viscosity solution to HJ PDE:

$$\max\left\{c(x)-V_2(t,x,z), \frac{\partial V_2}{\partial t}-\bar{H}_2^{\mathrm{TI}}(x,z,\frac{\partial V_2}{\partial x},\frac{\partial V_2}{\partial z})\right\}=0 \tag{4.99}$$

in $(0,T)\times\mathbb{R}^n\times\mathbb{R}$, where

$$\bar{H}_2^{\mathrm{TI}}(x,z,p,q)=\max\{0,\bar{H}(x,z,p,q)\} \tag{4.100}$$

for $(x,z,p,q)\in\mathbb{R}^n\times\mathbb{R}\times\mathbb{R}^n\times\mathbb{R}$, and

$$V_2(T,x,z)=\max\{c(x), g(x)-z\} \tag{4.101}$$

on $\{t=T\}\times\mathbb{R}^n\times\mathbb{R}$.

Then,

$$\vartheta_2(t,x)=\min z \text{ subject to } V_2(t,x,z)\le 0. \tag{4.102}$$

Our Hopf-Lax formulae in Chapter 4.3.3 will assume convex Hamiltonians in the costate space. $\bar{H}$ (3.41) and $\bar{H}_2^{\mathrm{TI}}$ (4.99) are convex in (p,q). Thus, Chapter 4.3.3 presents two Hopf-Lax formulae for time-varying SCRAP and time-invariant SCRAP.

4.3.3 Hopf-Lax Formulae for SCRAP

This chapter presents our main results that provide efficient computation: Hopf-Lax formulae for SCRAP. We first present the Hopf-Lax formula for the time-varying SCRAP, and then the Hopf-Lax formula for the time-invariant SCRAP.

Hopf-Lax formula for SCRAP

Define, for initial time and state $(t, x) \in [0, T] \times \mathbb{R}^n$,

$$\varphi_2(t,x) \coloneqq \inf_{\beta} \min_{\tau\in[t,T]} \int_t^\tau H^*(s, \mathrm{x}(s), \beta(s))ds + g(\tau, \mathrm{x}(\tau)), \tag{4.103}$$

$$\text{subject to } \begin{cases} \dot{\mathrm{x}}(s) = -\beta(s), s \in [t,T], \\ \beta(s) \in \mathrm{co}(\{-f(s, \mathrm{x}(s), a) \mid a \in A\}), s \in [t,T], \\ \mathrm{x}(t) = x, \\ c(s, \mathrm{x}(s)) \le 0, s \in [t,\tau], \end{cases} \tag{4.104}$$

where the description for all variables is the same as in Chapter 4.2.3.

Theorem 12. (Hopf-Lax formula for SCRAP) For all $(t, x) \in [0, T] \times \mathbb{R}^n$,

$$\vartheta_2(t,x) = \varphi_2(t,x), \tag{4.105}$$

where ϑ_2 is the optimal cost of SCRAP ((4.93) subject to (4.94)), and φ_2 is the Hopf-Lax formula for SCRAP ((4.103) subject to (4.104)).

Proof of the Hopf-Lax formula for the time-varying SCRAP

By applying the HJ analysis in Chapter 4.2.2, we have following results for the Hopf-Lax formula for SCRAP (φ_2 (4.103) subject to (4.104)). Combining (4.103) and (4.104), define a value function in the augmented state space, $W_2 : [0, T] \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$:

$$W_2(t,x,z) = \inf_{\beta} \min_{\tau\in[t,T]} \max\Big\{ \max_{s\in[t,\tau]} c(s, \mathrm{x}(s)), \int_t^\tau H^*(s, \mathrm{x}(s), \beta(s))ds + g(\tau, \mathrm{x}(\tau)) - z\Big\}, \tag{4.106}$$

where $\beta(s) \in \mathrm{co}(\{-f(s, \mathrm{x}(s), a) \mid a \in A\})$, and "co" is the convex-hull operator. As shown in [65],

$$\begin{aligned} \mathrm{Dom}(H^*(s, \mathrm{x}(s), \cdot)) &= \{b \mid H^*(s, \mathrm{x}(s), b) < \infty\} \\ &= \mathrm{co}(\{-f(s, \mathrm{x}(s), a) \mid a \in A\}). \end{aligned} \tag{4.107}$$

By Theorem 11, W_2 is the unique viscosity solution to

$$\max\Big\{c(t,x) - W_2(t,x,z), \min\{g(t,x) - z - W_2(t,x,z), \frac{\partial W_2}{\partial t} - \bar{H}_W(t,x,z, \frac{\partial W_2}{\partial x}, \frac{\partial W_2}{\partial z})\}\Big\} = 0 \tag{4.108}$$

in $(0, T) \times \mathbb{R}^n \times \mathbb{R}$, where $\bar{H}_W : [0, T] \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$

$$\bar{H}_W(t,x,z,p,q) \coloneqq \max_b [p \cdot b + qH^*(t,x,b)], \tag{4.109}$$

where p and q represent the gradients $\frac{\partial W_2}{\partial x}$ and $\frac{\partial W_2}{\partial z}$, H^* is defined in (4.61), and

$$W_2(T,x,z) = \max\{c(T,x), g(T,x) - z\} \tag{4.110}$$

on $\{t = T\} \times \mathbb{R}^n \times \mathbb{R}$. Then,

$$\varphi_2(t,x) = \min z \text{ subject to } W_2(t,x,z) \leq 0. \tag{4.111}$$

Now, it is sufficient to prove $V_2 \equiv W_2$.

Similar to Chapter 4.2, we utilize Theorem 10 to prove $V_2 \equiv W_2$. F_1 and F_2 in Chapter 4.2.4 refer to (4.96) and (4.108), respectively. Recall Lemma 12 as below.

Lemma 14. For $(t,x,z,p,q) \in [0,T] \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}$,

$$\bar{H}(t,x,z,p,q) = \bar{H}_W(t,x,z,p,q) \quad \text{ if } q \leq 0, \tag{4.112}$$

where $\bar{H}$ and $\bar{H}_W$ are defined in (4.97) and (4.109), respectively.

Proof of Theorem 12. Lemma 11 states that the sub-differentials and super-differentials of $V_2(t,x,z)$ with respect to z are less than or equal to 0 for all (t,x,z). Thus, by combining Theorem 10, Lemma 11 for SCRAP, and Lemma 14, we prove $V_2 \equiv W_2$. By (3.11) and (4.111), we conclude $\vartheta_2 \equiv \varphi_2$. $\square$

The proof of Theorem 12 is very similar to the above argument in this chapter.

Hopf-Lax formula for the time-invariant SCRAP

Define, for the initial time and state $(t,x) \in [0,T] \times \mathbb{R}^n$,

$$\varphi_2^{\mathrm{TI}}(t,x) \coloneqq \inf_{\beta} \int_t^T H_2^{\mathrm{TI}*}(\mathrm{x}(s), \beta(s))ds + g(\mathrm{x}(T)), \tag{4.113}$$

$$\text{subject to } \begin{cases} \dot{\mathrm{x}}(s) = -\beta(s), s \in [t,T], \\ \beta(s) \in \mathrm{co}(\{0\} \cup \{-f(\mathrm{x}(s),a) \mid a \in A\}), s \in [t,T], \\ \mathrm{x}(t) = x, \\ c(\mathrm{x}(s)) \leq 0, s \in [t,T], \end{cases} \tag{4.114}$$

where $\beta : [t,T] \to \mathbb{R}^n$ is a new measurable control signal, "co" is a convex-hull operator, $H_2^{\mathrm{TI}} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$

$$\begin{aligned} H_2^{\mathrm{TI}}(x,p) &\coloneqq \bar{H}_2^{\mathrm{TI}}(x,z,p,-1) \\ &= \max\{0, \max_{a \in A}[-p \cdot f(x,a) - L(x,a)]\}, \end{aligned} \tag{4.115}$$

where p is the costate with respect to x, $\bar{H}_2^{\mathrm{TI}}$ is defined in (4.100), and $H_2^{\mathrm{TI}*} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$

$$H_2^{\mathrm{TI}*}(x,b) \coloneqq \max_p [p \cdot b - H_2^{\mathrm{TI}}(x,p)], \tag{4.116}$$

where b is a new control input, and H_2^{TI*} is the Legendre-Fenchel transformation of H_2^{TI} with respect to p.

Theorem 13. (Hopf-Lax formula for the time-invariant SCRAP) Consider the time-invariant SCRAP. For all $(t,x) \in [0,T] \times \mathbb{R}^n$,

$$\vartheta_2(t,x) = \varphi_2(t,x) = \varphi_2^{\mathrm{TI}}(t,x), \tag{4.117}$$

where ϑ_2 is the optimal cost of SCRAP ((4.93) subject to (4.94)), φ_2 is the Hopf-Lax formula for SCRAP ((4.103) subject to (4.104)), and φ_2^{TI} is the Hopf-Lax formula for the time-invariant SCRAP ((4.113) subject to (4.114)).

Proof of the Hopf-Lax formula for the time-varying SCRAP is very similar to the one for the time-varying SCCIP.

Proof of the Hopf-Lax formula for the time-invariant SCRAP

By applying the HJ analysis in Chapter 4.3.3, we have following results for the Hopf-Lax formula for the time-invaraint SCRAP (φ_2^{TI} (4.113) subject to (4.114)). Combining (4.113) and (4.114), define a value function in the augmented state space. $W_2^{\mathrm{TI}} : [0,T] \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$:

$$W_2^{\mathrm{TI}}(t,x,z) = \inf_{\beta} \Big\{ \max_{s\in[t,T]} c(\mathrm{x}(s)), \int_t^T H_2^{\mathrm{TI}*}(\mathrm{x}(s),\beta(s))ds + g(\mathrm{x}(T)) - z \Big\}. \tag{4.118}$$

In (4.118), we can omit the control constraint in (4.114): $\beta(s) \in \mathrm{co}(\{0\} \cup \{-f(\mathrm{x}(s),a) \mid a \in A\})$, since this control constraint is the domain of the control input $\beta(s)$ for finite $H_2^{\mathrm{TI}*}(\mathrm{x}(s),\cdot)$. Note that "co" is a convex-hull operator.

Lemma 15 (Domain of $H_2^{\mathrm{TI}*}$)**.** For all $x \in \mathbb{R}^n$,

$$\begin{aligned} \mathrm{Dom}(H_2^{\mathrm{TI}*}(x,\cdot)) &= \{b \mid H_2^{\mathrm{TI}*}(x,b) < \infty\} \\ &= \mathrm{co}(\{0\} \cup \{-f(x,a) \mid a \in A\}), \end{aligned} \tag{4.119}$$

where $H_2^{\mathrm{TI}*}$ is defined in (4.116).

Proof. See Appendix E.1.

By Proposition 3.4 in [4], W_2^{TI} is the unique viscosity solution to

$$\max\Big\{c(x) - W_2^{\mathrm{TI}}, \frac{\partial W_2^{\mathrm{TI}}}{\partial t} - \bar{H}_W^{\mathrm{TI}}(x,z,\frac{\partial W_2^{\mathrm{TI}}}{\partial x}, \frac{\partial W_2^{\mathrm{TI}}}{\partial z})\Big\} = 0 \tag{4.120}$$

in $(0,T) \times \mathbb{R}^n \times \mathbb{R}$, where $\bar{H}_W^{\mathrm{TI}} : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$

$$\bar{H}_W^{\mathrm{TI}}(x,z,p,q) \coloneqq \max_b [p \cdot b + q H_2^{\mathrm{TI}*}(x,b)], \tag{4.121}$$

where $H_2^{\mathrm{TI}*}$ is defined in (4.116), and p and q represent the gradients $\frac{\partial W_2^{\mathrm{TI}}}{\partial x}$ and $\frac{\partial W_2^{\mathrm{TI}}}{\partial z}$. Then,

$$\varphi_2^{\mathrm{TI}}(t,x) = \min z \text{ subject to } W_2^{\mathrm{TI}}(t,x,z) \le 0. \tag{4.122}$$

Now, it is sufficient to show $V_2 \equiv W_2^{\mathrm{TI}}$. We will consider (4.99) as F_1 in Theorem 10 and (4.120) as F_2 in Theorem 10.

Lemma 16 analyzes the relationship between $\bar{H}_2^{\mathrm{TI}}$ (4.100) for V_2 and $\bar{H}_W^{\mathrm{TI}}$ (4.121) for W_2^{TI}.

Lemma 16. For $(x, z, p, q) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}$,

$$\bar{H}_2^{\mathrm{TI}}(x, z, p, q) = \bar{H}_W^{\mathrm{TI}}(x, z, p, q) \quad \text{if } q \leq 0, \tag{4.123}$$

where $\bar{H}_2^{\mathrm{TI}}$ and $\bar{H}_W^{\mathrm{TI}}$ are defined in (4.100) and (4.121), respectively.

Proof. See Appendix E.2.

Now, we are ready to conclude the proof of Theorem 13.

Proof of Theorem 13. Consider the HJ PDE for V_2 in (4.99) as F_1 in Theorem 10 and the HJ PDE for W_2^{TI} in (4.120) as F_2 in Theorem 10. Since Lemma 11 holds for $i = 2$ and 16 holds, Theorem 10 implies $V_2 \equiv W_2^{\mathrm{TI}}$. By Theorem 12, (4.102), and (4.122), we conclude $\vartheta_2 \equiv \varphi_2 \equiv \varphi_2^{\mathrm{TI}}$. □

4.3.4 Convexity analysis for SCRAP and its two Hopf-Lax formulae

This chapter provides convexity analysis for SCRAP and its Hopf-Lax formulae. This chapter utilizes the first-order forward Euler method.

We discretize the time interval to $\{t_0 = 0, ..., t_K = T\}$ where $\Delta_k \coloneqq t_{k+1} - t_k$. In the use of the first-order Euler method, discretization error for optimal control problems becomes smaller for smaller Δ_k [18, 3]. For notation, the state at t_k is $\mathrm{x}[k]$, and $\alpha[k]$ and $\beta[k]$ are control inputs at t_k. In this paper, we use x to denote both a state trajectory in the continuous-time setting ($\mathrm{x}(\cdot)$) and a state sequence in the discrete-time setting ($\mathrm{x}[\cdot]$). We apply the same notation rule for α, β.

This chapter presents convexity analysis for SCRAP, the Hopf-Lax formula for the time-varying SCRAP, and the Hopf-Lax formula for the time-invariant SCRAP. The temporally discretized SCRAP is

$$\vartheta_2(0, x) \simeq \min_{\mathrm{x}[\cdot], \alpha[\cdot], k' \in \{0,...,K\}} \sum_{k=0}^{k'} L(t_k, \mathrm{x}[k], \alpha[k])\Delta_k + g(t_{k'}, \mathrm{x}[k']), \tag{4.124}$$

$$\text{subject to } \begin{cases} \mathrm{x}[k+1] - \mathrm{x}[k] = \Delta_k f(t_k, \mathrm{x}[k], \alpha[k]), & k \in \{0, ..., K-1\}, \\ \alpha[k] \in A, & k \in \{0, ..., K-1\}, \\ \mathrm{x}[0] = x, & \\ c(t_k, \mathrm{x}[k]) \leq 0, & k \in \{0, ..., k'\}. \end{cases} \tag{4.125}$$

The cost in (4.124) is generally non-convex in $(\mathrm{x}[\cdot], \alpha[\cdot])$-space since the pointwise minimum operator over k' is defined in a non-convex set: $\{0, ..., K\}$. Thus, a sufficient convexity condition is that L and g are 0, as in Assumption 4. In this case, 0 is always a minimizer k'. Thus no additional conditions are necessary for convexity of the temporally discretized SCRAP.

Assumption 4. (Convexity condition for the temporally discretized SCRAP and its Hopf-Lax formula)

1. $L \equiv g \equiv 0$.

The temporally discretized Hopf-Lax formula for SCRAP ((4.103) subject to (4.104)) is as follows:

$$\varphi_2(0,x) \simeq \min_{\mathrm{x}[\cdot],\beta[\cdot],k' \in \{0,\dots,K\}} \sum_{k=0}^{k'} H^*(t_k, \mathrm{x}[k], \beta[k])\Delta_k + g(t_{k'}, \mathrm{x}[k']), \tag{4.126}$$

$$\text{subject to } \begin{cases} \mathrm{x}[k+1] - \mathrm{x}[k] = -\Delta_k \beta[k], & k \in \{0,\dots,K-1\}, \\ \beta[k] \in \mathrm{co}(\{-f(t_k, \mathrm{x}[k], a) \mid a \in A\}), & k \in \{0,\dots,K-1\}, \\ \mathrm{x}[0] = x, & \\ c(t_k, \mathrm{x}[k]) \le 0, & k \in \{0,\dots,k'\}. \end{cases} \tag{4.127}$$

By the same argument for SCRAP, H^* and g should be zero for the convexity of the discretized Hopf-Lax formula for SCRAP in $(\mathrm{x}[\cdot], \beta[\cdot])$-space. [65] proves that $L \equiv 0$ implies $H^* \equiv 0$, thus, Assumption 4 is also a sufficient condition for convexity of the temporally discretized Hopf-Lax formula for SCRAP.

We discretize the Hopf-Lax formula for the time-invariant SCRAP (φ_2^{TI} in (4.113) subject to (4.114)).

$$\varphi_2^{\mathrm{TI}}(0,x) \simeq \min_{\mathrm{x}[\cdot],\beta[\cdot]} \sum_{k=0}^{K} H_2^{\mathrm{TI}*}(\mathrm{x}[k], \beta[k])\Delta_k + g(\mathrm{x}(K)), \tag{4.128}$$

$$\text{subject to } \begin{cases} \mathrm{x}[k+1] - \mathrm{x}[k] = -\Delta_k \beta[k], & k \in \{0,\dots,K-1\}, \\ \beta[k] \in \mathrm{co}(\{0\} \cup \{-f(\mathrm{x}[k], a) \mid a \in A\}), & k \in \{0,\dots,K-1\}, \\ \mathrm{x}[0] = x, & \\ c(\mathrm{x}[k]) \le 0, & k \in \{0,\dots,K\}. \end{cases} \tag{4.129}$$

We will prove that Assumption 5 is sufficient for the temporally discretized time-invariant SCRAP to be convex in $(\mathrm{x}[\cdot], \beta[\cdot])$-space. We will first show $H_2^{\mathrm{TI}*}(x,b)$ and $\{(x,b) \mid b \in \mathrm{co}(\{0\} \cup \{-f(x,a) \mid a \in A\})\}$ are convex in (x,b). According to the definition of H_2^{TI} (4.115), H_2^{TI} does not have the state dependency since f and L do not have it. Thus, $H_2^{\mathrm{TI}*}$ also does not have the state dependency and is always convex in b. Similariy, $\{(x,b) \mid b \in \mathrm{co}(\{0\} \cup \{-f(x,a) \mid a \in A\})\}$ does not have the state dependency and is convex in b. Finally, the second and forth conditions in Assumption 5 implies the terminal cost and the state constraint are convex.

Assumption 5. (Convexity condition for the temporally discretized Hopf-Lax formula for the time-invariant SCRAP)

1. $L(x,a) = L(a)$,

2. $g(x)$ is convex in x,

3. $c(x)$ is convex in x,
4. $f(x,a) = f(a)$.

Remark 11. SCRAP and its Hopf-Lax formula for the time-varying case are generally non-convex. However, the Hopf-Lax formula for the time-invariant SCRAP is convex if Assumption 5 holds. Assumption 5 does not require any convexity conditions in the control-input space.

4.3.5 Numerical Algorithms

For the time-varying SCRAP, the numerical algorithm in Chapter 4.1 is similarly used. In this chapter, we present a numerical algorithm for the time-invariant SCRAP.

Optimal State and Control Trajectories

Consider feasible state and control trajectories x and β for the Hopf-Lax formula for the time-invariant SCRAP. Define a function, $\phi_\beta : [t,T] \to [t,T]$ with respect to β:

$$\phi_\beta(s) := t + \int_t^s \mathbb{1}\{\beta(\tau) \neq 0\} d\tau, \tag{4.130}$$

where

$$\mathbb{1}\{\beta(\tau) \neq 0\} := \begin{cases} 1, & \beta(\tau) \neq 0, \\ 0, & \beta(\tau) = 0, \end{cases} \tag{4.131}$$

and the corresponding inverse function:

$$\phi_\beta^{-1}(s) := \min \tau \text{ subject to } \phi_\beta(\tau) = s. \tag{4.132}$$

We consider state and control trajectories:

$$\beta_c(s) := \begin{cases} \beta(\phi_\beta^{-1}(s)), & s \in [t, \phi_\beta(T)] \\ 0, & s \in (\phi_\beta(T), T], \end{cases} \tag{4.133}$$

$$\dot{\mathrm{x}}_c(s) = -\beta_c(s), \quad s \in [t,T], \tag{4.134}$$

and $\mathrm{x}_c(t) = x$. β_c is generated from β such that β_c is bounded away from the zero vector almost everywhere in $[t, \phi_\beta(T)]$ and the zero vector almost everywhere in $(\phi_\beta(T), T]$. As illustrated in Figure 4.6, Lemma 17 states the relationship between x and x_c.

Lemma 17. For given initial time and state $(t,x) \in [0,T] \times \mathbb{R}^n$, any state trajectory $\mathrm{x} : [t,T] \to \mathbb{R}^n$ and control trajectory $\beta : [t,T] \to \mathbb{R}^n$ solving (4.104), x_c and β_c in (4.133) and (4.134) solve (4.104), and

$$\mathrm{x}_c(s) = \begin{cases} \mathrm{x}(\phi_\beta^{-1}(s)), & s \in [t, \phi_\beta(T)], \\ \mathrm{x}(T), & s \in (\phi_\beta(T), T]. \end{cases} \tag{4.135}$$

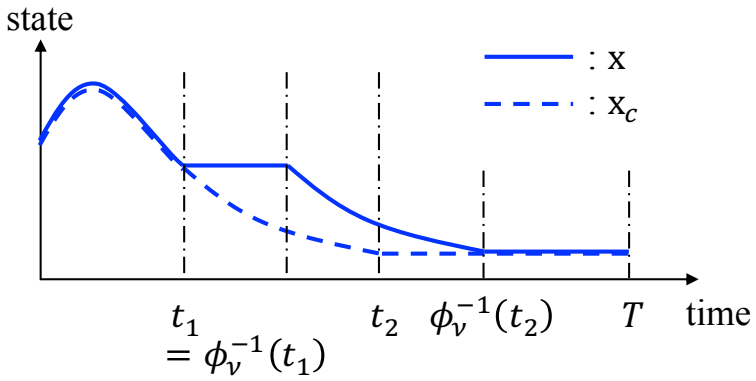


Figure 4.6: Illustration of Lemma 17. Note that $\phi_\beta(T) = t_2$ in this illustration.

Proof. For $s \in [t, \phi_\beta(T)]$,

$$\begin{aligned} \mathrm{x}_c(s) &= x - \int_t^s \beta(\phi_\beta^{-1}(\tau))d\tau = x - \int_t^{\phi_\beta^{-1}(s)} \beta(\tau)d\tau \\ &= \mathrm{x}(\phi_\beta^{-1}(s)). \end{aligned} \tag{4.136}$$

For $s \in (\phi_\beta(T), T]$, $\mathrm{x}_c(s) = \mathrm{x}_c(\phi_\beta(T)) = \mathrm{x}(T)$. Therefore, (4.135) is proved.

