

---

**Control Algorithms for Soft Walls<sup>1</sup>**  
by Adam Cataldo

---

**Research Project**

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

**Committee:**

---

Edward Lee  
Research Advisor

---

Date

\* \* \* \* \*

---

Shankar Sastry  
Second Reader

---

Date

---

<sup>1</sup>January 21, 2004  
Technical Memorandum UCB/ERL M03/42  
University of California, Berkeley, CA 94720

# Contents

<b>1</b>	<b>Introduction to Soft Walls</b>	<b>3</b>
1.1	Soft Walls . . . . .	3
1.2	System Model . . . . .	4
<b>2</b>	<b>Control from an Implicit Surface Function</b>	<b>7</b>
2.1	Collision-Avoidance Verification with the Hamilton-Jacobi-Isaacs Approach . . . . .	7
2.1.1	Verification Methodology . . . . .	7
2.1.2	The Cost Function . . . . .	9
2.1.3	Games of Finite Duration . . . . .	13
2.1.4	The Hamilton-Jacobi-Isaacs Approach . . . . .	14
2.1.5	Safe Controllers . . . . .	16
2.2	Implementation . . . . .	16
2.2.1	A Note on Our Dynamics Equation . . . . .	19
<b>3</b>	<b>Other Soft Walls Control Approaches</b>	<b>21</b>
3.1	Solution for Half-Space No-Fly Zone . . . . .	21
3.2	A Geometric Approach . . . . .	28
<b>4</b>	<b>Conclusions and Future Work</b>	<b>30</b>

# Chapter 1

## Introduction to Soft Walls

### 1.1 Soft Walls

In response to the September 11, 2001 hijackings, Edward Lee proposed a new flight-control technology called Soft Walls [9]. The Soft Walls strategy is to store a 3D database of “no-fly zones”, or restricted airspace, on-board each aircraft and enforce these no-fly zones using an avionics control system. Each aircraft will have its own Soft Walls system. Also, the database will require a digital signature to update the no-fly zones so that the system is non-hackable.

Soft Walls is not an autonomous control strategy. That is, the controller does not remove pilot input when the aircraft approaches a no-fly zone. Instead the controller adds a bias to the pilot input, and it never removes all pilot authority. A pilot who approaches a no-fly zone and holds steady will be turned away from the no-fly zone until it is safe to let the aircraft fly straight. A pilot who chooses to turn away faster can do so. A pilot who tries to fly into the no-fly zone will be unsuccessful. Through this, Soft Walls will maximize pilot authority subject to the constraint that no-fly zones are enforced. (The system may also include comfort constraints.) This will give the pilot more maneuverability in an emergency. Figure 1.1 shows how pilot decisions affect the control system. See [10] for some objections to this strategy.

Even though Soft Walls is not an autonomous control strategy, it is related to autonomous aircraft control in that it is a collision avoidance problem. This lets us apply collision avoidance techniques similar to those used for