By (4.136) and (4.133), for $s \in [t, \phi_\beta(T)]$, $\mathrm{x}_c(s), \beta_c(s)$ satisfy (4.104). For $s \in (\phi_\beta(T), T]$, $\beta_c(s) = 0$ and (4.104) is satisfied. □

For any state trajectory x and control trajectory β solving (4.104), there exists $\gamma_i(s), \bar{\gamma}(s) \geq 0$ and $\alpha_i(s) \in \mathcal{A}$ such that

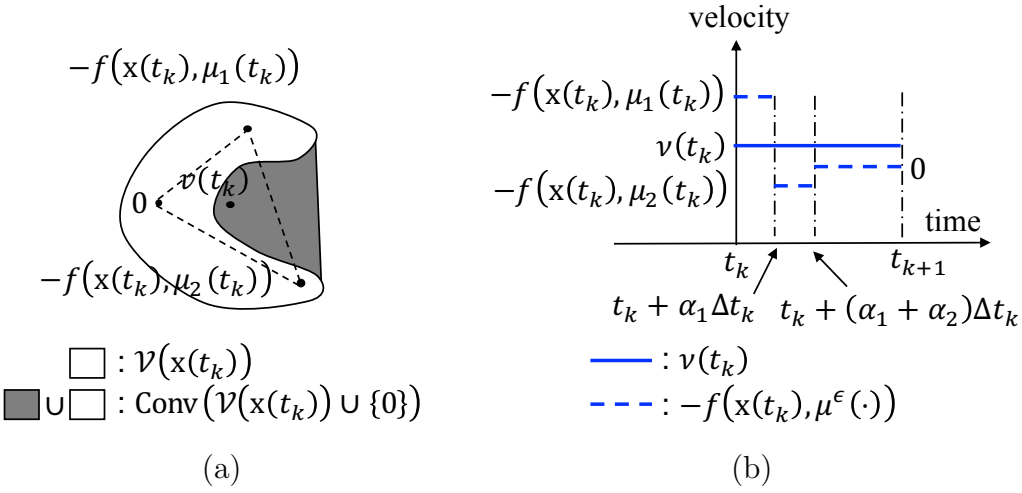
$$\beta(s) = -\sum_{i=1}^{\bar{i}(s)} \gamma_i(s) f(\mathrm{x}(s), \alpha_i(s)), \tag{4.137}$$

$$\sum_{i=1}^{\bar{i}(s)} \gamma_i(s) + \bar{\gamma}(s) = 1, \tag{4.138}$$

where $\bar{i}(s)$ is the number of $\gamma_i(s)$, since $\beta(s) \in \mathrm{co}(\{-f(\mathrm{x}(s), a) \mid a \in A\} \cup \{0\})$ for each $s \in [t, T]$. This analysis is illustrated in Figure 4.7.

Using (4.137) and (4.138), define state and control trajectories on a temporal discretization: $\{t_0 = t, ..., t_K = T\}$. We denote $\Delta t_k := t_{k+1} - t_k$. For each time interval $[t_k, t_{k+1})$, we can make finer discrete time interval $[t_{k,0} = t_k, ..., t_{k,i} = t_k + \sum_{j=1}^{i} \gamma_j(t_k)\Delta t_k, ..., t_{k,\bar{i}} = t_k + \sum_{j=1}^{\bar{i}(t_k)} \gamma_j(t_k)\Delta t_k, t_{k+1,0} = t_{k+1})$, where $\alpha_j(t_k)$ are defined in (4.137) and (4.138). Using the finer temporal discretization, define state and control trajectories:

$$\dot{\mathrm{x}}^\epsilon(s) = -\beta^\epsilon(s), \quad s \in [t, T], \quad \mathrm{x}^\epsilon(t) = x, \tag{4.139}$$



(a) (b)

Figure 4.7: Illustration of Lemma 18. In this example, $\beta(t_k) = -\gamma_1 f(\mathrm{x}(t_k),\mu_1(t_k)) - \gamma_2 f(\mathrm{x}(t_k),\mu_2(t_k))$, where $\mu_1(t_k),\mu_2(t_k) \in \mathcal{A}$ and $\gamma_1+\gamma_2<1$.

$$\beta^\epsilon(s) = \begin{cases} -f(\mathrm{x}^\epsilon(s),\alpha_i(t_k)), & s \in [t_{k,i-1},t_{k,i}), \\ 0, & s \in [t_{k,\bar{i}},t_{k+1}), \end{cases} \tag{4.140}$$

where $\alpha_i(t_k)$ is defined in (4.137). Note that β^ϵ is analogue to the velocity profile described in Figure 4.7 (b) by switching $\mathrm{x}(t_k)$ to $\mathrm{x}^\epsilon(s)$, $s \in [t_k,t_{k+1}]$. Then, Lemma 18 states that x^ϵ approximates x with ϵ-error bound.

Lemma 18. [65] For given initial time and state $(t,x) \in [0,T]\times\mathbb{R}^n$, consider any state trajectory x and control trajectory β solving (4.104). Suppose β is Riemann integrable in $[t,T]$. Then, for any $\epsilon>0$, there exists $\delta>0$ such that, for any discretization $\{t_0=t,...,t_K=T\}$ where $|\Delta t_k|<\delta$, $k=0,...,K-1$:

$$\|\mathrm{x}^\epsilon(s)-\mathrm{x}(s)\| \le \epsilon, \quad s\in[t,T]. \tag{4.141}$$

where x^ϵ solves (4.139).

Riemann integrability of β is a key assumption for Lemma 18 since β can be approximated by a piecewise constant function. As illustrated Figure 4.7 (b), the Riemann sum of $\beta(t_k)$ in $[t_k,t_{k+1}]$ is the same as the sum of Riemann sum of $-f(\mathrm{x}(t_k),\alpha_i(t_k))$ in $[t_{k,i-1},t_{k,i}]$ for all $i=1,...,\bar{i}$ by (4.137). By (4.137) and Lipschitz continuity of f, it can be proved that the norm of the difference between the Riemann sum of $\beta(t_k)$ and x^ϵ in (4.139) is bounded below by ϵ. The detail of this proof can be found in [65].

To combine Lemmas 17 and 18, with respect to state trajectory x and control trajectory β solving (4.104), define feasible state and control trajectories solving (4.91):

$$\dot{\mathrm{x}}_c^\epsilon(s) = f(\mathrm{x}_c^\epsilon(s),\alpha_c^\epsilon(s)),\ s\in[t,T], \quad \mathrm{x}_c^\epsilon(t)=x, \tag{4.142}$$

$$\alpha_c^\epsilon(s) := \begin{cases} \alpha_i(t_k), & s \in [\phi_{\beta^\epsilon}(t_{k,i-1}), \phi_{\beta^\epsilon}(t_{k,i})), \\ \text{any } a \in \mathcal{A}, & s \in [\phi_{\beta^\epsilon}(T), T], \end{cases} \tag{4.143}$$

where β^ϵ is defined in (4.140), ϕ_{β^ϵ} is defined in (4.130) for β^ϵ, and α_i is defined in (4.137). Then, we can approximate any feasible x solving (4.104) using x_c^ϵ.

Theorem 14. For given initial time and state $(t, x) \in [0, T] \times \mathbb{R}^n$, consider any state trajectory x and control trajectory β solving (4.104). Suppose Assumption 1 holds and β is Riemann integrable in $[t, T]$. Then, there exists $\delta > 0$ such that, for any discretization $\{t_0 = t, ..., t_K = T\}$ where $|\Delta t_k| < \delta$, $k = 0, ..., K-1$:

$$\|\mathrm{x}_c^\epsilon(\phi_{\beta^\epsilon}(s)) - \mathrm{x}(s)\| \leq \epsilon, \quad s \in [t, T], \tag{4.144}$$

where x_c^ϵ solves (4.142) and β^ϵ is defined in (4.140).

Proof. By Lemma 18, there exists $\delta > 0$ such that $|\Delta t_k| < \delta$ and $\|\mathrm{x}^\epsilon(s) - \mathrm{x}(s)\| \leq \epsilon$, where x^ϵ solves (4.139). By Lemma 17,

$$\mathrm{x}_c^\epsilon(\phi_{\beta^\epsilon}(s)) = \mathrm{x}^\epsilon(s), \quad s \in [t, T],$$

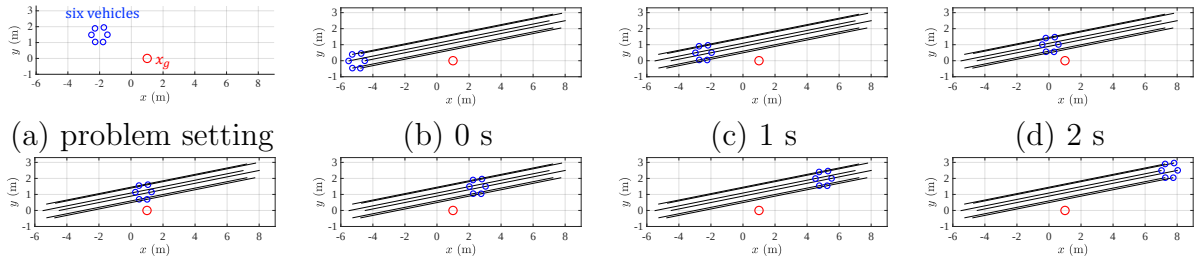
where x_c^ϵ solves (4.142) and β^ϵ is defined in (4.140). Therefore, we conclude (4.144). □

Algorithm 4 provides a numerical algorithm for the optimal state and control trajectories of the reach-avoid problem. We first utilize numerical optimization methods to compute optimal state and control trajectories for the proposed Hopf-Lax formula to determine $\beta[t_0], ..., \beta[t_K]$ over the temporal discretization $\{t_0 = t, ..., t_K = T\}$. For higher accuracy, smaller $\max_{k=0,...,K-1} \Delta t_k$ should be chosen. Using this, we numerically compute optimal state and control trajectories x_c^ϵ and α_c^ϵ in (4.142) and (4.143).

Algorithm 4 Numerical algorithm for the time-invariant SCRAP using its time-invariant Hopf-Lax formula

1: **Input:** the initial state, x; the initial time t; the terminal time T; temporal discretization: $\{t_0 = t, ..., t_K = T\}$
2: **Output:** the optimal control μ_*^ϵ, the optimal time s_*
3: solve x, β for the proposed Hopf-Lax formula using the temporal discretization
4: compute $\alpha_i(t_k)$ and $\gamma_i(t_k)$ in (4.137) and (4.138)
5: compute x^ϵ and β^ϵ in (4.139) and (4.140)
6: compute x_c^ϵ and α_c^ϵ in (4.142) and (4.143)

The computational complexity for the proposed Hopf-Lax formula using Algorithm 4 depends on the numerical optimization method to compute optimal state and control trajectories for the proposed Hopf-Lax formula. For example, the computational complexity using the interior-point method [19] is $\mathcal{O}(d^2 \log d)$, where d is the number of constraints. In many examples, d is proportional to the dimension of the state n. Our method significantly



(a) problem setting (b) 0 s (c) 1 s (d) 2 s

(e) 2.3 s: minimizer (f) 3 s (g) 4 s (h) 5 s

Figure 4.8: (a) shows a SCRAP problem setting that describes the target position x_g and the six vehicles' 2D positions. (b)-(h) show optimal trajectories for the SCRAP problem at each time, where the blue circles are the vehicles' positions, and the black curves are optimal state trajectories. The SCRAP cost is minimized at 2.3 s.

reduces the computational complexity where the Hopf-Lax formula for the reach-avoid problem provides a convex problem. Since the Hopf-Lax formula for the reach-avoid problem is analogous to the Hopf-Lax formula for the state-constrained optimal control problem in [65], the convexity conditions for the both formulae are the same: 1) g is convex, 2) f is linear in the state, and 3) c is convex in the state, which does not require any condition regarding the control. If the Hopf-Lax formula for the reach-avoid problem provides a non-convex problem, our method with gradient-based methods provides a sub-optimal solution to the reach-avoid problem.

4.3.6 Numerical Example

This chapter provides a numerical example to demonstrate the Hopf-Lax formulae for the time-invariant SCRAP.

Consider a twelve-dimensional system which consists of six two-dimensional vehicles: $\mathrm{x}^r(s) = (\mathrm{x}_1^r(s), \mathrm{x}_2^r(s)) \in \mathbb{R}^2$ refers to the r-th vehicle's two-dimensional state $(r = 1, ..., 6)$ at time s, and $\mathrm{x}(s) = (\mathrm{x}^1(s), ..., \mathrm{x}^6(s)) \in \mathbb{R}^{12}$. Each vehicle has two control inputs: $\alpha^r(s) = (\alpha_1^r(s), \alpha_2^r(s)) \in \mathbb{R}^2$. The system dynamics is

$$\dot{\mathrm{x}}_1^r(s) = \alpha_1^r(s) + 2, \quad \dot{\mathrm{x}}_2^r(s) = \alpha_2^r(s) + 1, \tag{4.145}$$

where x_1^r and x_2^r are horizontal and vertical positions of the r-th vehicle at s, and $\alpha_1^r(s)$ and $\alpha_2^r(s)$ are horizontal and vertical velocities.

As shown in Figure 4.8 (a), we would like to control the multi-vehicle system so that the distance between the 2D center position of the six vehicles and the goal point is minimized while maintaining the hexagonal formation with 0.1-error bound.

$$\inf_{\alpha} \min_{\tau \in [0,5]} \left\| \frac{\mathrm{x}^1(\tau) + ... + \mathrm{x}^6(\tau)}{6} - x_g \right\|_2 \tag{4.146}$$

$$\text{subject to} \begin{cases} (4.145), \|\alpha^r(s)\|_\infty \leq 0.5, r = 1, ..., 6, s \in [0, 5], \\ \mathrm{x}(0) = x, \\ \|\mathrm{x}^r(s) - \frac{\mathrm{x}^1(\tau)+...+\mathrm{x}^6(\tau)}{6} - d^r\|_\infty \leq 0.1, s \in [0, \tau], \end{cases} \tag{4.147}$$

where $x_g = (1, 0)$, and $d^r = 0.5(\cos((r-1)\pi/3), \sin((r-1)\pi/3))$ is a vector for the r-th vehicle that indicates the formation direction from the 2D center position of the six vehicles $(r = 1, ..., 6)$.

For this problem, we will derive the Hopf-Lax formula for the time-invariant SCRAP since the temporally discretized Hopf-Lax formula for the time-varying SCRAP as in Chapter 4.3.4 is generally non-convex.

The Hamiltonian $H_2^{\mathrm{TI}}(x, p)$ (4.115) is

$$H_2^{\mathrm{TI}}(x, p) = \max\{0, \sum_{r=1}^{6}[-2p_1^r - p_2^r + \|p_1^r\| + \|p_2^r\|]\}, \tag{4.148}$$

where $x = (x_1^1, x_2^1, ..., x_2^6) \in \mathbb{R}^{12}$, p_i^r is the costate with respect to x_i^r, $p^r = (p_1^r, p_2^r) \in \mathbb{R}^2$, and $p = (p^1, ..., p^6) \in \mathbb{R}^{12}$. Since H_2^{TI} is a pointwise maximum of two convex functions in p, H_2^{TI} is convex in p. Also, for all $b \in \mathbb{R}^{12}$, the supporting hyperplane of H_2^{TI} in p-space with respect to the normal vector b crosses the origin. Thus,

$$H_2^{\mathrm{TI}*}(x, b) = \begin{cases} 0, & \text{in } \mathrm{Dom}(H_2^{\mathrm{TI}*}(x, \cdot)), \\ \infty, & \text{otherwise}, \end{cases} \tag{4.149}$$

where $b = (b^1, ..., b^6) = (b_1^1, b_2^1, ..., b_1^6, b_2^6) \in \mathbb{R}^{12}$. By Lemma 15, $\mathrm{Dom}(H_2^{\mathrm{TI}*}(x, \cdot)) = \mathrm{co}(\{0\} \cup \{-f(x, a) \mid a \in [-0.5, 0.5]^{12}\}) \subset \mathbb{R}^{12}$ is analytically derived as follows.

$$\begin{aligned} \mathrm{Dom}(H_2^{\mathrm{TI}*}(x, \cdot)) =& \{b \mid \forall r_1, r_2 \in \{1, ..., 6\}, -2.5 \leq b_1^{r_1}, -1.5 \leq b_2^{r_1}, \frac{3}{5}b_1^{r_1} - b_1^{r_2} \geq 0, \\ & b_1^{r_1} - \frac{3}{5}b_1^{r_2} \leq 0, b_1^{r_1} \geq 5b_2^{r_2}, b_1^{r_1} \leq b_2^{r_2}, b_2^{r_2} \geq 3b_2^{r_1}, 3b_2^{r_2} \leq b_2^{r_1}\}. \end{aligned} \tag{4.150}$$

The Hopf-Lax formula for the time-invariant SCRAP in Theorem 13 is

$$\inf_\beta \left\| \frac{\mathrm{x}^1(5) + ... + \mathrm{x}^6(5)}{6} - x_g \right\|_2 \tag{4.151}$$

$$\text{subject to} \begin{cases} \dot{\mathrm{x}}^r(s) = -\beta^r(s), \\ \beta(s) \in \mathrm{Dom}(H_2^{\mathrm{TI}*}(x, \cdot)) \text{ in } (4.150), \\ \mathrm{x}(0) = x \\ \|\mathrm{x}^r(s) - \frac{\mathrm{x}^1(\tau)+...+\mathrm{x}^6(\tau)}{6} - d^r\|_\infty \leq 0.1, s \in [0, 5]. \end{cases} \tag{4.152}$$

Consider temporally discretized SCRAP and Hopf-Lax formula on any temporal discretization $\{t_0 = 0, ..., t_K = 5\}$ as in Chapter 4.3.4. The temporally discretized SCRAP is

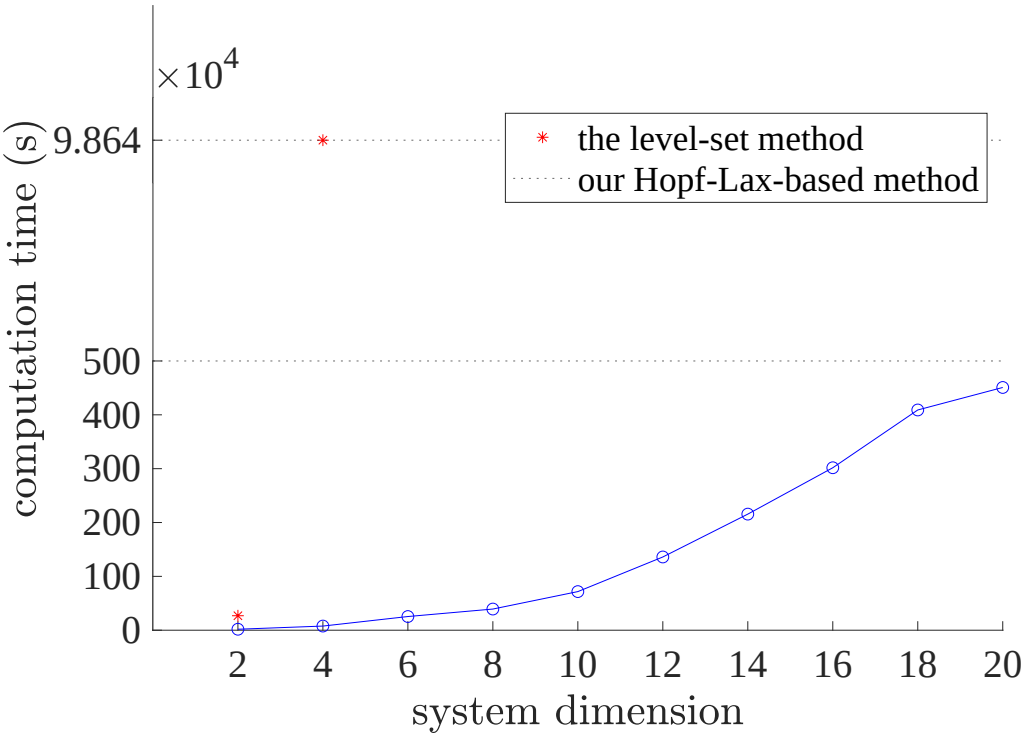


Figure 4.9: This figure shows the computation time of the time-invariant Hopf-Lax formula with different numbers of vehicles. Each vehicle is two-dimensional.

non-convex, but the temporally discretized Hopf-Lax formula is convex since Assumption 5 is satisfied, which allows the proposed Hopf-Lax formula for the time-invariant SCRAP to provide an optimal solution.

The computation time is 158.5 s. Figures 4.8 (b)-(h) show the optimal state (position) trajectories of the multi-vehicle system. The distance between the center of the six vehicles and the goal position x_g is minimized at $\tau = 2.3$ s, as shown in Figure 4.8 (e), and then the center of the six vehicles moves away from the goal position.

This example demonstrates our method's potential usefulness in multiple vehicle operations. For example, the control a spacecraft system to get close to an object of interest, such as an asteroid, and then the release of smaller exploration robots from the spacecraft to do more detailed sensing of the asteroid. The SCRAP analysis provides the optimal control and the time at which the exploration robots have to be released from the spacecraft.

Computation time

In this chapter, we compare the computation time of the Hopf-Lax formula for the time-invariant SCRAP and the level-set method. We can easily change the number of vehicles.

Figure 4.9 shows the computation time where the number of vehicles is 1 to 10. Since each vehicle is two-dimensional, the state dimension varies from 2 to 20.

The level-set method requires spatial and temporal discretization, which leads to exponential computational complexity in the state's dimension. The level-set method handles this problem if the number of vehicles is smaller than three. For numerical computation, we discretize the temporal space with 51 points and each axis of the augmented state (x, z) space with 61 points. The computation time is 26.9 seconds for the one-vehicle and 27.4 hours (9.86×10^4 seconds) for the two-vehicle.

The level-set method provides a closed-loop control. Thus, offline computation is allowed because a closed-loop control is robust to disturbances, measurement noise, and system modeling errors in practice. On the other hand, our Hopf-Lax methods provide an open-loop control, requiring real-time computation to be robust to the above factors.

Offline computation of the level-set method is intractable for high-dimensional systems due to the exponential growth of computational complexity in the state dimension and computing machines' memory limits. However, our Hopf-Lax method computes a solution even if the computation time might not meet real-time computation. For the three-vehicle setting (6D), the level set method requires 160 terabytes of memories for a numerical solution to the HJ PDE, where 51 temporal discretization points and 61 spatial discretization points on each axis of (x, z) are used. It still requires 367.4 gigabytes if the number of spatial discretization points on each axis is reduced by 21. On the other hand, our Hopf-Lax method computes a solution even for a ten-vehicle setting (20D). In order to reduce the computation time, the Hopf-Lax formula can be incorporated with approximation methods, such as the receding-horizon technique [39, 53] for more efficient computation.

4.4 Conclusion

This chapter proposed the Hopf-Lax formulae for SCGSP, SCCIP, and SCRAP. The formulae are the first to consider Hopf-Lax theory for HJ PDEs relevant to state-constrained problems. In addition, this chapter introduces two numerical algorithms: one for time-varying settings and the other for time-invariant settings. The Hopf-Lax formulae are posed in better convexity conditions than the problems for all three problems. Thus, we can guarantee the optimality for broader classes of problems.

Hopf-Lax theory allows us to deal with high-dimensional systems with efficient computation. This is because of the computational complexity of polynomial in the state dimension. However, Hopf-Lax theory provides open-loop control, although HJ analysis provides a closed-loop control. Therefore, real-time computation is necessary in the real world, where we do not have all the exact information, including states, dynamics, and environment. In order to achieve real-time computation, this method can also cooperate with approximation methods, including the receding horizon technique.

Our future directions are:

1. to find a method for an optimal control signal that has less chattering than our method;

2. to extend our theory for infinite-horizon optimal control problems;
3. to extend the theory in this chapter to hybrid systems;
4. to derive Hopf-Lax theory for zero-sum games.

Chapter 5

Reinforcement Learning for Reach-Avoid Problems

This chapter is based on the work presented in [70], which is joint work with Jingqi Li, Somayeh Sojoudi, and Claire Tomlin.

Ensuring the safety and performance of autonomous systems is essential for safety-critical systems, such as autonomous driving [81], surgical robots [102], and air traffic control [58]. Some safe control tasks could be modeled as driving the system's state to a target set in the state space while satisfying certain safety constraints [10]. This is referred to as the *reach-avoid* problem [100]. However, it is challenging to solve this problem under the presence of uncertainty [40], such as modeling errors and environmental disturbances. One way to accommodate this is to consider the reach-avoid problem under the worst-case scenario, which could be formulated as a zero-sum game between the control inputs and an adversary, accounting for the uncertainty or disturbance, with an objective to compromise the efforts of control inputs[103].

In the general case, this problem is challenging because it requires solving a zero-sum game with nonlinear dynamics, where the objective involves the worst-case performance instead of an average or cumulative performance over time, as in most constrained reinforcement learning (RL) papers [1, 35, 5]. Moreover, no future constraint violation is considered once the state trajectory enters the target set in the reach-avoid zero-sum game. Since constrained RL aims to learn a policy satisfying constraints at all times, it can only learn a conservative sub-optimal policy for the reach-avoid zero-sum game.

A well-known approach to solving the reach-avoid zero-sum game is the Hamilton-Jacobi (HJ) method [74, 46], in which one can design a value function such that the sign of the value function evaluated at a state encodes the safety and performance information of that state. In addition, reach-avoid games with multiple agents [92] and stochastic systems [104] have also been studied. These methods provide a closed-loop control policy for the continuous-time and finite-horizon zero-sum game setting, where an agent is required to reach a target set safely within a given finite-time horizon.

However, there are two major limitations of existing work on HJ-based methods for

finite-horizon reach-avoid games. Firstly, for complex or uncertain dynamical systems, it is difficult to predict a sufficient time horizon to guarantee the feasibility of the finite-time reach-avoid zero-sum game problem. This motivates the formulation of the infinite-horizon reach-avoid zero-sum game, where we remove the requirement of reaching the target set within a pre-specified finite horizon. The infinite-horizon reach-avoid game has been studied in only a few prior works. Among these is [57], in which the reach-avoid problem without an adversarial player is considered. We introduce the time-discount factor, a parameter discounting the impact of future reward and constraints, to the design of a contractive Bellman backup such that by annealing the time-discount factor to 1 they can obtain the reach-avoid set. Empirical results in [57] suggest the potential of this method in solving the infinite-horizon reach-avoid zero-sum games.

A second limitation is the computational complexity of classical HJ-based methods, which are exponential in the dimension of the state space [11]. Several approaches have been proposed in the literature to alleviate this. A line of work [75, 63, 96] formulate the reach-avoid zero-sum games as polynomial optimization problems for which there is no need to grid the entire state space. Another line of work on system decomposition, that is, decomposing a high-dimensional control problem into low-dimensional problems, is also a promising direction [27, 28]. Underapproximation of the reach-avoid set is proposed in [50] and an efficient open-loop policy method is developed in [110]. Nevertheless, these approaches presume certain structural assumptions about the dynamical systems and costs, which restrict their application.

With the power of neural networks, deep RL has been proven to be a promising technique for high-dimensional optimal control tasks [80, 95, 54], where a policy is derived to maximize the accumulated reward at each time step. It has been shown in [82] that neural networks could be leveraged to approximate the value function in high-dimensional optimal control problems and therefore alleviate the curse of dimensionality of Hamilton-Jacobi reachability analysis. For example, sinusoidal neural networks is proven to be a good functional approximator for learning the value function of the backward-reachable-tube problem for high-dimensional dynamical systems [9]. A deep reinforcement learning-based method is proposed in [45] to learn a neural network value function for *viability kernel*, the set of initial states from which a state trajectory could be maintained to satisfy pre-specified constraints. In particular, the paper [45] designs a contractive Bellman backup and learns a conservative approximation to the viability kernel. This method is further extended to solve the infinite-horizon reach-avoid problem in [57].

In this chapter, we propose a new HJ-based method for the infinite-horizon reach-avoid zero-sum game, and we develop a deep reinforcement learning method to alleviate the curse of dimensionality in solving it. Our contributions are threefold. We first propose a new value function for the reach-avoid zero-sum game, where the induced Bellman backup is a contraction mapping and ensures the convergence of value iteration to the unique fixed point. Subsequently, we analyze the designed value function by first proving its Lipschitz continuity, and then show that the new value function could be adapted to compute the viability kernel and the *backward reachable set*, that is, the set of states from which a state

trajectory could be controlled to reach the given target set. Finally, we alleviate the curse of dimensionality issue by proposing a deep RL algorithm, where we extend conservative Q-learning [62] to learn the value function of the infinite-horizon reach-avoid zero-sum game. We obtain upper and lower bounds for the learned value function. Our empirical results suggest that a (conservative) approximation of the reach-avoid set can be learned, even with neural network approximation.

Unlike the previous work [57] where the value function is discontinuous in general and the reach-avoid set can only be obtained when the time-discount factor is annealing to 1, the proposed value function is Lipschitz continuous under certain conditions and can exactly recover the reach-avoid set without annealing the time-discount factor. The obtained reach-avoid set is agnostic to any time-discount factors in the interval $[0, 1)$. In addition, this new value function could be adapted to compute the viability kernel and backward reachable set, which constitutes a unified theoretical perspective on reachability analysis concepts such as reach-avoid set, viability kernel, and backward reachable set.

Chapter 5.1 presents a formulation for the infinite-horizon reach avoid problem. Chapters 5.2 - 5.3 present our main theoretical results with proofs provided in the Appendix. Chapter 5.4 presents a deep reinforcement learning algorithm. Chapter 5.5 illustrates our algorithm in multiple experiments. Finally, Chapter 5.6 concludes this chapter.

We use an independent notation in this chapter. We denote the state of a system by $x \in \mathbb{R}^n$. $\mathbf{u} := \{u_0, u_1, \dots\}$ and $\mathbf{d} := \{d_0, d_1, \dots\}$ are control and disturbance sequences, where $u_t \in \mathcal{U} \subseteq \mathbb{R}^m$ and $d_t \in \mathcal{D} \subseteq \mathbb{R}^\ell$, respectively. We assume $\mathcal{U}$ and $\mathcal{D}$ to be compact sets. $\xi_x^{\mathbf{u},\mathbf{d}}(t)$ is the state trajectory evaluated at time t, which evolves according to the update rule

$$\begin{aligned} \xi_x^{\mathbf{u},\mathbf{d}}(0) &= x, \\ \xi_x^{\mathbf{u},\mathbf{d}}(t+1) &= f(\xi_x^{\mathbf{u},\mathbf{d}}(t), u_t, d_t),\ t \in \mathbb{Z}_+, \end{aligned} \tag{5.1}$$

where $\mathbb{Z}_+$ is the set of all non-negative integers and the system dynamics $f(\cdot,\cdot,\cdot) : \mathbb{R}^n \times \mathcal{U} \times \mathcal{D} \to \mathbb{R}^n$ is assumed to be Lipschitz continuous in the state.

5.1 Problem Formulation

Many control problems involving safety-critical systems can be interpreted as driving the system's state to a specific region while satisfying certain safety constraints. This intuition can be formalized by introducing the concept of *target set* as the set of desirable states to reach as well as the concept of *constraint set* as the set of all the states complying with the given constraints. The task is to design control inputs such that even under the worst-case disturbance the system state trajectory hits the target set while staying in the constraint set all times. We represent the target set and constraint set by $\mathcal{T} = \{x \in \mathbb{R}^n : r(x) > 0\}$ and $\mathcal{C} = \{x \in \mathbb{R}^n : c(x) > 0\}$, respectively, for some Lipschitz continuous and bounded functions $r(\cdot) : \mathbb{R}^n \to \mathbb{R}$ and $c(\cdot) : \mathbb{R}^n \to \mathbb{R}$.

At each time instance $t \in \mathbb{Z}_+$, the control input u_t aims to move the state towards the target set while staying inside the constraint set, whereas the disturbance d_t attempts to

drive the state away from the target set and the constraint set. In this work, we consider a conservative setting, where at each time instance the control input plays before the disturbance. We propose to account for this playing order by adopting the notion of *non-anticipative strategy*:

Definition 5.1.1 (Non-anticipative strategy)**.** A map $\phi : \{u_t\}_{t=0}^{\infty} \to \{d_t\}_{t=0}^{\infty}$, where $u_t \in \mathcal{U}$ and $d_t \in \mathcal{D}$, $\forall t \in \mathbb{Z}_+$, is a non-anticipative strategy if it satisfies the following condition: Let $\{u_t\}_{t=0}^{\infty}$ and $\{\bar{u}_t\}_{t=0}^{\infty}$ be two sequences of control inputs. Let $\{d_t\}_{t=0}^{\infty} = \phi(\{u_t\}_{t=0}^{\infty})$ and $\{\bar{d}_t\}_{t=0}^{\infty} = \phi(\{\bar{u}_t\}_{t=0}^{\infty})$. For all $T \geq 0$, if $u_t = \bar{u}_t$, $\forall t \in \{0, 1, \ldots, T\}$, then $d_t = \bar{d}_t$, $\forall t \in \{0, 1, \ldots, T\}$.

The intuition behind this non-anticipative strategy is that, given two sequences of control inputs sharing the same values for the first T entries, the corresponding entries of the two sequences of disturbance must also be the same, meaning the disturbance cannot use any information about the future control inputs when taking its own action. We denote by Φ the set of all non-anticipative strategies. Building upon the previous discussion on reach-avoid zero-sum games, we are now ready to define the definition the *reach-avoid set*:

$$\begin{aligned}\mathcal{RA}(\mathcal{T}, \mathcal{C}) :=& \{x \in \mathbb{R}^n : \forall \phi \in \Phi, \exists \{u_t\}_{t=0}^{\infty} \text{ and } T \geq 0, \\ & \text{s.t., } \forall t \in [0, T], \xi_x^{\mathbf{u}, \phi(\mathbf{u})}(T) \in \mathcal{T} \wedge \xi_x^{\mathbf{u}, \phi(\mathbf{u})}(t) \in \mathcal{C}\},\end{aligned}$$

where 's.t.' is the abbreviation of the phrase 'such that'. As such, we formalize the problem to be studied in this chapter below.

Problem 1 (Infinite-horizon reach-avoid game)**.** Let $\mathcal{T} := \{x \in R^n : r(x) > 0\}$ and $\mathcal{C} := \{x \in \mathbb{R}^n : c(x) > 0\}$ be the target and the constraint set, respectively. Find the reach-aviod set $\mathcal{RA}(\mathcal{T}, \mathcal{C})$ and an optimal sequence of control inputs $\{u_t\}_{t=0}^{\infty}$ for each state in $\mathcal{RA}(\mathcal{T}, \mathcal{C})$ such that under the worst-case disturbance $\{d_t\}_{t=0}^{\infty}$ the state trajectory will reach the target set without violating the constraints.

5.2 New Value Function for Infinite-horizon Reach-Avoid Games

In this chapter, we address Problem 1 by introducing a new Lipschitz continuous value function such that its sign evaluated at a state indicates whether that state can be driven to the target set without violating constraints. To be more specific, we will show that the *super-zero level set* of the designed value function recovers the reach-avoid set in Problem 1. Subsequently, we derive a Bellman backup equation. The fixed-point iteration based on this Bellman backup equation induces a unique fixed point solution, which will be proven to be the designed value function (5.3). Finally, we will show that the designed value function is Lipschitz continuous, which is a favorable property for HJ analysis [14].

Suppose that a state trajectory reaches a target set $\mathcal{T}$ at time t, i.e., $\xi_x^{\mathbf{u},\mathbf{d}}(t) \in \mathcal{T}$, while being maintained within the constraint set $\mathcal{C}$ for all $\tau \in \{0, 1, \ldots, t\}$ under an arbitrary sequence of disturbances $\mathbf{d} = \{d_t\}_{t=0}^{\infty}$. Let $\gamma \in [0, 1)$ be the time-discount factor, which discounts the impact of future reward and constraints. We observe that such a sequence of control inputs $\mathbf{u} = \{u_t\}_{t=0}^{\infty}$ should satisfy

$$\max_{u_0} \min_{d_0} \max_{u_1} \min_{d_1} \cdots \max_{u_t} \min_{d_t} \min \{\gamma^t r(\xi_x^{\mathbf{u},\mathbf{d}}(t)), \min_{\tau=0,\ldots,t} \gamma^\tau c(\xi_x^{\mathbf{u},\mathbf{d}}(\tau))\} > 0.$$

Therefore, to verify the existence of a sequence of control inputs driving the state trajectory to the target set safely from an initial state x_0, it suffices to check the sign of the following term:

$$\max_{u_0} \min_{d_0} \max_{u_1} \min_{d_1} \ldots \sup_{t=0,\ldots} \min \{\gamma^t r(\xi_x^{\mathbf{u},\mathbf{d}}(t)), \min_{\tau=0,\ldots,t} \gamma^\tau c(\xi_x^{\mathbf{u},\mathbf{d}}(\tau))\}. \tag{5.2}$$

By using the notion of the non-anticipative strategy ϕ, we can simplify (5.2) to the term

$$\inf_{\phi} \max_{\mathbf{u}} \sup_{t=0,\ldots,} \min \{\gamma^t r(\xi_x^{\mathbf{u},\phi(\mathbf{u})}(t)), \min_{\tau=0,\ldots,t} \gamma^\tau c(\xi_x^{\mathbf{u},\phi(\mathbf{u})}(\tau))\}.$$

This implies that a new value function for Problem 1 can be defined as

$$V(x) := \inf_{\phi} \max_{\mathbf{u}} \sup_{t=0,\ldots} \min \{\gamma^t r(\xi_x^{\mathbf{u},\phi(\mathbf{u})}(t)), \min_{\tau=0,\ldots,t} \gamma^\tau c(\xi_x^{\mathbf{u},\phi(\mathbf{u})}(\tau))\}, \ \forall x \in \mathbb{R}^n. \tag{5.3}$$

The sign of the value function encodes some crucial safety information about each state, that is, whether the state could be driven towards the target set while satisfying the given constraints. More precisely, $V(x) > 0$ if for every disturbance strategy ϕ, there exist $\mathbf{u} = \{u_t\}_{t=0}^{\infty}$ and $T < \infty$ such that $\xi_x^{\mathbf{u},\phi(\mathbf{u})}(T) \in \mathcal{T}$ and $\xi_x^{\mathbf{u},\phi(\mathbf{u})}(t) \in \mathcal{C}$ for all $t \in [0, T]$. Based on this intuition, we characterize the relationship between the super-zero level set of the value function (5.3) and the reach-avoid set next.

Theorem 15. Consider the value function $V(x)$ defined in (5.3). For every $\gamma \in [0, 1)$, it holds that $\{x \in \mathbb{R}^n : V(x) > 0\} = \mathcal{RA}(\mathcal{T}, \mathcal{C})$.

We propose to compute the value function (5.3) using dynamic programming. To this end, a contractive Bellman backup will be derived below.

Theorem 16. Let $\gamma \in [0, 1)$ be the time-discount factor. Suppose $U : \mathbb{R}^n \to \mathbb{R}$ is a bounded function. Consider the Bellman backup $B[\cdot]$ defined as,

$$B[U](x) \coloneqq \min \{c(x), \max\{r(x), \gamma \max_{u} \min_{d} U(f(x, u, d))\}\}. \tag{5.4}$$

Then, (5.3) is the unique solution to the Bellman backup equation, i.e., $V = B[V]$, and $\|B[V_1] - B[V_2]\|_\infty \leq \gamma \|V_1 - V_2\|_\infty$, for all bounded functions V_1 and V_2.

With the Bellman backup (5.4), we define the *value iteration* as the following recursion, starting from an arbitrary bounded function $V^{(0)}(\cdot) : \mathbb{R}^n \to \mathbb{R}$,

$$V^{(k+1)} := B[V^{(k)}],\ \forall k \in \mathbb{Z}_+. \tag{5.5}$$

Since the fixed-point iteration based method using a contraction mapping B will guarantee the convergence to a fixed point within that bounded function space, the Bellman backup (5.4) ensures the convergence of value iteration to (5.3), i.e., $\lim_{k\to\infty} V^{(k)} = V$.

Once we obtain the value function (5.3), we can calculate the following Q function:

$$Q(x, u, d) \coloneqq \min\{c(x), \max\{r(x), \gamma V(f(x, u, d))\}\}, \tag{5.6}$$

where $V(x) = \max_u \min_d Q(x, u, d)$. Given a state $x \in \mathbb{R}^n$, we can find an optimal control input and disturbance by $u^* = \max_u \min_d Q(x, u, d)$ and $d^* = \min_d Q(x, u, d)$, respectively. This provides a way to extract the optimal control and worst-case disturbance for Problem 1. Therefore, the proposed value function (5.3) constitutes a solution to Problem 1 in the sense that one can find the reach-avoid set, the optimal sequence of control inputs, and the worst-case sequence of disturbances.

In what follows, we characterize the Lipschitz continuity of the proposed value function (5.3). We show that the Lipschitz continuity of the value function (5.3) could be ensured under certain conditions on the dynamics and the time-discount factor.

Theorem 17 (Lipschitz continuity)**.** Suppose that the bounded functions $r(\cdot)$ and $c(\cdot)$ are L_r- and L_c-Lipschitz continuous, respectively. Assume also that the dynamics $f(x, u, d)$ is L_f-Lipschitz continuous in x, for all $u \in \mathcal{U}$ and $d \in \mathcal{D}$. Let $L := \max(L_r, L_c)$. Then, $V(x)$ is L-Lipschitz continuous if $L_f\gamma < 1$.

Remark 12. It is known that the sample complexity of neural network approximation could be improved if the function to be approximated is continuous [52]. Therefore, the Lipschitz continuity of the value funtion (5.3) is a beneficial property yielding reliable empirical performance when we deploy neural networks approximation to handle curse of dimensionality as highlighted in Chapter 5.5.2.

5.3 Computing the Viability Kernel and the Backward Reachable Set

In this chapter, we show that the designed value function (5.3) can be adapted to compute the viability kernel [45] and the backward reachable set [9]. We first present the formal definitions of these notions below.

We adopt the same definition of viability kernel as in [45], where the *viability kernel* is defined with respect to a closed set $\bar{\mathcal{C}} = \{x \in \mathbb{R}^n : c(x) \geq 0\}$, i.e., the closure of the constraint set $\mathcal{C}$, as

$$\Omega(\bar{\mathcal{C}}) := \{x \in \mathbb{R}^n : \forall \phi \in \Phi, \exists \mathbf{u}, \text{s.t.}, \xi_x^{\mathbf{u},\phi(\mathbf{u})}(t) \in \bar{\mathcal{C}}, \forall t \geq 0\},$$

which contains all states x from which a state trajectory $\xi_x^{\mathbf{u},\phi(\mathbf{u})}$ can be drawn such that $\xi_x^{\mathbf{u},\phi(\mathbf{u})}(t) \in \bar{\mathcal{C}}$, $\forall t > 0$. This set is of importance when the control task is to make the system satisfy some safety-critical constraints in real-world application [45].

The *backward-reachable set* [9] of a target set $\mathcal{T} = \{x \in \mathbb{R}^n : r(x) > 0\}$ is defined as

$$\mathcal{R}(\mathcal{T}) := \{x \in \mathbb{R}^n : \forall \phi \in \Phi, \exists \mathbf{u}, \exists t \geq 0, \text{s.t.}, \xi_x^{\mathbf{u},\phi(\mathbf{u})}(t) \in \mathcal{T}\},$$

which includes all states that can be driven towards the target set $\mathcal{T}$ in finite time. The backward reachable set is a useful concept in reachability analysis [9] because it specifies the set of states that can be controlled to reach a given target set. For instance, if an airplane is in the backward reachable set of another one, then it indicates that the two planes could collide with each other if they are not properly controlled.

In what follows, we show how the value function (5.3) can be adapted to compute the viability kernel and backward-reachable set.

Proposition 1. Assume that $r(x) = -1$ for all $x \in \mathbb{R}^n$. It holds that $V(x) \leq 0$ for all $x \in \mathbb{R}^n$. In addition, $V(x) = 0$ if and only if $x \in \Omega(\bar{\mathcal{C}})$.

Remark 13. The value function in Proposition 1 could serve as a control barrier function [6], from which we can derive a policy keeping the system states within a given set of the state space.

Similarly, we can compute the backward-reachable set by substituting particular constraint functions into (5.3).

Proposition 2. Assume that $c(x) = 1$ for all $x \in \mathbb{R}^n$. It holds that $V(x) \geq 0$ for all $x \in \mathbb{R}^n$. In addition, $V(x) > 0$ if and only if $x \in \mathcal{R}(\mathcal{T})$.

Given the above results, our method provides a new theoretical angle to computing the reach-avoid set, viability kernel, and backward reachable set for infinite-horizon games, where a common value function (5.3) can be adaptively used to compute multiple important sets for safety-critical analysis.

5.4 Deep Reinforcement Learning Algorithm

In this chapter, we develop a deep RL algorithm, which alleviates the curse of dimensionality issue for high dimensional problems. In particular, since the satisfaction of given safety constraints is vital for safety-critical systems, we propose to extend conservative Q-learning (CQL) [62] to Problem 1 and develop a deep RL algorithm for solving it, where the conservatism is favorable due to the neural network approximation error. More precisely, we learn a value function parameterized by a neural network which in theory is a lower bound of the value function (5.3). The super-zero level set of the converged value function in CQL is a subset of $\mathcal{RA}(\mathcal{T}, \mathcal{C})$, i.e., a conservative approximation of $\mathcal{RA}(\mathcal{T}, \mathcal{C})$.

The reason why CQL manages to learn a value function lower bound (5.3) is that CQL minimizes not only the Bellman backup error but also the value function itself at each iteration [62]. Since the super-zero level set of the value function (5.3) is the reach-avoid set $\mathcal{RA}(\mathcal{T}, \mathcal{C})$, the super-zero level set of a function that is a lower bound of (5.3) is a subset of $\mathcal{RA}(\mathcal{T}, \mathcal{C})$.

Similar to prior works on Deep RL [80, 95, 62], we approximate the Q function (5.6) by a neural network $Q_\theta(x, u, d) : \mathbb{R}^n \times \mathcal{U} \times \mathcal{D} \to \mathbb{R}$, where $\theta \in \mathbb{R}^M$ is the vector of parameters of the neural network and M is the total number of parameters. We define the neural network value function as $V_\theta(x) := \max_u \min_d Q_\theta(x, u, d)$, $\forall x \in \mathbb{R}^n$. We adopt the CQL framework [62] to learn the value function (5.3) by replacing the Bellman backup therein with (5.4). With $\lambda \geq 0$, we extend the loss function in [62] to the reach-avoid zero-sum game setting and propose the following loss function $L(\theta) : \theta \in \mathbb{R}^M \to \mathbb{R}$ for the neural network parameters:

$$L(\theta) := \mathbb{E}_{x\sim\mu}[\|V_\theta(x) - B[V_\theta](x)\|_2^2 + \lambda V_\theta(x)], \tag{5.7}$$

where μ is the uniform distribution over a compact set $\mathcal{X} \subseteq \mathbb{R}^n$ and the set $\mathcal{X}$ is subject to user's choice.

As suggested in [62], given a Bellman backup (5.4), its corresponding CQL Bellman backup $B_{CQL}[\cdot]$ can be derived as

$$B_{CQL}[U](x) := B[U](x) - \lambda, \tag{5.8}$$

where $U : \mathbb{R}^n \to \mathbb{R}$ is a bounded function and $B[\cdot]$ is the Bellman backup (5.4). One can show that this Bellman backup is a contraction mapping by a similar reasoning as in the proof of Theorem 16.

Given a bounded functional space, the fixed-point theorem [2] ensures that the value iteration based on (5.8) will converge to a unique value function, which we refer to as $V_{CQL}(x)$. We characterize the relationship between the nominal value function $V(x)$ in (5.3) and its conservative counterpart $V_{CQL}(x)$ below.

Theorem 18. Let $V(x)$ and $V_{CQL}(x)$ be the nominal value function defined in (5.3) and CQL value function, respectively. We have

$$V(x) - \frac{\lambda}{1-\gamma} \leq V_{CQL}(x) \leq V(x) - \lambda, \ \forall x \in \mathbb{R}^n. \tag{5.9}$$

Theorem 18 implies that the value function learned by CQL would be a lower bound of the true value function (5.3). However, the learned value function from CQL would still be bounded, and it is not too pessimistic, i.e., no state will have a value going to negative infinity. We remark here that the bounds in Theorem 18 are specialized for the reach-avoid zero-sum game.

Building on the above results, we propose Algorithm 5, where we substitute the Bellman backup in the algorithm of CQL [62] with (5.4). As in DQN [80], we adopt *experience replay* in Algorithm 5, where we store the state transition $\psi_t = (\xi_x^{\mathbf{u},\mathbf{d}}(t), u_t, d_t, \xi_x^{\mathbf{u},\mathbf{d}}(t+1))$

Algorithm 5 Conservative Reach-Avoid Deep Q-learning

1: Initialize the action-value function $Q_{\theta_0}(x, u, d)$ with random weights θ_0. Select a stepsize $\alpha > 0$, the total number of iterations K, the batch size J, and the rollout horizon T
2: **for** k = 0,. . . , K **do**
3: Uniformly sample $x \in \mathcal{X}$
4: Compute $\mathbf{u}$ and $\mathbf{d}$ such that $u_t \leftarrow \arg\max_u \min_d Q_{\theta_k}(\xi_x^{\mathbf{u},\mathbf{d}}(t), u, d)$ and $d_t \leftarrow \arg\min_d Q_{\theta_k}(\xi_x^{\mathbf{u},\mathbf{d}}(t), u_t, d)$
5: **for** t= 0,. . . , T **do**
6: Store $\psi_t = (\xi_x^{\mathbf{u},\mathbf{d}}(t), u_t, d_t, \xi_x^{\mathbf{u},\mathbf{d}}(t+1))$ in Ψ
7: **for** j=1,. . . ,J **do**
8: Sample a transition ψ_j in Ψ
9: Set $y_j \leftarrow \min\{c(\xi_x^{\mathbf{u},\mathbf{d}}(j)), \max\{r(\xi_x^{\mathbf{u},\mathbf{d}}(j)),\ \gamma \max_u \min_d Q_{\theta_k}(\xi_x^{\mathbf{u},\mathbf{d}}(j), u, d)\}\}$
10: $\theta_{k+1} \leftarrow \theta_k - \alpha\nabla_\theta\Big(\sum_{j=1}^{J}((y_j - Q_\theta(\xi_x^{\mathbf{u},\mathbf{d}}(j), u_j, d_j))^2 + \lambda Q_\theta(\xi_x^{\mathbf{u},\mathbf{d}}(j), u_j, d_j))\Big)$

at each time step in a replay memory data-set $\Psi = \{\psi_0, \psi_1, \dots\}$. Subsequently, we apply Q-learning updates to random samples from Ψ in step 10 of Algorithm 5. It is shown that applying Q-learning to random samples drawn from experience replay could break down the correlation between consecutive transitions and recall rare transitions, which leads to a better convergence performance [80].

5.5 Experiments

In this chapter, we present experiments on Algorithm 5 and show that our method could learn a (conservative) approximation to the reach-avoid set reliably even with neural network approximation.

5.5.1 2D Experiment for Reach-Avoid Game

In this experiment, we compare the reach-avoid set learned by Algorithm 5 with the one learned by tabular Q-learning, where we first grid the continuous state space and then run value iteration (5.4) over the grid. We treat the reach-avoid set learned by tabular Q-learning as the ground truth solution. We apply Algorithm 5 to learn neural network Q functions with 4 hidden layers, where each hidden layer has 128 neurons with ReLu activation functions. This neural network architecture is chosen because empirically it provides sufficient model capacity for approximating the true value function.

Let the target set and constraint set be $\mathcal{T} = \{(x, y) \in \mathbb{R}^2 : 1 - (x^2 + y^2) > 0\}$ and $\mathcal{C} = \{(x, y) \in \mathbb{R}^2 : 1 - \frac{(x-2)^2}{1.5^2} - y^2 > 0\}$, respectively. We visualize the target set $\mathcal{T}$ and the constraint set $\mathcal{C}$ in Figure 5.1. Consider the following discrete-time double integrator dynamics, with the time constant $\Delta t = 0.02s$:

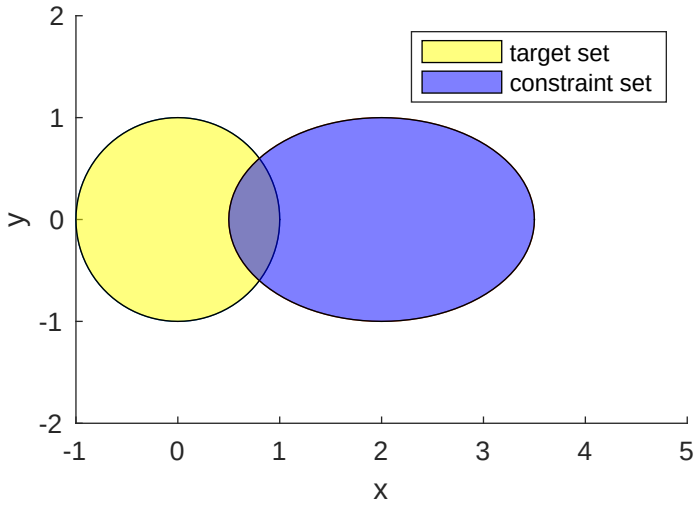


Figure 5.1: The target set and the constraint set in the 2D experiment

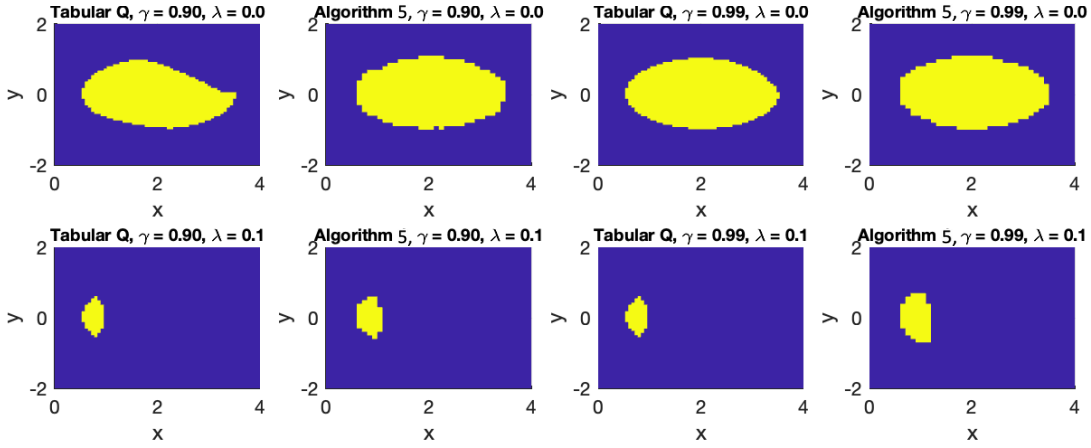


Figure 5.2: Comparison of the 2D reach-avoid set learned by tabular Q-learning and Algorithm 5. The yellow area corresponds to the reach-avoid set.

$$\begin{bmatrix} x(t+1) \\ y(t+1) \end{bmatrix} = \begin{bmatrix} x(t) + \Delta t y(t) \\ y(t) + \Delta t (u(t) + d(t)) \end{bmatrix} \tag{5.10}$$

where $x(t)$ and $y(t)$ are the position and velocity, respectively. The $u(t) \in \{-1, 1\}$ and $d(t) \in \{-0.5, 0.5\}$ are the control action and disturbance at time $t \in \{0, 1, 2, \dots\}$, respectively.

We see in Figure 5.2 that the reach-avoid set learned by Algorithm 5 is similar to the one computed by tabular Q-learning. An interesting observation is that, under different

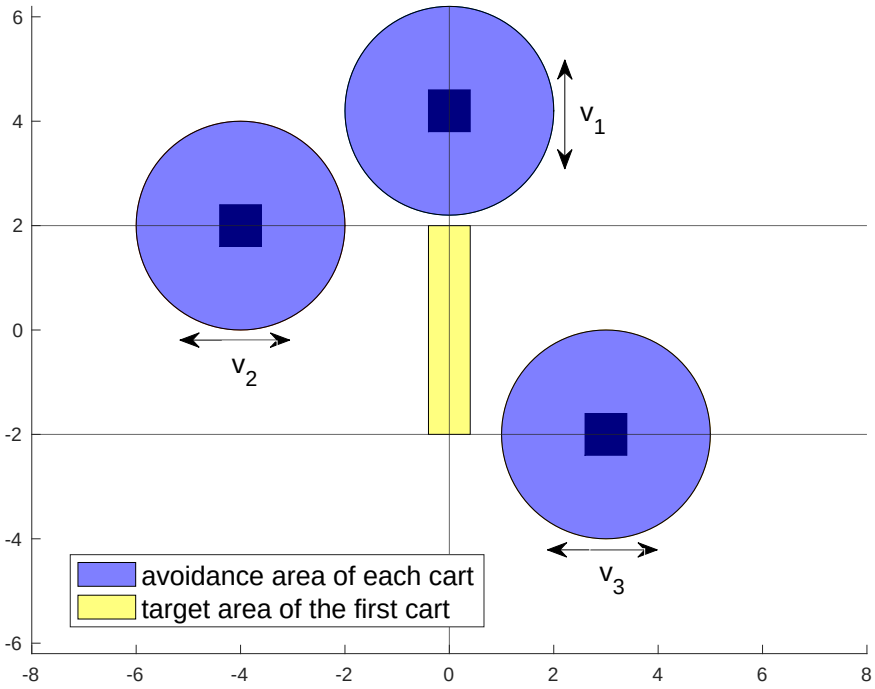


Figure 5.3: The avoidance area and the target area in the three-cart reach-avoid zero-sum game experiment

time-discount factors, the reach-avoid sets computed by tabular Q-learning are somewhat different. This could potentially be due to the numerical difficulty introduced by small values of γ. To be more specific, from the definition of the value function (5.3), we see that the powers of small constant $\gamma \in [0, 1)$ decays to 0 faster than those for a large parameter $\gamma \in [0, 1)$, and this makes it difficult to distinguish zero from the product of the power of a small constant γ and a bounded function term. Despite this numerical difficulty, it can be observed that both tabular Q-learning and Algorithm 5 learn similar reach-avoid sets under different time-discount factors $\gamma \in [0, 1)$.

In addition, as we increase the CQL penalty λ, the reach-avoid set shrinks to a conservative subset of the true reach-avoid set in both tabular Q-learning and Algorithm 5. For the same λ, although subject to certain numerical errors, both of these methods yield similar results, which empirically supports Theorem 18.

5.5.2 6D Experiment for Reach-Avoid Game

In this experiment, we consider a three-cart dynamical system, where the dynamics of each cart is modeled as a double integrator. The carts move along different axes. The first cart with the position $x_1(t)$ and velocity $v_1(t)$ moves vertically, while the second and the third carts move along the upper and the lower horizontal lines in Figure 5.3, respectively.

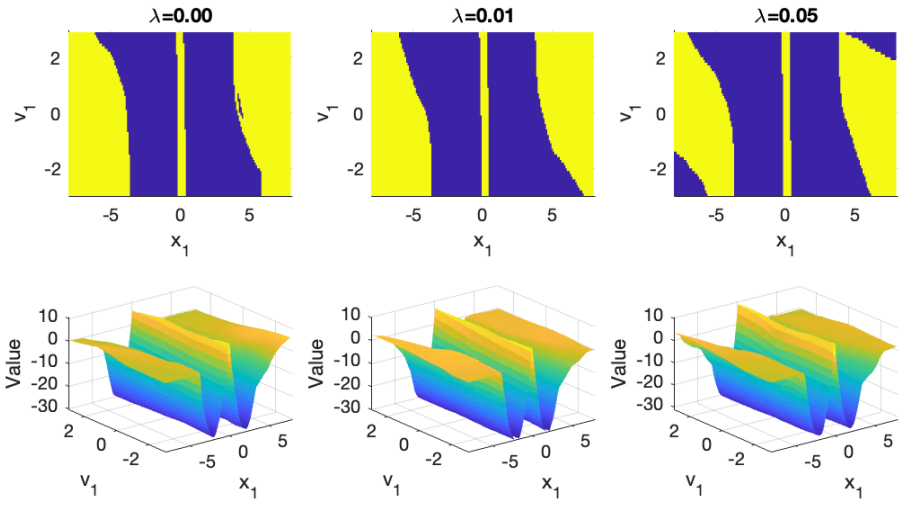


Figure 5.4: Visualization of reach-avoid sets under different CQL penalty weights λ, with $[x_2, v_2, x_3, v_3] = [-1, 1, 1, -1]$. In the first row, the yellow area represents the reach-avoid set. In the second row, we plot the value of each point in the corresponding plots in the first row.

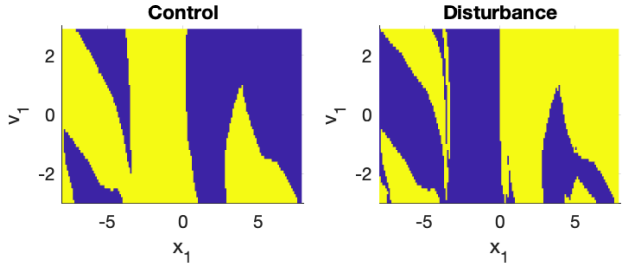


Figure 5.5: The control and disturbance policies extracted from the neural network value function learned by Algorithm 5 with $\lambda = 0.0$. The other four states are $[x_2, v_2, x_3, v_3] = [-1, 1, 1, -1]$. The yellow and blue areas in the left plot correspond to the control inputs with the values 1 and -1, respectively. The yellow and blue ares in the right plot correspond to the disturbances with the values 0.5 and -0.5, respectively.

The task is to drive the first cart towards the yellow region in Figure 5.3 while keeping the distance between every two cars at least 2. The target and the constraint sets can then be formulated as $\mathcal{T} = \{x \in \mathbb{R}^6 : 2 - |x_1| > 0\}$ and $\mathcal{C} = \{x \in \mathbb{R}^6 : \min\{(x_1 - 2)^2 + x_2^2, (x_1 +$

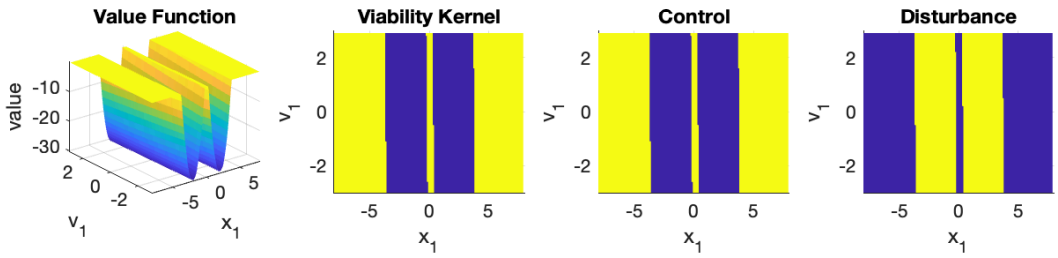


Figure 5.6: Visualization of the learned viability kernel, with $[x_2, v_2, x_3, v_3] = [1, 1, -1, 1]$. In the second subplot, the yellow area is the viability kernel. In the third subplot, the yellow and blue areas correspond to the control inputs 1 and -1, respectively. In the fourth subplot, the yellow and blue areas correspond to the disturbances 0.5 and -0.5, respectively.

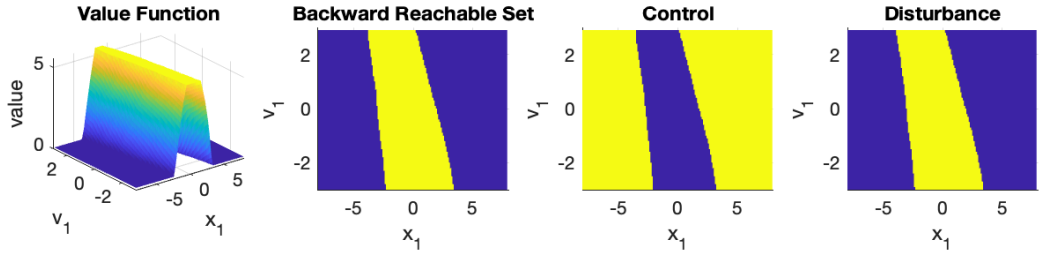


Figure 5.7: Visualization of the learned backward reachable set, with $[x_2, v_2, x_3, v_3] = [0.6, 0.0, 0.7, 0.1]$. In the second subplot, the yellow area is the backward reachable set. In the third subplot, the yellow and blue areas correspond to the control actions 1 and -1, respectively. In the fourth subplot, the yellow and blue areas correspond to the disturbance action 0.5 and -0.5, respectively.

$2)^2 + x_3^2\} - 4 > 0\}$, respectively. The dynamics is

$$\begin{bmatrix} x_1(t+1) \\ v_1(t+1) \\ x_2(t+1) \\ v_2(t+1) \\ x_3(t+1) \\ v_3(t+1) \end{bmatrix} = \begin{bmatrix} x_1(t) + \Delta t v_1(t) \\ v_1(t) + \Delta t(u_t + d_t) \\ x_2(t) + \Delta t v_2(t) \\ v_2(t) + 0.02\Delta t \\ x_3(t) + \Delta t v_3(t) \\ v_3(t) + 0.02\Delta t \end{bmatrix} \tag{5.11}$$

where $u_t \in \{-1, 1\}$ and $d_t \in \{-0.5, 0.5\}$ are the control input and disturbance, respectively. The time-integration constant Δt is equal to $0.02s$. The time-discount factor γ is 0.99.

Due to the curse of dimensionality, tabular Q learning explained in the previous experiment suffers numerical difficulties in this 6-dimensional experiment. In this subchapter, we apply Algorithm 5 with the same neural network architecture as in Chapter 5.5.1. We plot the learned reach-avoid set by projecting it onto a 2-dimensional plane. In Figure 5.4, we visualize the reach-avoid set as well as the effect of the CQL penalty λ on the learned reach-avoid set.

As the penalty λ increases, the volume of the reach-avoid set shrinks while the empirical success rate improves. This suggests that a larger penalty λ induces a more conservative estimation of the reach-avoid set and the policy.

We sample 1000 initial states in each of the three learned reach-avoid set in Figure 5.4, and then collect the trajectories with maximum length 1000 time steps, under the control and disturbance policies induced by the learned neural network value function. For the three learned value functions, the portion of those 1000 trajectories under each of the three values of λ that successfully reach the target set without violating the constraints are 85.4%, 88.4%, and 91.3%. In addition, we randomly sample 10,000 points in the state space, and observe that the ratios of points lying in the learned reach-avoid sets for $\lambda = 0.01$ and $\lambda = 0.05$ to those for $\lambda = 0.0$ are 99.7% and 82.5%, respectively. However, note that the training epochs taken to learn the three value functions are 1290, 1650, and 1470, respectively.

In Figure 5.5, we visualize the control and disturbance policies extracted from the Q function of the neural network corresponding to $\lambda = 0.0$ in Figure 5.4. The two policies are considered as to be reasonable because at most state, the control policy either drives the agent towards the target set or prevent the violation of the constraint. Notice that there are a few states, e.g., some points at the right lower corner of the disturbance law in Figure 5.5, where they do not complement to each other. One possible reason is that we learn a local optimal neural network Q function by minimizing the non-convex loss function (5.7).

5.5.3 6D Experiments on Learning Viability Kernel and Backward Reachable Set

In this subchapter, we apply Algorithm 5 to learn viability kernel and backward reachable set for the 6-dimensional dynamical system in Subchapter 5.5.2. The results empirically confirm Propositions 1 and 2. In the following two experiments, the same neural network architecture as in Chapter 5.5.2 does not yield satisfactory results and we conjecture that this is due to the limited model capacity. As such, in this subchapter, we increase the model capacity by adopting 4-layer neural networks with 256 neurons in each layer. We set the CQL penalty parameter to be $\lambda = 0.0$.

We first consider learning the viability kernel where the constraint set is the same as the one in Subchapter 5.5.2. The reward function is set to be $r(x) = -1$, for all $x \in \mathbb{R}^n$. In the first subplot of Figure 5.6, we visualize the learned value function. The value function is non-positive, which is predicted by Proposition 2. We visualize the learned viability kernel in the second subplot of Figure 5.6. We sample 1000 initial positions in the learned viability kernel and simulate a trajectory for each of them with 600 steps. The portion of those sampled points that can be maintained inside the constraint set is 79.2%.

Subsequently, we consider learning a backward reachable set by leveraging Proposition 2, where the constraint function is set to be a constant function with the value 1. We consider the target set $\mathcal{T} = \{x \in \mathbb{R}^6 : \min(2 - |x_1|, 1 - |x_2|, 1 - |x_3|) > 0\}$. We visualize the learned value function and backward reachable set in Figure 5.7. As shown in Proposition 2, the value

function is non-negative. We sample 1000 points in the obtained backward reachable set. The portion of initial states that can be driven towards the target set within 600 simulation steps is 70.0%.

It is observed that the empirical success rate of viability kernel is higher than backward reachable set. There are multiple reasons behind it. First, the simulation horizon is not long enough for an initial state to be driven towards the backward reachable set. In addition, for backward reachable set, a single failure in control may push the trajectory away from the target set in the future. However, for the viability kernel, multiple failures in control are allowed if controls at the boundary of the viability kernel keep the trajectory inside the constraint set.

We remark here that the high-dimensional experiments are challenging because there is a lack of efficient methods to check the sufficiency of neural network model capacity and to solve the non-convex optimization problem (5.7), which are common problems in deep RL and on-going research directions in the deep RL community [44].

5.6 Conclusion

In this chapter, we investigate the infinite-horizon reach-avoid zero-sum game problem, in which the goal is to learn the reach-avoid set in the state space and an associated policy such that each state in the reach-avoid set could be driven towards a given target set while satisfying constraints. We design a value function that offers several properties: 1) its super-zero level set coincides with the reach-avoid set and the induced Bellman Backup equation is a contraction mapping; 2) the value function is Lipschitz continuous under certain conditions; and 3) the value function can be adapted to compute the viability kernel and backward reachable set. We propose to alleviate the curse of dimensionality issue by developing a deep RL algorithm. The provided theoretical and empirical results suggest that our method is able to learn reach-avoid sets reliably even with neural network approximation errors.

Chapter 6

Conclusion

This dissertation presents state-constrained reachability problems and provides three approaches to solve them: HJ analysis, Hopf-Lax theory, and reinforcement learning. The dissertation demonstrates the impact of the state-constrained reachability problems in various applications. In this dissertation, we focus on optimal control problems that are meaningful in the real-world. For example, the state-constrained control-invariance problem provides system design parameters, and the state-constrained reach-avoid problem provides the closest distance between the system and the goal, both of which have use in real-world applications. Aligned with this direction, for future work, I aim to provide a methodology that formulates a correct optimal control problem or zero-sum game for other problems.

Regarding the theoretical contribution, the three approaches contribute to the evolution of reachability theory by considering state-constrained problems that have not been considered in most previous work. The proposed HJ analysis uses the epigraphical technique to deal with general state-constrained reachability problems, and is the first analysis to do so within the reachability community. The proposed Hopf-Lax theory first deals with HJ PDEs relevant to state-constrained problems. The proposed reinforcement learning framework is applicable for general optimal control or zero-sum game problems for which the dynamic programming principle works.

Regarding theoretical guarantees, HJ analysis generally guarantees safety and performance for nonlinear systems and non-convex state constraints. Hopf-Lax theory guarantees safety and performance under sufficient conditions that do not require convexity in the control-input space. On the other hand, the proposed reinforcement learning framework does not guarantee safety and performance. Thus the reinforcement learning chapter only provides empirical analysis.

Regarding computational efficiency, it is intractable to compute solutions to HJ PDEs for high-dimensional systems numerically. On the other hand, the proposed Hopf-Lax theory allows for efficient computation for a certain class of Hamilton-Jacobi equations. It gives an open-loop control, so the real-time computation is necessary. Approximation methods, such as the receding horizon technique, can be used here. While the reinforcement learning framework introduced in the third part of this dissertation is less computationally efficient

than the proposed Hopf-Lax theory, it provides a closed-loop control and thus can be used offline. Overall, these two numerically efficient approaches provides approximations to help alleviate the curse of dimensionality.

In the future, I am interested in developing efficient real-time methods for Hopf-Lax theory and reinforcement learning techniques, and to pursue theoretical guarantees for these methods.

Appendix A

Proof for Section 3.1

A.1 Proof of Lemma 1

Proof.
(i) $\vartheta_1^+(t,x) - z \le 0 \Rightarrow V_1^+(t,x,z) \le 0$

$\vartheta_1^+(t,x) - z \le 0$ implies that, for all $\eta \in H(t)$, there exists $\alpha \in \mathcal{A}(t)$ such that

$$\max_{\tau\in[t,T]} \int_t^\tau L(s, \mathrm{x}(s), \alpha(s), \eta[\alpha](s))ds + g(\tau, \mathrm{x}(\tau)) - z \le \epsilon \tag{A.1}$$

and $c(s, \mathrm{x}(s)) \le 0$ for $s \in [t,\tau]$ for any small $\epsilon > 0$, where x solves (2.1) for $(\alpha, \eta[\alpha])$. Thus, for all η, there exists α such that $J_1(t,x,z,\alpha,\eta) \le \epsilon$, which concludes $V_1^+(t,x,z) \le 0$. Note that J_1 is defined in (3.7).

(ii) $V_1^+(t,x,z) \le 0 \Rightarrow \vartheta_1^+(t,x) - z \le 0$

Assumption 1 implies that, for any $\eta \in H(t)$, there exists $\alpha \in \mathcal{A}$ such that $J_1(t,x,z,\alpha,\eta[\alpha]) \le V_1^+(t,x,z)$. If $V_1^+(t,x,z) \le 0$, for any $\eta \in H(t)$, there exists α such that $\max_{s\in[t,\tau]} c(s,\mathrm{x}(s)) \le 0$ and $\int_t^\tau L(s,\mathrm{x}(s),\alpha(s),\eta[\alpha](s))ds + g(\tau,\mathrm{x}(s)) - z \le 0$ for all $\tau \in [t,T]$. Thus, $\vartheta_1^+(t,x) - z \le 0$.

(i) and (ii) concludes (3.11).

(iii) Let $\tilde{\vartheta}_1^+$ be the right hand term in (3.12) subject to (3.13). Then, the following statement can be proved by analogous proofs in (i) and (ii).

$$\begin{aligned}\tilde{\vartheta}_1^+(t,x) = \min z \text{ subject to } \sup_\eta \inf_\alpha \max\{ &\max_{s\in[t,T]} c(s,\mathrm{x}(s)), \\ &\max_{\tau\in[t,T]} g(\tau,\mathrm{x}(\tau)) - \mathrm{z}(\tau)\} \le 0.\end{aligned} \tag{A.2}$$

By (3.10) and (3.11), we conclude $\vartheta_1^+(t,x) = \tilde{\vartheta}_1^+(t,x)$.

(iv) The proof for V_1^- and ϑ_1^- is similar to that for V_1+ and ϑ_1^+. □

A.2 Proof of Lemma 2

Proof. Consider (x, z) solving (3.9) for any (α, δ), and a small $h > 0$. (3.10) implies

$$J_1(t,x,z,\alpha,\delta) = \max\Big\{ \max_{s\in[t,t+h]} c(s,\mathrm{x}(s)),\ \max_{s\in[t,t+h]} g(s,\mathrm{x}(s)) - \mathrm{z}(s), \\ \max\{ \max_{s\in[t+h,T]} c(s,\mathrm{x}(s)),\ \max_{s\in[t+h,T]} g(s,\mathrm{x}(s)) - \mathrm{z}(s)\}\Big\}. \tag{A.3}$$

(i) For all $\alpha \in \mathcal{A}(t)$ and $\eta \in H(t)$, there exists $\alpha_1 \in \mathcal{A}(t)$, $\eta_1 \in H(t)$, $\alpha_2 \in \mathcal{A}(t+h)$, $\eta_2 \in H(t+h)$ such that

$$\alpha(s) = \begin{cases} \alpha_1(s), & s \in [t, t+h], \\ \alpha_2(s), & s \in (t+h, T], \end{cases} \tag{A.4}$$

$$\eta[\alpha](s) = \begin{cases} \eta_1[\alpha](s), & s \in [t, t+h], \\ \eta_2[\alpha](s), & s \in (t+h, T]. \end{cases} \tag{A.5}$$

Then, we have

$$V_1^+(t,x,z) = \sup_{\substack{\eta_1\in H(t)\\ \eta_2\in H(t+h)}} \inf_{\substack{\alpha_1\in\mathcal{A}(t)\\ \alpha_2\in\mathcal{A}(t+h)}} J_1(t,x,z,\alpha,\eta[\alpha]) \\ = \sup_{\eta_1\in H(t)} \inf_{\alpha_1\in\mathcal{A}(t)} \max\Big\{ \max_{s\in[t,t+h]} c(s,\mathrm{x}(s)),\ \max_{s\in[t,t+h]} g(s,\mathrm{x}(s)) - \mathrm{z}(s), \\ \sup_{\eta_2\in H(t+h)} \inf_{\alpha_2\in\mathcal{A}(t+h)} \max\{ \max_{s\in[t+h,T]} c(s,\mathrm{x}(s)),\ \max_{s\in[t+h,T]} g(s,\mathrm{x}(s)) - \mathrm{z}(s)\}\Big\}. \tag{A.6}$$

The last equality is deduced by combining (A.3) and that the first two terms of V_1^+ ($\max_{s\in[t,t+h]} c(s,\mathrm{x}(s))$, $\max_{s\in[t,t+h]} g(s,\mathrm{x}(s)) - \mathrm{z}(s)$) are independent of (α_2, η_2). (A.6) concludes (3.16).

(ii) The proof for (3.17) is similar to (i). □

A.3 Proof of Theorem 1

Proof. (i) At $t = T$, the definition of $V_1^\pm$ ((3.5) and (3.6)) implies (3.21).
(ii) For $U \in C^\infty([0,T]\times\mathbb{R}^n\times\mathbb{R})$ such that $V_1^+ - U$ has a local maximum at $(t_0,x_0,z_0) \in (0,T)\times\mathbb{R}^n\times\mathbb{R}$ and $(V_1^+ - U)(t_0,x_0,z_0) = 0$, we will prove

$$\max\{c(t_0,x_0) - U_0,\ g(t_0,x_0) - z_0 - U_0, \\ \frac{\partial U}{\partial t}(t_0,x_0,z_0) - \bar{H}^+(t_0,x_0,z_0,\frac{\partial U}{\partial x}(t_0,x_0,z_0),\frac{\partial U}{\partial z}(t_0,x_0,z_0))\} \geq 0. \tag{A.7}$$

Suppose not. There exists $\theta > 0$, $a_1 \in A$ such that

$$c(t,x) - U_0 < -\theta, \quad g(t,x) - z - U_0 < -\theta, \tag{A.8}$$

$$\frac{\partial U}{\partial t}(t,x,z) + \frac{\partial U}{\partial x}(t,x,z) \cdot f(t,x,a_1,b) - \frac{\partial U}{\partial z}(t,x,z)L(t,x,a_1,b) \le -\theta \tag{A.9}$$

for all $b \in B$ and all points (t,x,z) sufficiently close to (t_0,x_0,z_0): $|t-t_0|+\|x-x_0\|+|z-z_0| < h_1$ for small enough $h_1 > 0$. Consider state trajectories x and z solving (3.9) for $\alpha_1 \equiv a_1$, $t = t_0$, $x = x_0$, $z = z_0$, and any $\delta \in \mathcal{B}(t_0)$. By Assumption 1, there exists a small h such that $\|\mathrm{x}(s) - x_0\|+|\mathrm{z}(s) - z_0| < h_1 - h$ $(s \in [t_0, t_0+h])$, then,

$$c(s,\mathrm{x}(s)) - U_0 < -\theta, \ g(s,\mathrm{x}(s)) - \mathrm{z}(s) - U_0 < -\theta, \tag{A.10}$$

$$\frac{\partial U}{\partial t}(s,\mathrm{x}(s),\mathrm{z}(s)) + \frac{\partial U}{\partial x}(s,\mathrm{x}(s),\mathrm{z}(s)) \cdot f(s,\mathrm{x}(s),a_1,\delta(s)) - \frac{\partial U}{\partial z}(s,\mathrm{x}(s),\mathrm{z}(s))L(s,\mathrm{x}(s),a_1,\delta(s)) \le -\theta \tag{A.11}$$

for all $s \in [t_0, t_0+h]$ and $\delta \in \mathcal{B}(t_0)$.

Since $V_1^+ - U$ has a local maximum at (t_0,x_0,z_0),

$$\begin{aligned} &V_1^+(t_0+h,\mathrm{x}(t_0+h),\mathrm{z}(t_0+h)) - V_1^+(t_0,x_0,z_0) \\ \le &U(t_0+h,\mathrm{x}(t_0+h),\mathrm{z}(t_0+h)) - U(t_0,x_0,z_0) \int_{t_0}^{t_0+h} \frac{\partial U}{\partial t}(s,\mathrm{x}(s),\mathrm{z}(s)) \\ + &\frac{\partial U}{\partial x}(s,\mathrm{x}(s),\mathrm{z}(s)) \cdot f(s,\mathrm{x}(s),a_1,\eta[\alpha_1](s)) - \frac{\partial U}{\partial z}(s,\mathrm{x}(s),\mathrm{z}(s))L(s,\mathrm{x}(s),a_1,\eta[\alpha_1](s))ds \le -\theta h \end{aligned} \tag{A.12}$$

for all $\eta \in H(t_0)$, according to (A.11). Lemma 2 implies

$$\begin{aligned} V_1^+(t_0,x_0,z_0) \le \sup_{\eta \in H(t_0)} \max\{ &\max_{s \in [t_0,t_0+h]} c(s,\mathrm{x}(s)), \\ &\max_{s \in [t,t+h]} g(\mathrm{x}(s)) - \mathrm{z}(s), V_1^+(t_0+h,\mathrm{x}(t_0+h),\mathrm{z}(t_0+h))\}. \end{aligned} \tag{A.13}$$

By subtracting U_0 on the both sides in (A.13) and then applying (A.10) and (A.12), we have

$$0 \le \max\{-\theta, -\theta, -\theta h\} < 0, \tag{A.14}$$

which is contradiction. Thus, (A.7) is proved.

(iii) For $U \in C^\infty([0,T] \times \mathbb{R}^n \times \mathbb{R})$ such that $V_1^+ - U$ has a local minimum at $(t_0,x_0,z_0) \in (0,T) \times \mathbb{R}^n \times \mathbb{R}$ and $(V_1^+ - U)(t_0,x_0,z_0) = 0$, we will prove

$$\begin{aligned} \max\{&c(t_0,x_0) - U_0, g(t_0,x_0) - z_0 - U_0, \\ &\frac{\partial U}{\partial t}(t_0,x_0,z_0) - \bar{H}^+(t_0,x_0,z_0,\frac{\partial U}{\partial x}(t_0,x_0,z_0),\frac{\partial U}{\partial z}(t_0,x_0,z_0))\} \le 0, \end{aligned} \tag{A.15}$$

Since $J_1(t_0, x_0, z_0, \alpha)$ (3.7) is greater than the value at $\tau = t_0$,

$$J_1(t_0, x_0, z_0, \alpha, \eta[\alpha]) \geq \max\{c(x_0, x_0), g(t_0, x_0) - z_0\}, \tag{A.16}$$

for all $\alpha \in \mathcal{A}(t_0), \eta \in H(t_0)$. By subtracting U_0 on the both sides, and taking the supremum over η and the infimum over α, sequentially, on the both side, we have

$$0 \geq \max\{c(x_0, x_0) - U_0, g(t_0, x_0) - z_0 - U_0\}. \tag{A.17}$$

The rest of the proof is to show

$$\frac{\partial U}{\partial t}(t_0, x_0, z_0) - \bar{H}^+(t_0, x_0, z_0, \frac{\partial U}{\partial x}(t_0, x_0, z_0), \frac{\partial U}{\partial z}(t_0, x_0, z_0)) \leq 0. \tag{A.18}$$

Suppose not. For some $\theta > 0$,

$$\frac{\partial U}{\partial t}(t, x, z) + \max_{b \in B} \frac{\partial U}{\partial x}(t, x, z) \cdot f(t, x, a, b) - \frac{\partial U}{\partial z}(t, x, z) L(t, x, a, b) \geq \theta \tag{A.19}$$

for all $a \in A$ and all points (t, x, z) sufficiently close to (t_0, x_0, z_0): $|t - t_0| + \|x - x_0\| + |z - z_0| < h_1$ for small enough $h_1 > 0$. Consider state trajectories x_1 and z_1 solving (3.9) for any $\alpha \in \mathcal{A}(t_0)$, $\delta = \eta_1[\alpha]$, where

$$\begin{aligned} \eta_1[\alpha](s) \in \arg\max_{b \in B} & \frac{\partial U}{\partial x}(s, \mathrm{x}_1(s), \mathrm{z}_1(s)) \cdot f(s, \mathrm{x}_1(s), \alpha(s), b) \\ & - \frac{\partial U}{\partial z}(s, \mathrm{x}_1(s), \mathrm{z}_1(s)) L(s, \mathrm{x}_1(s), \alpha(s), b), \end{aligned} \tag{A.20}$$

$t = t_0$, $x = x_0$, and $z = z_0$. Since there exists a small $h > 0$ such that $\|\mathrm{x}_1(s) - x_0\| + |\mathrm{z}_1(s) - z_0| < h_1 - h$ $(s \in [t_0, t_0 + h])$,

$$\begin{aligned} \frac{\partial U}{\partial t}(s, \mathrm{x}_1(s), \mathrm{z}_1(s)) + & \frac{\partial U}{\partial x}(s, \mathrm{x}_1(s), \mathrm{z}_1(s)) \cdot f(s, \mathrm{x}_1(s), \alpha(s), \eta_1[\alpha](s)) \\ & - \frac{\partial U}{\partial z}(s, \mathrm{x}_1(s), \mathrm{z}_1(s)) L(s, \mathrm{x}_1(s), \alpha(s), \eta_1[\alpha](s)) \geq \theta \end{aligned} \tag{A.21}$$

for all $s \in [t_0, t_0 + h]$. By integrating (A.21) over $s \in [t_0, t_0 + h]$, we have

$$U(t_0 + h, \mathrm{x}_1(t_0 + h), \mathrm{z}_1(t_0 + h)) - U(t_0, x, z) \geq \theta h. \tag{A.22}$$

Since (A.22) holds for all $\alpha \in \mathcal{A}(t_0)$ and $\eta \in H(t_0)$,

$$\sup_{\eta \in H(t_0)} \inf_{\alpha \in \mathcal{A}(t_0)} U(t_0 + h, \mathrm{x}(t_0 + h), \mathrm{z}(t_0 + h)) - U(t_0, x, z) \geq \theta h, \tag{A.23}$$

where x, z solve (3.9) for $(\alpha, \eta, t_0, x_0, z_0)$.

Since $V_1^+ - U$ has a local minimum at (t_0, x_0, z_0),

$$\begin{aligned} &\sup_{\eta\in H(t_0)} \inf_{\alpha\in\mathcal{A}(t_0)} \quad V_1^+(t_0+h, \mathrm{x}(t_0+h), \mathrm{z}(t_0+h)) - V_1^+(t_0, x_0, z_0) \\ \geq &\sup_{\eta\in H(t_0)} \inf_{\alpha\in\mathcal{A}(t_0)} \quad U(t_0+h, \mathrm{x}(t_0+h), \mathrm{z}(t_0+h)) - U(t_0, x_0, z_0) \\ \geq &\ \theta h \end{aligned} \tag{A.24}$$

according to (A.23). However, Lemma 2 implies

$$\sup_{\eta\in H(t_0)} \inf_{\alpha\in\mathcal{A}(t_0)} V_1^+(t_0+h, \mathrm{x}(t_0+h), \mathrm{z}(t_0+h)) \leq V_1^+(t_0, x_0, z_0), \tag{A.25}$$

which contradicts (A.24).

(iv) The proof for the viscosity solution V_1^- is similar to (ii) and (iii) for V_1^+. Also, the uniqueness follows from the uniqueness theorems for viscosity solutions, Theorem 4.2 in [13], and the extension of Theorem 1 in [42]. □

A.4 Proof of Lemma 3

Proof. Set $\tilde{V}_1^+$ and $\tilde{V}_1^-$ be the right hand terms in (3.32) and (3.33), respectively. V_1^+ are V_1^- are defined in (3.5) and (3.6), respectively.

(i) In this proof, we utilize the following properties in [76, 66], presented as below.

Define a pseudo-time operator $\sigma_\mu : [t, T] \to [t, T]$ for a given $\mu \in \mathcal{M}(t)$ (defined in (3.23)) and the corresponding inverse operator:

$$\sigma_\mu(s) = \int_t^s \mu(\tau)d\tau + t; \tag{A.26}$$

$$\sigma_\mu^{-1}(s) \coloneqq \min \tau \text{ subject to } \sigma_\mu(\tau) = s. \tag{A.27}$$

Then,

$$\sigma_\mu(\sigma_\mu^{-1}(s)) = s, \quad s \in [t, \sigma_\mu(T)], \tag{A.28}$$

$$\sigma_\mu^{-1}(\sigma_\mu(s)) = s, \quad s \in \mathrm{Range}(\sigma_\mu^{-1}), \tag{A.29}$$

where $\mathrm{Range}(\sigma_\mu^{-1}) \coloneqq \{\sigma_\mu^{-1}(s) \mid s \in [t, \sigma_\mu(T)]\}$.

Consider two state trajectories: (x, z) solving (3.9) for $(\tilde{\alpha}(\sigma_\mu^{-1}(\cdot)), \tilde{\delta}(\sigma_\mu^{-1}(\cdot)))$ for $s \in [t, \sigma_\mu(T)]$; $(\tilde{\mathrm{x}}, \tilde{\mathrm{z}})$ solving (3.31) for $(\tilde{\alpha}, \tilde{\delta}, \mu)$, and $\mathrm{x}(t) = \tilde{\mathrm{x}}(t) = x$. Then,

$$\mathrm{x}(\sigma_\mu(s)) = \tilde{\mathrm{x}}(s), \quad s \in [t, T], \tag{A.30}$$

$$g(\mathrm{x}(\sigma_\mu(T))) - \mathrm{z}(\sigma_\mu(T)) = g(\tilde{\mathrm{x}}(T)) - \tilde{\mathrm{z}}(T). \tag{A.31}$$

(A.30) is according to Lemma 4 in [79], and (A.31) is derived by combining two lemmas (Lemma 4 and 6) in [79].

(ii) $\tilde{V}_1^+(t,x,z) \geq V_1^+(t,x,z)$

For small $\epsilon > 0$, there exists $\eta_1 \in H(t)$ such that

$$V_1^+(t,x,z) - \epsilon \leq \inf_{\alpha} \max_{\tau\in[t,T]} \max\{ \max_{s\in[t,\tau]} c(\mathrm{x}_1(s)), g(\mathrm{x}_1(\tau)) - \mathrm{z}_1(\tau)\}, \tag{A.32}$$

where $(\mathrm{x}_1, \mathrm{z}_1)$ solves (3.9) for $(\alpha, \eta_1[\alpha])$. Denote $\tau_*(\alpha)$ is the maximizer of the right hand term in (A.32) for each $\alpha \in \mathcal{A}(t)$:

$$\tau_*(\alpha) \coloneqq \arg\max_{\tau\in[t,T]} \max\{ \max_{s\in[t,\tau]} c(\mathrm{x}_1(s)), g(\mathrm{x}_1(\tau)) - \mathrm{z}_1(\tau)\}. \tag{A.33}$$

Define a particular strategy $\nu_{A,1} \in \mathrm{N}_A(t)$:

$$\nu_{A,1}[\alpha](s) \coloneqq \begin{cases} 1, & s \in [t, \tau_*(\alpha)], \\ 0, & s \in (\tau_*(\alpha), T]. \end{cases} \tag{A.34}$$

Consider a state trajectory $(\tilde{\mathrm{x}}_1, \tilde{\mathrm{z}}_1)$ solving (3.31) for $(\alpha, \eta_1[\alpha], \nu_{A,1}[\alpha])$. Then, we have

$$(\tilde{\mathrm{x}}_1, \tilde{\mathrm{z}}_1)(s) = \begin{cases} (\mathrm{x}_1, \mathrm{z}_1)(s), & s \in [t, \tau_*(\alpha)], \\ (\mathrm{x}_1, \mathrm{z}_1)(\tau_*(\alpha)), & s \in (\tau_*(\alpha), T], \end{cases} \tag{A.35}$$

Since $\tilde{V}_1^+$ has the supremum over (η, ν_A)-space operation,

$$\begin{aligned} \tilde{V}_1^+(t,x,z) &\geq \inf_{\alpha} \max\{ \max_{s\in[t,T]} c(\tilde{\mathrm{x}}_1(s)), g(\tilde{\mathrm{x}}_1(T)) - \tilde{\mathrm{z}}_1(T)\} \\ &= \inf_{\alpha} \max\{ \max_{s\in[t,\tau_*(\alpha)]} c(\mathrm{x}_1(s)), g(\mathrm{x}_1(\tau_*(\alpha))) - \mathrm{z}_1(\tau_*(\alpha))\} \\ &\geq V_1^+(t,x,z) - \epsilon. \end{aligned} \tag{A.36}$$

The second equality is according to (A.35), and the third inequality is by (A.32).

(iii) $V_1^+(t,x,z) \geq \tilde{V}_1^+(t,x,z)$

Define $\tilde{\mathfrak{A}}_\mu : \mathcal{A}(t) \to \mathcal{A}(t)$ and its psuedo inverse function $\mathfrak{A}_\mu : \mathcal{A}(t) \to \mathcal{A}(t)$:

$$(\tilde{\mathfrak{A}}_\mu(\alpha))(s) \coloneqq \begin{cases} \alpha(\sigma_\mu(s)), & s \in \mathrm{Range}(\sigma_\mu^{-1}), \\ \text{any } a \in A, & s \notin \mathrm{Range}(\sigma_\mu^{-1}), \end{cases} \tag{A.37}$$

$$(\mathfrak{A}_\mu(\tilde{\alpha}))(s) \coloneqq \begin{cases} \tilde{\alpha}(\sigma_\mu^{-1}(s)), & s \in [t, \sigma_\mu(T)], \\ \text{any } a \in A, & s \in (\sigma_\mu(T), T], \end{cases} \tag{A.38}$$

Also, define $\tilde{\mathfrak{D}}_\mu : H(t) \to H(t)$ and its psuedo inverse function $\mathfrak{D}_\mu : H(t) \to H(t)$:

$$(\tilde{\mathfrak{D}}_\mu(\eta))[\tilde{\alpha}](s) = \begin{cases} \eta[\mathfrak{A}_\mu(\tilde{\alpha})](\sigma_\mu(s)), & s \in \mathrm{Range}(\sigma_\mu^{-1}), \\ \text{any } b \in B, & s \notin \mathrm{Range}(\sigma_\mu^{-1}), \end{cases} \tag{A.39}$$

$$(\mathfrak{D}_\mu(\tilde{\eta}))[\alpha](s) = \begin{cases} \tilde{\eta}[\tilde{\mathfrak{A}}_\mu(\alpha)](\sigma_\mu^{-1}(s)), & s \in [t, \sigma_\mu(T)], \\ \text{any } b \in B, & s \in (\sigma_\mu(T), T]. \end{cases} \tag{A.40}$$

These definitions satisfy the following properties:

$$\begin{aligned} &(\tilde{\mathfrak{A}}_\mu(\mathfrak{A}_\mu(\tilde{\alpha})))(s) = \tilde{\alpha}(s), \\ &(\tilde{\mathfrak{D}}_\mu(\mathfrak{D}_\mu(\tilde{\eta})))[\tilde{\alpha}](s) = \tilde{\eta}[\tilde{\alpha}](s), \end{aligned} \quad \text{for } s \in \mathrm{Range}(\sigma_\mu^{-1}) \tag{A.41}$$

$$\begin{aligned} &(\mathfrak{A}_\mu(\tilde{\mathfrak{A}}_\mu(\alpha)))(s) = \alpha(s), \\ &(\mathfrak{D}_\mu(\tilde{\mathfrak{D}}_\mu(\eta)))[\alpha](s) = \eta[\alpha](s), \end{aligned} \quad \text{for } s \in [t, \sigma_\mu(T)], \tag{A.42}$$

$$\{\alpha = \mathfrak{A}_\mu(\tilde{\alpha}) \mid \tilde{\alpha} \in \mathcal{A}(t)\} = \mathcal{A}(t), \forall \mu \in \mathcal{M}(t) \tag{A.43}$$

$$\{\eta = \mathfrak{D}_\mu(\tilde{\eta}) \mid \tilde{\eta} \in H(t)\} = H(t), \forall \mu \in \mathcal{M}(t). \tag{A.44}$$

Consider $(\tilde{\mathrm{x}}, \tilde{\mathrm{z}})$ solving (3.31) for $(\tilde{\alpha}, \tilde{\eta}[\tilde{\alpha}], \mu)$, (x, z) solving (3.9) for $(\mathfrak{A}_\mu(\tilde{\alpha}), (\mathfrak{D}_\mu(\tilde{\eta}))[\mathfrak{A}_\mu(\tilde{\alpha})])$, and $(\mathrm{x}_1, \mathrm{z}_1)$ solving (3.9) for $(\alpha, \eta[\alpha])$. Then, we have

$$\begin{aligned} &\sup_{\tilde{\eta} \in H(t)} \inf_{\tilde{\alpha} \in \mathcal{A}(t)} \max\{ \max_{s \in [t,T]} c(\tilde{\mathrm{x}}(s)), g(\tilde{\mathrm{x}}(T)) - \tilde{\mathrm{z}}(T)\} \\ =& \sup_{\tilde{\eta} \in H(t)} \inf_{\tilde{\alpha} \in \mathcal{A}(t)} \max\{ \max_{s \in [t, \sigma_\mu(T)]} c(\mathrm{x}(s)), g(\mathrm{x}(\sigma_\mu(T))) - \mathrm{z}(\sigma_\mu(T))\}, && \text{(A.45)} \\ =& \sup_{\eta \in H(t)} \inf_{\alpha \in \mathcal{A}(t)} \max\{ \max_{s \in [t, \sigma_\mu(T)]} c(\mathrm{x}_1(s)), g(\mathrm{x}_1(\sigma_\mu(T))) - \mathrm{z}_1(\sigma_\mu(T))\}, && \text{(A.46)} \\ \leq& V_1^+(t, x, z). && \text{(A.47)} \end{aligned}$$

(A.45) is by (A.30) and (A.31), and (A.46) is according to (A.43) and (A.44). Since the above inequality holds for all μ, we substitute $\nu_A[\alpha]$ for μ and take the supremum over ν_A on the both sides, which concludes $\tilde{V}_1^+(t, x, z) \leq V_1^+(t, x, z)$.

By (ii) and (iii), we conclude $V_1^+(t, x, z) = \tilde{V}_1^+(t, x, z)$.

(iv) $V_1^-(t, x) = \tilde{V}_1^-(t, x)$

Define $\tilde{\mathfrak{B}}_\mu : \mathcal{B}(t) \to \mathcal{B}(t)$ and its pseudo inverse function $\mathfrak{B}_\mu : \mathcal{B}(t) \to \mathcal{B}(t)$:

$$(\tilde{\mathfrak{B}}_\mu(\delta))(s) \coloneqq \begin{cases} \delta(\sigma_\mu(s)), & s \in \mathrm{Range}(\sigma_\mu^{-1}), \\ \text{any } b \in B, & s \notin \mathrm{Range}(\sigma_\mu^{-1}), \end{cases} \tag{A.48}$$

$$(\mathfrak{B}_\mu(\tilde{\delta}))(s) \coloneqq \begin{cases} \tilde{\delta}(\sigma_\mu^{-1}(s)), & s \in [t, \sigma_\mu(T)], \\ \text{any } b \in B, & s \in (\sigma_\mu(T), T], \end{cases} \tag{A.49}$$

Also, define $\tilde{\mathfrak{C}}_\mu : \Gamma(t) \to \tilde{\Gamma}(t)$, where $\tilde{\Gamma}(t)$ is defined in (3.25), and its pseudo inverse function $\mathfrak{C}_\mu : \tilde{\Gamma}(t) \to \Gamma(t)$:

$$(\tilde{\mathfrak{C}}_\mu(\gamma))[\tilde{\delta}, \mu](s) = \begin{cases} \gamma[\mathfrak{B}_\mu(\tilde{\delta})](\sigma_\mu(s)), & s \in \mathrm{Range}(\sigma_\mu^{-1}), \\ \text{any } a \in A, & s \notin \mathrm{Range}(\sigma_\mu^{-1}), \end{cases} \tag{A.50}$$

$$(\mathfrak{C}_\mu(\tilde{\gamma}))[\delta](s) = \begin{cases} \tilde{\gamma}[\tilde{\mathfrak{B}}_\mu(\delta), \mu](\sigma_\mu^{-1}(s)), & s \in [t, \sigma_\mu(T)], \\ \text{any } a \in A, & s \in (\sigma_\mu(T), T]. \end{cases} \tag{A.51}$$

These definitions satisfy the following properties: for any $\mu \in \mathcal{M}(t)$,

$$\{\delta = \mathfrak{B}_\mu(\tilde{\delta}) \mid \tilde{\delta} \in \mathcal{B}(t)\} = \mathcal{B}(t), \tag{A.52}$$

$$\{\gamma = \mathfrak{C}_\mu(\tilde{\gamma}) \mid \tilde{\gamma} \in \tilde{\Gamma}(t)\} = \Gamma(t). \tag{A.53}$$

Consider $(\tilde{\mathrm{x}}, \tilde{\mathrm{z}})$ solving (3.31) for $(\tilde{\gamma}[\tilde{\delta}, \mu], \tilde{\delta}, \mu)$, (x, z) solving (3.9) for $(\mathfrak{C}_\mu(\tilde{\gamma})[\mathfrak{B}_\mu(\tilde{\delta})], \mathfrak{B}_\mu(\tilde{\delta}))$, and $(\mathrm{x}_1, \mathrm{z}_1)$ solving (3.9) for $(\gamma[\delta], \delta)$.

$$\begin{aligned} \tilde{V}_1^-(t,x,z) &= \inf_{\tilde{\gamma}\in\tilde{\Gamma}(t)} \sup_{\tilde{\delta}\in\mathcal{B}(t),\mu\in\mathcal{M}(t)} \max\{ \max_{s\in[t,T]} c(\tilde{\mathrm{x}}(s)), g(\tilde{\mathrm{x}}(T)) - \tilde{\mathrm{z}}(T)\} \\ &= \inf_{\tilde{\gamma}\in\tilde{\Gamma}(t)} \sup_{\tilde{\delta}\in\mathcal{B}(t),\mu\in\mathcal{M}(t)} \max\{ \max_{s\in[t,\sigma_\mu(T)]} c(\mathrm{x}(s)), g(\mathrm{x}(\sigma_\mu(T))) - \mathrm{z}(\sigma_\mu(T))\} \qquad \text{(A.54)} \\ &= \inf_{\gamma\in\Gamma(t)} \sup_{\delta\in\mathcal{B}(t),\mu\in\mathcal{M}(t)} \max\{ \max_{s\in[t,\sigma_\mu(T)]} c(\mathrm{x}_1(s)), g(\mathrm{x}_1(\sigma_\mu(T))) - \mathrm{z}_1(\sigma_\mu(T))\}. \qquad \text{(A.55)} \end{aligned}$$

(A.54) is by (A.30) and (A.31), and (A.55) is by (A.52) and (A.53). In the term in (A.55), μ only controls the terminal time ($\sigma_\mu(T)$), hence, the supremum over μ can be converted to the maximum over τ, which concludes $V_1^-(t,x,z) = \tilde{V}_1^-(t,x,z)$.

□

A.5 Proof of Theorem 2

Proof. The terminal value is derived by substituting T for t in (3.32) or (3.33):

$$V_1^\pm(T,x,z) = \max\{c(T,x), g(T,x) - z\} \tag{A.56}$$

for all $(x,z) \in \mathbb{R}^n \times \mathbb{R}$.

(i) [4] has presented the HJ equation for state-constrained problems, in which the terminal time is fixed. By applying the HJ equation in [4] to V_1^+,

$$0 = \max\{c(x) - V_1^+, V_{1,t}^+ - \tilde{H}_1^+(x, z, \frac{\partial V_1^+}{\partial x}, \frac{\partial V_1^+}{\partial z})\}, \tag{A.57}$$

where

$$\tilde{H}_1^+(x,z,p,q) \coloneqq \max_{a\in A} \min_{\substack{b\in B \\ b_d\in[0,1]}} -p\cdot f(x,a,b)b_d + qL(x,a,b)b_d. \tag{A.58}$$

Since, for all $a \in A, b \in B$, the term $-p\cdot f(x,a,b)b_d + qL(x,a,b)b_d$ is minimized at $b_d = 0$ or 1,

$$\tilde{H}_1^+(x,z,p,q) = \max_{a\in A} \min\{0, \min_{b\in B} -p\cdot f(x,a,b) + qL(x,a,b)\}. \tag{A.59}$$

Also, 0 does not depend on a, thus, the maximum over a operation can move into the minimum operation:

$$\tilde{H}_1^+(x,z,p,q) = \min\{0, \bar{H}^+(x,z,p,q)\}, \tag{A.60}$$

where $\bar{H}^+$ is defined in (3.19). By applying (A.60) to (A.57), (3.35) is proved for V_1^+.

(ii) By applying [4] to V_1^-,

$$0 = \max\{c(x) - V_1^-, V_{1,t}^- - \tilde{H}_1^-(x,z,\frac{\partial V_1^-}{\partial x}, \frac{\partial V_1^-}{\partial z})\}, \tag{A.61}$$

where

$$\tilde{H}_1^-(x,z,p,q) \coloneqq \min_{\substack{b\in B \\ b_d\in[0,1]}} \max_{a\in A} -p\cdot f(x,a,b)b_d + qL(x,a,b)b_d. \tag{A.62}$$

Since $b_d \in [0,1]$ is non-negative,

$$\begin{aligned}\tilde{H}_1^-(x,z,p,q) &= \min_{b_d\in[0,1]} b_d \min_{b\in B}\max_{a\in A}[-p\cdot f(x,a,b) + qL(x,a,b)] \\ &= \min\{0, \bar{H}^-(x,z,p,q)\},\end{aligned} \tag{A.63}$$

where $\bar{H}^-$ is defined in (3.20). (A.61) and (A.63) prove (3.35) for V_1^-. □

Appendix B

Proof for Section 3.2

B.1 Proof of Theorem 3

Proof. We first prove that $\vartheta_2^+(t,x) - z \leq 0 \Leftrightarrow V_2^+(t,x,z) \leq 0$.

1. $\vartheta_2^+(t,x) - z \leq 0 \Rightarrow V_2^+(t,x,z) \leq 0$
   By (3.77), for any admissible $\delta \in \Delta(t)$,

$$\vartheta^+(t,x) \geq \inf_{\alpha \in \mathcal{A}(t)} \min_{\tau \in [t,T]} \int_t^\tau L(s, \mathrm{x}(s), \alpha(s), \eta[\alpha](s))ds + g(\tau, \mathrm{x}(\tau)) \tag{B.1}$$

   subject to $c(s, \mathrm{x}(s)) \leq 0$ ($s \in [t,\tau]$), where x solves (2.1) for α and $\eta[\alpha]$. By $\vartheta_2^+(t,x) - z \leq 0$ and (B.1),

$$0 \geq \inf_{\alpha \in \mathcal{A}(t)} \min_{\tau \in [t,T]} \max \left\{ \max_{s \in [t,\tau]} c(s, \mathrm{x}(s)), \int_t^\tau L(s, \mathrm{x}(s), \alpha(s), \eta[\alpha](s))ds + g(\tau, \mathrm{x}(\tau)) - z \right\}. \tag{B.2}$$

   Since (B.2) holds for all $\eta \in H(t)$, $V_2^+(t,x,z) \leq 0$.

2. $V_2^+(t,x,z) \leq 0 \Rightarrow \vartheta_2^+(t,x) - z \leq 0$
   By (3.77), for any admissible $\eta \in H(t)$,

$$V_2^+(t,x,z) \geq \inf_{\alpha \in \mathcal{A}(t)} \min_{\tau \in [t,T]} \max \left\{ \max_{s \in [t,\tau]} c(s, \mathrm{x}(s)), \int_t^\tau L(s, \mathrm{x}(s), \alpha(s), \eta[\alpha](s))ds + g(\tau, \mathrm{x}(\tau)) - z \right\}. \tag{B.3}$$

   Since A is convex, there exists an admissible $\bar{\alpha} \in \mathcal{A}(t)$ that achieves the infimum of the right term of (B.3) [4]. Also, by continuity of $\mathrm{x}(\cdot)$ in t, there also exists $\bar{\tau}$ that

minimizes the right term of (B.3) in time. Define $\bar{\mathrm{x}}(\cdot)$ solving (2.1) for $\bar{\alpha}$ and $\eta[\bar{\alpha}]$. Then,

$$0 \geq \int_t^{\bar{\tau}} L(s, \bar{\mathrm{x}}(s), \bar{\alpha}(s), \eta[\bar{\alpha}](s))ds + g(\bar{\tau}, \mathrm{x}(\bar{\tau})) - z, \tag{B.4}$$

and $c(s, \bar{\mathrm{x}}(s)) \leq 0$ for $s \in [t, \bar{\tau}]$. Since (B.4) holds for any $\eta \in H(t)$, $\vartheta^+(t,x) - z \leq 0$.

We will prove $\vartheta_2^-(t,x) - z \leq 0 \Leftrightarrow V_2^-(t,x,z) \leq 0$.

3. $\vartheta_2^-(t,x) - z \leq 0 \Rightarrow V_2^-(t,x,z) \leq 0$

For any small $\epsilon > 0$, there exists $\bar{\zeta} \in Z(t)$ such that

$$\vartheta_2^-(t,x) + \epsilon > \sup_{\delta \in \mathcal{D}(t)} \min_{\tau \in [t,T]} \int_t^{\tau} L(s, \bar{\mathrm{x}}(s), \bar{\zeta}[\delta](s), \delta(s))ds + g(\tau, \bar{\mathrm{x}}(\tau)) \tag{B.5}$$

and $c(s, \bar{\mathrm{x}}(s)) \leq 0$ for all $s \in [t, \tau]$, where $\bar{\mathrm{x}}(\cdot)$ solves (2.1) with $\bar{\zeta}[\delta]$ and δ. Since $\vartheta^-(t,x) - z \leq 0$,

$$\begin{aligned} \epsilon &> \sup_{\delta \in \mathcal{D}(t)} \min_{\tau \in [t,T]} \max \left\{ \max_{s \in [t,\tau]} c(s, \bar{\mathrm{x}}(s)), \int_t^{\tau} L(s, \bar{\mathrm{x}}(s), \bar{\zeta}[\delta](s), \delta(s))ds + g(\tau, \bar{\mathrm{x}}(\tau)) - z \right\} \\ &\geq V_2^-(t,x,z). \end{aligned} \tag{B.6}$$

Since (B.6) holds for any small $\epsilon > 0$, $0 \geq V_2^-(t,x,z)$.

4. $V_2^-(t,x,z) \leq 0 \Rightarrow \vartheta^-(t,x) - z \leq 0$

Since A is convex, $\mathcal{A}(t)$ is compact. Thus, there exists $\bar{\zeta} \in Z(t)$ that achieves the minimum of $V_2^-(t,x,z)$.

$$V_2^-(t,x,z) = \sup_{\delta \in \mathcal{D}(t)} \min_{\tau \in [t,T]} \max \left\{ \max_{s \in [t,\tau]} c(s, \bar{\mathrm{x}}(s)), \int_t^{\tau} L(s, \bar{\mathrm{x}}(s), \bar{\zeta}[\delta](s), \delta(s))ds + g(\tau, \bar{\mathrm{x}}(\tau)) - z \right\},$$

where $\bar{\mathrm{x}}(\cdot)$ solves (2.1) with $\bar{\zeta}[\delta]$ and δ. Since $V^-(t,x,z) \leq 0$, $c(s, \bar{\mathrm{x}}(s)) \leq 0$ for $s \in [t, \tau]$ and

$$z \geq \int_t^{\tau} L(s, \bar{\mathrm{x}}(s), \bar{\zeta}[\delta](s), \delta(s))ds + g(\tau, \bar{\mathrm{x}}(\tau)) \tag{B.7}$$

The right term in (B.7) is greater than or equal to $\vartheta_2^-(t,x)$, and we conclude $\vartheta_2^-(t,x) - z \leq 0$.

□

Appendix C

Proof for Section 4.1

C.1 Proof of Lemma 6

(i) Proof of (4.15) and (4.16)

A state-unconstrained problem

$$\inf_{\alpha} \int_t^T L(s, \mathrm{x}(s), \alpha(s))ds + g(\mathrm{x}(T)), \tag{C.1}$$

where x solves (2.1), is equivalent to

$$\inf_{\beta} \int_t^T L^b(s, \mathrm{x}(s), \beta(s))ds + g(\mathrm{x}(T)), \tag{C.2}$$

where x solves $\dot{\mathrm{x}}(s) = -\beta(s)$ and $\beta(s) \in B(s, \mathrm{x}(s))$, defined in (4.7), for $s \in [t, T]$. Since the corresponding Hamiltonian for the two state-unconstrained problems is the same,

$$\max_{b \in B(s,x)} [p \cdot b - L^b(s, x, b)] = H(s, x, p).$$

Hence, $H \equiv (L^b)^*$ and $H^* \equiv (L^b)^{**}$ since L^b is semi lower-continuous in b for each $(s, x) \in [t, T] \times \mathbb{R}^n$.

(ii) Proof of (4.17)

Case 1. $b \in \mathrm{co}(B(s, x))$

There exist $b_i \in B(s, x)$ and $\gamma_i \geq 0$ ($\sum_i \gamma_i = 1$) such that $b = \sum_i \gamma_i b_i$. Since $H^*(s, x, \cdot)$ is convex in b,

$$H^*(s, x, b) \leq \sum_i \gamma_i H^*(s, x, b_i) < \infty$$

since each of $H^*(s, x, b_i)$ is finite.

Case 2. $b \notin \mathrm{co}(B(s, x))$

For two closed convex sets $\{b\}$ and $\text{co}(B(s,x))$, the seperating hyperplane theorem [19] implies that there exists a hyperplane $(P : \mathbb{R}^n \to \mathbb{R})$: $P(b') := p' \cdot b' + c$ such that $P(b')$ is positive if $b' \notin \text{co}(B(s,x))$, otherwise, $P(b')$ is non-positive. By picking $p = \gamma p'$ where $\gamma \in \mathbb{R}$,

$$\begin{aligned} H^*(s,x,b) &= \max_p \min_{b'} [p \cdot (b - b') + L^b(s,x,b')] \\ &\geq \max_\gamma \min_{b'} \gamma p' \cdot (b - b') + L^b(s,x,b') = \infty \end{aligned}$$

since $p' \cdot (b - b') > 0$ for all $b' \in \text{co}(B(s,x))$. $\square$

C.2 Additional Lemma used in the proof of Theorem 7

In this appendix, we describe the below lemma that is used in the proof of Theorem 7.

Lemma 19. Suppose Assumption 1 holds. Given the initial state $x \in \mathbb{R}^n$, there exists a constant $C > 0$ such that

$$\|\mathrm{x}(s) - x\| \leq C(s - t) \tag{C.3}$$

for all $s \in [t,T]$, $\alpha \in \mathcal{A}(t)$ where x solves (2.1).

Proof. The Lipschitz assumption for the dynamics and Gronwall's inequality conclude the proof. $\square$

C.3 Proof of Theorem 7

Proof.
Step 1. For a temporal partition $t_0 = t < ... < t_K = T$, consider a state trajectory $(\mathrm{x}_0 : [t,T] \to \mathbb{R}^n)$ solving

$$\dot{\mathrm{x}}_0^\epsilon(s) = -\beta(t_k), \quad s \in [t_k, t_{k+1})$$

for $s \in [t,T]$ and $\mathrm{x}_0(t) = x$. Since β is Riemann integrable, for all ϵ, there exists $\delta > 0$ such that $|t_{k+1} - t_k| < \delta$ for all k and

$$\|\mathrm{x}_0 - \mathrm{x}\|_{L^\infty(t,T)} < \epsilon. \tag{C.4}$$

As δ converges to 0, ϵ converges to 0.

Since $\beta(t_k)$ is in $\text{co}(B(t_k, \mathrm{x}(t_k)))$, by Corollary 2, there exists $b_i^k \in B(t_k, \mathrm{x}(t_k))$, $a_i^k \in A$, and γ_i^k such that

$$H^*(t_k, \mathrm{x}(t_k), \beta(t_k)) = \sum_i \gamma_i^k L^b(t_k, \mathrm{x}(t_k), b_i^k) = \sum_i \gamma_i^k L(t_k, \mathrm{x}(t_k), a_i^k), \tag{C.5}$$

$$\beta(t_k) = \textstyle\sum_i \gamma_i^k b_i^k, \quad b_i^k = -f(t_k, \mathrm{x}(t_k), a_i^k), \tag{C.6}$$

and $\sum_i \gamma_i^k = 1$ for $k = 0, ..., K-1$.

Step 2. In each time interval $[t_k, t_{k+1})$, we construct a finer temporal discretization: $[t_{k,0} = t_k, ..., t_{k,i}, ..., t_{k,\bar{i}_k} = t_{k+1})$, where $t_{k,i} = t_k + \sum_{j=1}^{i} \gamma_j^k \Delta t_k$ and $\bar{i}_k$ denotes the number of γ_i^k in (C.5) for each k. Define a control input $\beta_1 : [t, T] \to \mathbb{R}^n$:

$$\beta_1(s) = b_{i+1}^k, \quad s \in [t_{k,i}, t_{k,i+1}),$$

where b_i^k is defined in (C.5) and (C.6), and also define the corresponding state trajectory $(\mathrm{x}_1 : [t, T] \to \mathbb{R}^n)$ solving

$$\dot{\mathrm{x}}_1(s) = -\beta_1(s) = f(t_k, \mathrm{x}(t_k), a_{i+1}^k)$$

for $s \in [t_{k,i}, t_{k,i+1})$, and $\mathrm{x}_1(t) = x$. Then, for $k = 0, ..., K$, $\mathrm{x}_1(t_k) = \mathrm{x}_0(t_k)$, and for $s \in [t_k, t_{k+1})$,

$$\begin{aligned} \|\mathrm{x}_1(s) - \mathrm{x}(s)\| &\le \|\mathrm{x}_1(s) - \mathrm{x}_0(s)\| + \|\mathrm{x}_0(s) - \mathrm{x}(s)\| \\ &\le \int_{t_k}^{s} \|-\beta_1(\tau) + \beta(t_k)\| d\tau + \epsilon \quad \text{by (C.4)} \\ &\le c_1 \delta + \epsilon, \end{aligned} \tag{C.7}$$

where $c_1 = \max_{k,i} \|b_i^k - \beta(t_k)\|$. c_1 is bounded since x and A are bounded.

Step 3. We consider a control input $\alpha_2 \in \mathcal{A}(t)$

$$\alpha_2(s) = a_{i+1}^k, \quad s \in [t_{k,i}, t_{k,i+1}), \tag{C.8}$$

where a_i^k is defined in (C.5) and (C.6). and the corresponding state trajectory $(\mathrm{x}_2 : [t, T] \to \mathbb{R}^n)$ solving

$$\dot{\mathrm{x}}_2(s) = f(s, \mathrm{x}_2(s), \alpha_2(s)), \quad s \in [t, T].$$

We have

$$\begin{aligned} \|\mathrm{x}_2(t_{k,i+1}) - \mathrm{x}_1(t_{k,i+1})\| &\le \|\mathrm{x}_2(t_{k,i}) - \mathrm{x}_1(t_{k,i})\| + \int_{t_{k,i}}^{t_{k,i+1}} \|f(\tau, \mathrm{x}_2(\tau), a_{i+1}^k) - f(t_k, \mathrm{x}(t_k), a_{i+1}^k)\| d\tau \\ &\le \|\mathrm{x}_2(t_{k,i}) - \mathrm{x}_1(t_{k,i})\| + \int_{t_{k,i}}^{t_{k,i+1}} L_f((\tau - t_k) + \|\mathrm{x}_2(\tau) - \mathrm{x}_2(t_{k,i})\| + \|\mathrm{x}_2(t_{k,i}) - \mathrm{x}_1(t_{k,i})\| \\ &\qquad + \|\mathrm{x}_1(t_{k,i}) - \mathrm{x}_1(t_k)\| + \|\mathrm{x}_1(t_k) - \mathrm{x}(t_k)\|) d\tau \end{aligned} \tag{C.9}$$

by Lipschitz of f.

By the way, for $\tau \in [t_{k,i}, t_{k,i+1})$,

$$\|\mathrm{x}_2(\tau) - \mathrm{x}_2(t_{k,i})\| \le c_2 \delta \tag{C.10}$$

for some c_2 by Lemma 19, and

$$\|\mathrm{x}_1(t_{k,i}) - \mathrm{x}_1(t_k)\| = \left\|\sum_{j=1}^{i} f(t_k, \mathrm{x}(t_k), a_j^k)(t_{k,j} - t_{k,j-1})\right\| \le c_3\delta \tag{C.11}$$

for some c_3 since $f(t_k, \mathrm{x}(t_k), a_j^k)$ is bounded.

Combining (C.10), (C.11), (C.7), and (C.9), we have

$$\|\mathrm{x}_2(t_{k,i+1}) - \mathrm{x}_1(t_{k,i+1})\| + \delta c_4 + \epsilon \le (1 + L_f(t_{k,i+1} - t_{k,i}))(\|\mathrm{x}_2(t_{k,i}) - \mathrm{x}_1(t_{k,i})\| + \delta c_4 + \epsilon) \tag{C.12}$$

for some $c_4 > 0$. By multiplying the both side of (C.12) for all $k' \ge 0, i' \ge 0$ such that $t \le t_{k',i'} \le t_{k,i}$, we have

$$\begin{aligned}
&\|\mathrm{x}_2(t_{k,i}) - \mathrm{x}_1(t_{k,i})\| \\
\le&\left[\prod_{k',i'}^{t_{k',i'} \le t_{k,i}} (1 + L_f(t_{k',i'} - t_{k',i'-1})) - 1\right](\delta c_4 + \epsilon) \\
\le&\left[\prod_{k',i'}^{t_{k',i'} \le T} (1 + L_f(t_{k',i'} - t_{k',i'-1})) - 1\right](\delta c_4 + \epsilon) \\
\le&\left[(1 + \frac{L_f T}{\bar{m}})^{\bar{m}} - 1\right](\delta c_4 + \epsilon)
\end{aligned} \tag{C.13}$$

for $\bar{m} = \sum_{k',i'} 1$ subject to $t \le t_{k',i'} \le T$: the number of the discrete time points in $[t, T]$. The last inequality in (C.13) holds since $\prod_{i=1}^n x_i \le (\sum_{i=1}^n x_i/n)^n$ for any positive x_is and $n \in \mathbb{N}$. Note that $\bar{m} \to \infty$ as $\delta \to 0$, and $\lim_{\bar{m}\to\infty}(1 + L_f T/\bar{m})^{\bar{m}} = \exp(L_f T) < \infty$. Thus, (C.13) implies

$$\|\mathrm{x}_2(t_{k,i}) - \mathrm{x}_1(t_{k,i})\| \le c_5\delta + c_6\epsilon \text{ for all } t_{k,i} \in [t, T]. \tag{C.14}$$

For $s \in (t_{k,i}, t_{k,i+1})$,

$$\begin{aligned}
\|\mathrm{x}_2(s) - \mathrm{x}_1(s)\| \le&\|\mathrm{x}_2(s) - \mathrm{x}_2(t_{k,i})\| + |\mathrm{x}_2(t_{k,i}) - \mathrm{x}_1(t_{k,i})\| + \|\mathrm{x}_1(t_{k,i}) - \mathrm{x}_1(s)\| \\
\le&c_7\delta + c_8\epsilon
\end{aligned}$$

for some $c_7, c_8 > 0$ by (C.10), (C.14), and (C.11) with little modification.

Step 4. We choose $\mathrm{x}^\epsilon = \mathrm{x}_2$ and $\alpha^\epsilon = \alpha_2$ for approximate state and control trajectories in Theorem 7. To sum up Step 1 to Step 3,

$$\|\mathrm{x}^\epsilon(s) - \mathrm{x}(s)\| \le c_9\delta + c_{10}\epsilon, \quad s \in [t, T]. \tag{C.15}$$

As $\delta \to 0$, $\epsilon \to 0$ and $\|\mathrm{x}^\epsilon - \mathrm{x}\|_{L^\infty(t,T)} \to 0$. This proves (4.22).

Step 5. Define a discrete sum of H^*:

$$S(t) \coloneqq \sum_{k=0}^{K-1} H^*(t_k, \mathrm{x}(t_k), \beta(t_k))(t_{k+1} - t_k).$$

Note that K denotes the index of the discrete time point for the terminal time: $t_K = T$. Since $H^*(s, \mathrm{x}(s), \beta(s))$ is also Riemann integrable,

$$\left| \int_t^T H^*(s, \mathrm{x}(s), \beta(s))ds - S(t) \right| < \epsilon_1 \tag{C.16}$$

for some $\epsilon_1 > 0$, and $\lim_{\delta \to 0} \epsilon_1 = 0$ where $\delta \geq \max_i \Delta t_k$ $(k = 0, ..., K-1)$.

By (C.5), (C.6), (C.15), and Lipschitz continuity of L in Assumption 1,

$$\begin{aligned} &\left| \int_{t_k}^{t_{k+1}} L(s, \mathrm{x}^\epsilon(s), \alpha^\epsilon(s)) - H^*(t_k, \mathrm{x}(t_k), \beta(t_k))ds \right| \\ =&\left| \sum_i \int_{t_{k,i}}^{t_{k,i+1}} L(s, \mathrm{x}^\epsilon(s), a_{i+1}^k) - L(t_k, \mathrm{x}(t_k), a_{i+1}^k)ds \right| \\ \leq&\Delta t_k L_L(c_{11}\delta + c_{12}\epsilon) \end{aligned} \tag{C.17}$$

for some $c_{11}, c_{12} > 0$, where L_L is the Lipschitz constant of L. Therefore,

$$\left| \int_t^T L(s, \mathrm{x}^\epsilon(s), \alpha^\epsilon(s))ds - S(t) \right| \leq T L_L(c_{11}\delta + c_{12}\epsilon). \tag{C.18}$$

By (C.16), (C.18), (C.15), and Lipschitz continuity of g in Assumption 1,

$$\left| \int_t^T H^*(s, \mathrm{x}(s), \beta(s))ds + g(\mathrm{x}(T)) - \int_t^T L(s, \mathrm{x}^\epsilon(s), \alpha^\epsilon(s))ds - g(\mathrm{x}^\epsilon(T)) \right| < c_{13}\delta + c_{14}\epsilon + \epsilon_1,$$

where $c_{13} = T L_L c_{11} + L_g c_9$, $c_{14} = T L_L c_{12} + L_g c_{10}$, where L_g is the Lipschitz constant of g. This proves (4.23). $\square$

Appendix D

Proof for Section 4.2

D.1 Proof of Theorem 10

(i) The terminal values of X_1 and X_2 are the same.

(ii) Consider a smooth function $U : (0,T) \times \mathbb{R}^n \times \mathbb{R}$. If $X_1 - U$ has a local maximum at $(t_0, x_0, z_0) \in (0,T) \times \mathbb{R}^n \times \mathbb{R}$ and $(X_1 - U)(t_0, x_0, z_0) = 0$,

$$F_1\Big(t_0, x_0, z_0, U(t_0, x_0, z_0), \frac{\partial U}{\partial t}(t_0, x_0, z_0), \frac{\partial U}{\partial x}(t_0, x_0, z_0), \frac{\partial U}{\partial z}(t_0, x_0, z_0)\Big) \geq 0, \tag{D.1}$$

Since $X_1 - U$ has a local maximum at (t_0, x_0, z_0), $\frac{\partial U}{\partial z}(t_0, x_0, z_0)$ is in $\partial_z^+ X_1(t_0, x_0, z_0)$. Then, by (4.69),

$$F_2\Big(t_0, x_0, z_0, U(t_0, x_0, z_0), \frac{\partial U}{\partial t}(t_0, x_0, z_0), \frac{\partial U}{\partial x}(t_0, x_0, z_0), \frac{\partial U}{\partial z}(t_0, x_0, z_0)\Big) \geq 0, \tag{D.2}$$

(iii) Consider a smooth function $U : (0,T) \times \mathbb{R}^n \times \mathbb{R}$. If $X_1 - U$ has a local minimum at $(t_0, x_0, z_0) \in (0,T) \times \mathbb{R}^n \times \mathbb{R}$ and $(X_1 - U)(t_0, x_0, z_0) = 0$,

$$F_1\Big(t_0, x_0, z_0, U(t_0, x_0, z_0), \frac{\partial U}{\partial t}(t_0, x_0, z_0), \frac{\partial U}{\partial x}(t_0, x_0, z_0), \frac{\partial U}{\partial z}(t_0, x_0, z_0)\Big) \leq 0, \tag{D.3}$$

Since $X_1 - U$ has a local minimum at (t_0, x_0, z_0), $\frac{\partial U}{\partial z}(t_0, x_0, z_0)$ is in $\partial_z^- X_1(t_0, x_0, z_0)$. By (4.69),

$$F_2\Big(t_0, x_0, z_0, U(t_0, x_0, z_0), \frac{\partial U}{\partial t}(t_0, x_0, z_0), \frac{\partial U}{\partial x}(t_0, x_0, z_0), \frac{\partial U}{\partial z}(t_0, x_0, z_0)\Big) \leq 0, \tag{D.4}$$

□

D.2 Proof of Lemma 10

(i) For all $y_1, y_2, y_3, y_4 \in \mathbb{R}$,

$$\max\{y_1 + y_2, y_3 + y_4\} \leq \max\{y_1, y_3\} + \max\{y_2, y_4\}. \tag{D.5}$$

(ii) Proof of (4.70) for V_1.
Let

$$J_c(\alpha,\tau) \coloneqq \max_{s\in[t,T]} c(s,\mathrm{x}(s)), \tag{D.6}$$

$$J_r(\alpha,\tau) \coloneqq \int_t^\tau L(s,\mathrm{x}(s),\alpha(s))ds + g(\tau,\mathrm{x}(\tau)). \tag{D.7}$$

$$\begin{aligned} V_1(t,x,\theta_1 z_1+\theta_2 z_2) &= \inf_{\alpha\in\mathcal{A}}\max_{\tau\in[t,T]} \max\{\theta_1 J_c(\alpha,\tau)+\theta_2 J_c(\alpha,\tau), \theta_1[J_r(\alpha,\tau)-z_1]+\theta_2[J_r(\alpha,\tau)-z_2]\} \\ &\le \inf_{\alpha\in\mathcal{A}}\max_{\tau\in[t,T]} \theta_1\max\{J_c(\alpha,\tau), J_r(\alpha,\tau)-z_1\}+\theta_2\max\{J_c(\alpha,\tau), J_r(\alpha,\tau)-z_2\}. \end{aligned} \tag{D.8}$$

The first equality is according to the distributive property of the maximum operations, and the second inequality is by (D.5). For $\alpha\in\mathcal{A}$, we use $\tau_*(\alpha)$ to denote a maximizer of the last term in (D.8). By the triangular inequality, we simplify (D.8) to

$$V_1(t,x,\theta_1 z_1+\theta_2 z_2) \le \theta_1 V_1(t,x,z_1)+\theta_2 V_1(t,x,z_2). \tag{D.9}$$

The last inequality holds by the definition of V_1 in (4.51).
(iii) Similar to (D.8),

$$V_2(t,x,\theta_1 z_1+\theta_2 z_2) \le \inf_{\alpha\in\mathcal{A}}\min_{\tau\in[t,T]} \theta_1\max\{J_c(\alpha,\tau), J_r(\alpha,\tau)-z_1\}+\theta_2\max\{J_c(\alpha,\tau), J_r(\alpha,\tau)-z_2\}. \tag{D.10}$$

Since the the last term in (D.10) is greater than or equal to $\theta_1 V_2(t,x,z_1)+\theta_2 V_2(t,x,z_2)$ by the triangular inequality, we conclude the proof. □

D.3 Proof of Lemma 11

(i) The proof of (4.71).
For $\bar{z}\ge 0$, $V_1(t,x,z+\bar{z}) \le V_1(t,x,z)$, and by the distributive property of the maximum operations,

$$\begin{aligned} V_1(t,x,z+\bar{z}) &= \inf_{\alpha\in\mathcal{A}}\max_{\tau\in[t,T]}\max\Big\{\max_{s\in[t,T]} c(s,\mathrm{x}(s))+\bar{z}, \int_t^\tau L(s,\mathrm{x}(s),u(s))ds+g(\tau,\mathrm{x}(\tau))-z\Big\}-\bar{z} \\ &\ge V_1(t,x,z)-\bar{z}. \end{aligned}$$

Thus, for $\bar{z}\ge 0$,

$$V_1(t,x,z)-\bar{z} \le V_1(t,x,z+\bar{z}) \le V_1(t,x,z), \tag{D.11}$$

and, by the same derivation, for $\bar{z} \leq 0$,

$$V_1(t,x,z) \leq V_1(t,x,z+\bar{z}) \leq V_1(t,x,z) - \bar{z}. \tag{D.12}$$

Suppose there exists $q > 0$ in $\partial_z^- V_1(t,x,z)$. Then, there exists $\epsilon > 0$ such that $V_1(t,x,z+\bar{z}) \geq V_1(t,x,z) + q\bar{z}$ for all $\bar{z} \in [-\epsilon, \epsilon]$. However, for $\bar{z} \in (0, \epsilon)$,

$$V_1(t,x,z+\bar{z}) > V_1(t,x,z). \tag{D.13}$$

This contradicts (D.11).

Suppose there exists $q < -1$ in $\partial_z^- V_1(t,x,z)$. Then, there exists $\epsilon > 0$ such that $V_1(t,x,z+\bar{z}) \geq V_1(t,x,z) + q\bar{z}$ for all $\bar{z} \in [-\epsilon, \epsilon]$. However, for $\bar{z} \in (-\epsilon, 0)$,

$$V_1(t,x,z+\bar{z}) > V_1(t,x,z) - \bar{z}. \tag{D.14}$$

This contradicts (D.12). Thus, $q \in [-1, 0]$.

With the analogous derivation for V_2, we conclude that $\partial_z^- V_2(t,x,z) \subset [-1,0]$.

(ii) The proof of (4.72).

The convexity of $V_i(t,x,z)$ $(i = 1, 2)$ stated in Lemma 10 implies that $\partial_z^+ V_i(t,x,z)$ contains a single superdifferential q if $V_i(t,x,z)$ is locally affine in z, otherwise, $\partial_z^+ V_i(t,x,z)$ is the empty set. If $V_i(t,x,z)$ is locally affine in z, it is also differentiable in z. As $\bar{z}$ converges to 0 in (D.11), we have $\partial_z^+ V_i(t,x,z) = \{\frac{\partial V_i}{\partial z}(t,x,z)\} \subset [-1,0]$. □

D.4 Proof of Lemma 12

(i) Case 1: $q = 0$.

$$\bar{H}(t,x,z,p,0) = \max_{a \in A}[-p \cdot f(t,x,a)] = \max\{p \cdot b \mid b = -f(t,x,a),\ a \in A\}. \tag{D.15}$$

Since $\{-f(t,x,a) \mid a \in A\} \subset \mathrm{co}(\{-f(t,x,a) \mid a \in A\})$,

$$\bar{H}(t,x,z,p,0) \leq \bar{H}_W(t,x,z,p,0). \tag{D.16}$$

On the other hand, let $b_* \in \arg\max[p \cdot b \mid b \in \mathrm{co}(\{-f(t,x,a) \mid a \in A\})]$. Since $\{-f(t,x,a) \mid a \in A\}$ is compact, there exists a finite number of $b_i \in \{-f(t,x,a) \mid a \in A\}$ and $\theta_i \in [0,1]$ such that $b_* = \sum_i \theta_i b_i$ and $\sum_i \theta_i = 1$. Then,

$$\bar{H}_W(t,x,z,p,0) = \sum_i \theta_i p \cdot b_i \leq \max_i\{p \cdot b_i\} \leq \max_{b \in \{-f(t,x,a) \mid a \in A\}}[p \cdot b] = \bar{H}(t,x,z,p,0). \tag{D.17}$$

The last inequality holds since all b_is are in $\{-f(t,x,a) \mid a \in A\}$. By (D.16) and (D.17), we have

$$\bar{H}(t,x,z,p,0) = \bar{H}_W(t,x,z,p,0). \tag{D.18}$$

(ii) Case 2: $q < 0$.

$$\bar{H}(t,x,z,p,q) = \max_{a\in A} -p\cdot f(t,x,a) + qL(t,x,a) = -qH\left(t,x,-\frac{p}{q}\right). \tag{D.19}$$

Since H is convex in p for each (t,x) and lower semi-continuous in p, $H^{**} \equiv H$, where $**$ indicates the Legendre-Fenchel transformation of the Legendre-Fenchel transformation. Thus, we have

$$\begin{aligned}\bar{H}_W(t,x,z,p,q) &= -q \max_{b\in \mathrm{co}(\{-f(t,x,a)\ |\ a\in A\})} [-\frac{p}{q}\cdot b - H^*(t,x,b)] \\ &= -qH^{**}\left(t,x,-\frac{p}{q}\right) = -qH\left(t,x,-\frac{p}{q}\right).\end{aligned} \tag{D.20}$$

By (D.19) and (D.20), we conclude

$$\bar{H}(t,x,z,p,q) = \bar{H}_W(t,x,z,p,q). \tag{D.21}$$

for all $q < 0$. □

Appendix E

Proof for Section 4.3

E.1 Proof of Lemma 15

This proof generalizes the proof of Lemma 1 [66], which is for the zero stage cost problem.

(i) For $b \in B(x)$, $b = -f(x, \bar{a})$ for some $\bar{a} \in A$. Then,

$$\begin{aligned} H_2^{\mathrm{TI}*}(x, b) &= \max_p -p \cdot f(x, \bar{a}) - H_2^{\mathrm{TI}}(x, p) \\ &\leq \max_p -p \cdot f(x, \bar{a}) + \min_{a \in A} p \cdot f(x, a) + L(x, a) \\ &< \infty. \end{aligned} \tag{E.1}$$

The last inequality holds since $\min_{a \in A} p \cdot f(x, a) + L(x, a) \leq p \cdot f(x, \bar{a}) + L(x, \bar{a})$ and L is finite for a fixed x.

(ii) If $b = 0$,

$$H_2^{\mathrm{TI}*}(x, b) = \max_p -H_2^{\mathrm{TI}}(x, p) \leq 0 < \infty. \tag{E.2}$$

(iii) $b \in \mathrm{co}(\{0\} \cup B(x))$

There exists a finite set of $\theta_i \in [0, 1]$ ($\sum_i \theta_i \leq 1$), $a_i \in A$ such that $b = -\sum_i \theta_i f(x, a_i)$. Since $H_2^{\mathrm{TI}*}$ is convex in b,

$$H_2^{\mathrm{TI}*}(x, b) \leq \sum_i \theta_i H_2^{\mathrm{TI}*}(x, b_i) + (1 - \sum_i \theta_i) H_2^{\mathrm{TI}*}(x, 0) < \infty$$

by (E.1) and (E.2).

(iv) $b \notin \mathrm{co}(\{0\} \cup B(x))$

For the two convex sets $\{b\}$ and $\mathrm{co}(\{0\} \cup B(x))$, by the separating hyperplane theorem [19], there exists a hyperplane ($P : \mathbb{R}^n \to \mathbb{R}$): $P(b') := p' \cdot b' + c$ such that $P(b) > 0$ but $P(b') < 0$ for all $b' \in \mathrm{co}(\{0\} \cup B(x))$. By picking $p = dp'$,

$$H_2^{\mathrm{TI}*}(x, b) \geq \max_d \min \{dp' \cdot b, \min_{a \in A} dp' \cdot (b + f(x, a)) + L(x, a)\}.$$

Since $p' \cdot b > 0$ and $p' \cdot (b + f(x, a)) > 0$ for all $a \in A$, the maximum of the right term in the above equation is attained at $d = \infty$, thus, $H_2^{\mathrm{TI}*}(x, b) = \infty$. □

E.2 Proof of Lemma 16

(i) Case 1: $q = 0$.

$$\bar{H}_2^{\mathrm{TI}}(x, z, p, 0) = \max\{0, \max_{b \in B(x)} p \cdot b\}, \tag{E.3}$$

where $B(x) = \mathrm{co}(\{-f(s, \mathrm{x}(s), a) \mid a \in A\})$. Since $B(x) \subset \mathrm{co}(\{0\} \cup B(x))$,

$$\bar{H}_2^{\mathrm{TI}}(x, z, p, 0) \leq \bar{H}_W^{\mathrm{TI}}(x, z, p, 0). \tag{E.4}$$

On the other hand, let $b_* \in \arg\max_{b \in \mathrm{co}(\{0\} \cup B(x))} p \cdot b$, then, there exists a finite number of $b_i \in B(x)$ and $\theta_i \in [0, 1]$ such that $b_* = \sum_i \theta_i b_i$ and $\sum_i \theta_i < 1$. Thus, we have

$$\begin{aligned} \bar{H}_W^{\mathrm{TI}}(x, z, p, 0) &= \sum_i \theta_i p \cdot b_i \leq \max\{0, \max_i p \cdot b_i\} \\ &\leq \bar{H}_2^{\mathrm{TI}}(x, z, p, 0). \end{aligned} \tag{E.5}$$

Combining (E.4) and (E.5), we have

$$\bar{H}_W^{\mathrm{TI}}(x, z, p, 0) = \bar{H}_2^{\mathrm{TI}}(x, z, p, 0). \tag{E.6}$$

(ii) Case 2: $q < 0$.

$$\bar{H}_2^{\mathrm{TI}}(x, z, p, q) = -q \max\left\{0, H\left(x, -\frac{p}{q}\right)\right\} \tag{E.7}$$

and, by the convexity of H_2^{TI} in p, $H_2^{\mathrm{TI}**} \equiv H_2^{\mathrm{TI}}$, where $**$ indicates the Legendre-Fenchel transformation of the Legendre-Fenchel transformation. Then,

$$\bar{H}_W^{\mathrm{TI}}(x, z, p, q) = -q H_2^{\mathrm{TI}}\left(x, -\frac{p}{q}\right). \tag{E.8}$$

By combining (E.7), (E.8), and (4.100), we conclude the proof. □

Bibliography

[1] Joshua Achiam et al. "Constrained policy optimization". In: *International Conference on Machine Learning*. 2017, pp. 22–31.

[2] Ravi P Agarwal, Maria Meehan, and Donal O'regan. *Fixed point theory and applications*. Vol. 141. Cambridge university press, 2001.

[3] Walter Alt, Ursula Felgenhauer, and Martin Seydenschwanz. "Euler discretization for a class of nonlinear optimal control problems with control appearing linearly". In: *Computational Optimization and Applications* 69.3 (2018), pp. 825–856.

[4] Albert Altarovici, Olivier Bokanowski, and Hasnaa Zidani. "A general Hamilton-Jacobi framework for non-linear state-constrained control problems". In: *ESAIM: Control, Optimisation and Calculus of Variations* 19.2 (2013), pp. 337–357.

[5] Eitan Altman. *Constrained Markov decision processes*. Vol. 7. CRC Press, 1999.

[6] Aaron D Ames et al. "Control barrier function based quadratic programs for safety critical systems". In: *IEEE Transactions on Automatic Control* 62.8 (2016), pp. 3861–3876.

[7] Summer Atkins et al. "Solving Singular Control Problems in Mathematical Biology, Using PASA". In: *arXiv preprint arXiv:2010.06744* (2020).

[8] Jean-Pierre Aubin and Hélène Frankowska. "The viability kernel algorithm for computing value functions of infinite horizon optimal control problems". In: *Journal of mathematical analysis and applications* 201.2 (1996), pp. 555–576.

[9] Somil Bansal and Claire Tomlin. "Deepreach: A deep learning approach to high-dimensional reachability". In: *arXiv preprint arXiv:2011.02082* (2020).

[10] Somil Bansal et al. "Hamilton-Jacobi reachability: A brief overview and recent advances". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 2242–2253.

[11] Somil Bansal et al. "Safe sequential path planning under disturbances and imperfect information". In: *American Control Conference (ACC), 2017*. IEEE. 2017, pp. 5550–5555.

[12] Martino Bardi and Lawrence C Evans. "On Hopf's formulas for solutions of Hamilton-Jacobi equations". In: *Nonlinear Analysis: Theory, Methods & Applications* 8.11 (1984), pp. 1373–1381.

[13] EN Barron and H Ishii. "The Bellman equation for minimizing the maximum cost". In: *Nonlinear Analysis: Theory, Methods & Applications* 13.9 (1989), pp. 1067–1090.

[14] Vincenzo Basco and Hélène Frankowska. "Lipschitz continuity of the value function for the infinite horizon optimal control problem under state constraints". In: *Trends in Control Theory and Partial Differential Equations* (2019), pp. 17–38.

[15] Toni Barjas Blanco, Mark Cannon, and Bart De Moor. "On efficient computation of low-complexity controlled invariant sets for uncertain linear systems". In: *International journal of Control* 83.7 (2010), pp. 1339–1346.

[16] Olivier Bokanowski, Anya Désilles, and Hasnaa Zidani. "Relationship between maximum principle and dynamic programming in presence of intermediate and final state constraints". In: *ESAIM: Control, Optimisation and Calculus of Variations* 27 (2021), p. 91.

[17] Olivier Bokanowski, Athena Picarelli, and Hasnaa Zidani. "State-constrained stochastic optimal control problems via reachability approach". In: *SIAM Journal on Control and Optimization* 54.5 (2016), pp. 2568–2593.

[18] J Frédéric Bonnans and Adriano Festa. "Error estimates for the Euler discretization of an optimal control problem with first-order state constraints". In: *SIAM Journal on Numerical Analysis* 55.2 (2017), pp. 445–471.

[19] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[20] Stephen Boyd et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends® in Machine learning* 3.1 (2011), pp. 1–122.

[21] Arthur E Bryson and Yu-Chi Ho. *Applied optimal control: optimization, estimation, and control*. Routledge, 2018.

[22] John Charles Butcher and Gerhard Wanner. "Runge-Kutta methods: some historical notes". In: *Applied Numerical Mathematics* 22.1-3 (1996), pp. 113–151.

[23] Marco Caponigro et al. "Regularization of chattering phenomena via bounded variation controls". In: *IEEE Transactions on Automatic Control* 63.7 (2018), pp. 2046–2060.

[24] Italo Capuzzo-Dolcetta and P-L Lions. "Hamilton-Jacobi equations with state constraints". In: *Transactions of the American mathematical society* 318.2 (1990), pp. 643–683.

[25] André M Carvalho and João S Sequeira. "Approximating Viability Kernels of High-Dimensional Linear Systems". In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 4289–4294.

[26] Margaret P Chapman et al. "Reachability Analysis as a Design Tool for Stormwater Systems". In: *2018 IEEE Conference on Technologies for Sustainability (SusTech)*. IEEE. 2018, pp. 1–8.

[27] Mo Chen, Sylvia Herbert, and Claire J Tomlin. "Exact and efficient Hamilton-Jacobi-based guaranteed safety analysis via system decomposition". In: *arXiv preprint arXiv:1609.05248* (2016).

[28] Mo Chen et al. "Decomposition of reachable sets and tubes for a class of nonlinear systems". In: *IEEE Transactions on Automatic Control* 63.11 (2018), pp. 3675–3688.

[29] Mo Chen et al. "Fastrack: a modular framework for real-time motion planning and guaranteed safe tracking". In: *IEEE Transactions on Automatic Control* 66.12 (2021), pp. 5861–5876.

[30] Mo Chen et al. "Optimal control helper toolbox". In: `http://github.com/HJReachability/helperOC`.

[31] Mo Chen et al. "Reachability-based safety and goal satisfaction of unmanned aerial platoons on air highways". In: *Journal of Guidance, Control, and Dynamics* 40.6 (2017), pp. 1360–1373.

[32] Yuxiao Chen and Necmiye Ozay. "Data-driven computation of robust control invariant sets with concurrent model selection". In: *IEEE Transactions on Control Systems Technology* (2021).

[33] Jason J Choi et al. "Robust control barrier-value functions for safety-critical control". In: *arXiv preprint arXiv:2104.02808* (2021).

[34] Yat Tin Chow et al. "Algorithm for overcoming the curse of dimensionality for state-dependent hamilton-jacobi equations". In: *Journal of Computational Physics* 387 (2019), pp. 376–409.

[35] Yinlam Chow et al. "Risk-constrained reinforcement learning with percentile risk criteria". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 6070–6120.

[36] Christian G Claudel and Alexandre M Bayen. "Lax–Hopf based incorporation of internal boundary conditions into Hamilton–Jacobi equation. Part I: Theory". In: *IEEE Transactions on Automatic Control* 55.5 (2010), pp. 1142–1157.

[37] Michael G Crandall and P-L Lions. "Two approximations of solutions of Hamilton-Jacobi equations". In: *Mathematics of computation* 43.167 (1984), pp. 1–19.

[38] Jérôme Darbon and Stanley Osher. "Algorithms for overcoming the curse of dimensionality for certain Hamilton–Jacobi equations arising in control theory and elsewhere". In: *Research in the Mathematical Sciences* 3.1 (2016), p. 19.

[39] Alexander Domahidi et al. "Efficient interior point methods for multistage problems arising in receding horizon control". In: *2012 IEEE 51st IEEE conference on decision and control (CDC)*. IEEE. 2012, pp. 668–674.

[40] Peter Dorato. "A historical review of robust control". In: *IEEE Control Systems Magazine* 7.2 (1987), pp. 44–47.

[41] Robert James Elliott and Nigel John Kalton. *The existence of value in differential games*. Vol. 126. American Mathematical Soc., 1972.

[42] Lawrence C. Evans. *Partial differential equations*. American Mathematical Society, 2010.

[43] Georgios E Fainekos, Savvas G Loizou, and George J Pappas. "Translating temporal logic to controller specifications". In: *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE. 2006, pp. 899–904.

[44] Jianqing Fan et al. "A theoretical analysis of deep Q-learning". In: *Learning for Dynamics and Control*. PMLR. 2020, pp. 486–489.

[45] Jaime F Fisac et al. "Bridging hamilton-jacobi safety analysis and reinforcement learning". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8550–8556.

[46] Jaime F Fisac et al. "Reach-avoid problems with time-varying dynamics, targets and constraints". In: *Proceedings of the 18th international conference on hybrid systems: computation and control*. ACM. 2015, pp. 11–20.

[47] H Frankowska and RB Vinter. "Existence of neighboring feasible trajectories: applications to dynamic programming for state-constrained optimal control problems". In: *Journal of Optimization Theory and Applications* 104.1 (2000), pp. 20–40.

[48] Hélene Frankowska and Sławomir Plaskacz. "Semicontinuous solutions of Hamilton-Jacobi-Bellman equations with degenerate state constraints". In: *Journal of Mathematical Analysis and Applications* 251.2 (2000), pp. 818–838.

[49] Nidhal Gammoudi and Hasnaa Zidani. "A differential game control problem with state constraints". In: *Mathematical Control & Related Fields* (2022).

[50] Joseph D Gleason, Abraham P Vinod, and Meeko MK Oishi. "Underapproximation of reach-avoid sets for discrete-time stochastic systems via lagrangian methods". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 4283–4290.

[51] Tom Goldstein and Stanley Osher. "The split Bregman method for L1-regularized problems". In: *SIAM journal on imaging sciences* 2.2 (2009), pp. 323–343.

[52] Henry Gouk et al. "Regularisation of neural networks by enforcing lipschitz continuity". In: *Machine Learning* 110.2 (2021), pp. 393–416.

[53] Jacopo Guanetti, Yeojun Kim, and Francesco Borrelli. "Control of connected and automated vehicles: State of the art and future challenges". In: *Annual reviews in control* 45 (2018), pp. 18–40.

[54] Tuomas Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International conference on machine learning*. PMLR. 2018, pp. 1861–1870.

[55] Richard F Hartl, Suresh P Sethi, and Raymond G Vickson. "A survey of the maximum principles for optimal control problems with state constraints". In: *SIAM review* 37.2 (1995), pp. 181–218.

[56] Eberhard Hopf. "Generalized solutions of non-linear equations of first order". In: *Journal of Mathematics and Mechanics* 14.6 (1965), pp. 951–973.

[57] Kai-Chieh Hsu et al. "Safety and liveness guarantees through reach-avoid reinforcement learning". In: *arXiv preprint arXiv:2112.12288* (2021).

[58] Inseok Hwang, Jegyom Kim, and Claire Tomlin. "Protocol-based conflict resolution for air traffic control". In: *Air Traffic Control Quarterly* 15.1 (2007), pp. 1–34.

[59] Hitoshi Ishii and Shigeaki Koike. "A new formulation of state constraint problems for first-order PDEs". In: *SIAM Journal on Control and Optimization* 34.2 (1996), pp. 554–571.

[60] Mrdjan Jankovic. "Robust control barrier functions for constrained stabilization of nonlinear systems". In: *Automatica* 96 (2018), pp. 359–367.

[61] Milan Korda, Didier Henrion, and Colin N Jones. "Convex computation of the maximum controlled invariant set for polynomial control systems". In: *SIAM Journal on Control and Optimization* 52.5 (2014), pp. 2944–2969.

[62] Aviral Kumar et al. "Conservative Q-learning for offline reinforcement learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1179–1191.

[63] Benoit Landry et al. "Reach-avoid problems via sum-or-squares optimization and dynamic programming". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 4325–4332.

[64] Peter D Lax. "Hyperbolic systems of conservation laws II". In: *Communications on pure and applied mathematics* 10.4 (1957), pp. 537–566.

[65] Donggun Lee and Claire J Tomlin. "A computationally efficient Hamilton-Jacobi-based formula for state-constrained optimal control problems". In: *arXiv preprint arXiv:2106.13440* (2021).

[66] Donggun Lee and Claire J Tomlin. "A Hopf-Lax formula in Hamilton-Jacobi analysis of reach-avoid problems". In: *IEEE Control Systems Letters* 5.3 (2020), pp. 1055–1060.

[67] Donggun Lee and Claire J Tomlin. "Hamilton-Jacobi equations for two classes of state-constrained zero-sum games". In: *arXiv preprint arXiv:2106.15006* (2021).

[68] Donggun Lee and Claire J Tomlin. "Lax Formulae for Efficiently Solving Two Classes of State-Constrained Optimal Control Problems". In: *arXiv preprint arXiv:2109.00140* (2021).

[69] Donggun Lee et al. "Hamilton-Jacobi Formulation for State-Constrained Optimal Control and Zero-Sum Game Problems". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 1078–1085.

[70] Jingqi Li et al. "Infinite-Horizon Reach-Avoid Zero-Sum Games via Deep Reinforcement Learning". In: *arXiv preprint arXiv:2203.10142* (2022).

[71] Lars Lindemann and Dimos V Dimarogonas. "Control barrier functions for signal temporal logic tasks". In: *IEEE control systems letters* 3.1 (2018), pp. 96–101.

[72] P-L Lions and J-C Rochet. "Hopf formula and multitime Hamilton-Jacobi equations". In: *Proceedings of the American Mathematical Society* 96.1 (1986), pp. 79–84.

[73] John Lygeros. "On reachability and minimum cost optimal control". In: *Automatica* 40.6 (2004), pp. 917–927.

[74] Kostas Margellos and John Lygeros. "Hamilton-Jacobi formulation for reach-avoid differential games". In: *IEEE Transactions on Automatic Control* 56.8 (2011), pp. 1849–1861.

[75] Ian M Mitchell. "The flexible, extensible and efficient toolbox of level set methods". In: *Journal of Scientific Computing* 35.2 (2008), pp. 300–329.

[76] Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games". In: *IEEE Transactions on Automatic Control* 50.7 (2005), pp. 947–957.

[77] Ian M Mitchell, Jacob Budzis, and Andriy Bolyachevets. "Invariant, viability and discriminating kernel under-approximation via zonotope scaling". In: *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*. 2019, pp. 268–269.

[78] Ian M Mitchell and Jeremy A Templeton. "A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems". In: *International Workshop on Hybrid Systems: Computation and Control*. Springer. 2005, pp. 480–494.

[79] Ian M. Mitchell and Claire J. Tomlin. "Overapproximating reachable sets by Hamilton-Jacobi projections". In: *J. Scientific Computing* 19.1-3 (2003), pp. 323–346. DOI: 10.1023/A:1025364227563.

[80] Volodymyr Mnih et al. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).

[81] Khan Muhammad et al. "Deep learning for safe autonomous driving: Current challenges and future directions". In: *IEEE Transactions on Intelligent Transportation Systems* (2020).

[82] Remi Munos, Leemon C Baird, and Andrew W Moore. "Gradient descent approaches to neural-net-based solutions of the Hamilton-Jacobi-Bellman equation". In: *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*. Vol. 3. IEEE. 1999, pp. 2152–2157.

[83] Quan Nguyen and Koushil Sreenath. "Robust safety-critical control for dynamic robotics". In: *IEEE Transactions on Automatic Control* (2021).

[84] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[85] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Vol. 153. Springer-Verlag, 2003. DOI: 10.1007/b98879.

[86] Antoine Oustry, Matteo Tacchi, and Didier Henrion. "Inner approximations of the maximal positively invariant set for polynomial dynamical systems". In: *IEEE Control Systems Letters* 3.3 (2019), pp. 733–738.

[87] Sasa V Rakovic et al. "Invariant approximations of the minimal robust positively invariant set". In: *IEEE Transactions on automatic control* 50.3 (2005), pp. 406–410.

[88] Saša V Rakovic and Miroslav Baric. "Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks". In: *IEEE Transactions on Automatic Control* 55.7 (2010), pp. 1599–1614.

[89] Alexander Robey et al. "Learning control barrier functions from expert demonstrations". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 3717–3724.

[90] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[91] Matthias Rungger and Paulo Tabuada. "Computing robust controlled invariant sets of linear systems". In: *IEEE Transactions on Automatic Control* 62.7 (2017), pp. 3665–3670.

[92] Jhanani Selvakumar and Efstathios Bakolas. "Feedback strategies for a reach-avoid game with a single evader and multiple pursuers". In: *IEEE transactions on cybernetics* (2019).

[93] James A Sethian. "A fast marching level set method for monotonically advancing fronts". In: *Proceedings of the National Academy of Sciences* 93.4 (1996), pp. 1591–1595.

[94] Chi-Wang Shu and Stanley Osher. "Efficient implementation of essentially non-oscillatory shock-capturing schemes". In: *Journal of computational physics* 77.2 (1988), pp. 439–471.

[95] David Silver et al. "Deterministic policy gradient algorithms". In: 2014.

[96] Sumeet Singh et al. "Robust Tracking with Model Mismatch for Fast and Safe Planning: an SOS Optimization Approach". In: *arXiv preprint arXiv:1808.00649* (2018).

[97] Halil Mete Soner. "Optimal control with state-space constraint I". In: *SIAM Journal on Control and Optimization* 24.3 (1986), pp. 552–561.

[98] Halil Mete Soner. "Optimal control with state-space constraint. II". In: *SIAM journal on control and optimization* 24.6 (1986), pp. 1110–1122.

[99] Eric Squires, Pietro Pierpaoli, and Magnus Egerstedt. "Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance". In: *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2018, pp. 1656–1661.

[100] Sean Summers et al. "A stochastic reach-avoid problem with random obstacles". In: *Proceedings of the 14th international conference on Hybrid systems: computation and control*. 2011, pp. 251–260.

[101] Andrew Taylor et al. "Learning for safety-critical control with control barrier functions". In: *Learning for Dynamics and Control*. PMLR. 2020, pp. 708–717.

[102] Brijen Thananjeyan et al. "Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3612–3619.

[103] Claire Tomlin, George J Pappas, and Shankar Sastry. "Conflict resolution for air traffic management: A study in multiagent hybrid systems". In: *IEEE Transactions on automatic control* 43.4 (1998), pp. 509–521.

[104] Rene Vidal et al. "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation". In: *IEEE transactions on robotics and automation* 18.5 (2002), pp. 662–669.

[105] Li Wang, Aaron D Ames, and Magnus Egerstedt. "Safety barrier certificates for collisions-free multirobot systems". In: *IEEE Transactions on Robotics* 33.3 (2017), pp. 661–674.

[106] Ivan Yegorov and Peter M Dower. "Perspectives on characteristics based curse-of-dimensionality-free numerical approaches for solving Hamilton-Jacobi equations". In: *Applied Mathematics & Optimization* (2018), pp. 1–49.

[107] Shuyou Yu et al. "Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems". In: *Systems & Control Letters* 62.2 (2013), pp. 194–200.

[108] Michail I Zelikin and Vladimir F Borisov. *Theory of chattering control: with applications to astronautics, robotics, economics, and engineering*. Springer Science & Business Media, 2012.

[109] Fuchen Zhang et al. "Estimating the ultimate bound and positively invariant set for a synchronous motor and its application in chaos synchronization". In: *Chaos, Solitons & Fractals* 44.1-3 (2011), pp. 137–144.

[110] Zhengyuan Zhou et al. "A general, open-loop formulation for reach-avoid games". In: *2012 IEEE 51st IEEE conference on decision and control (CDC)*. IEEE. 2012, pp. 6501–6506.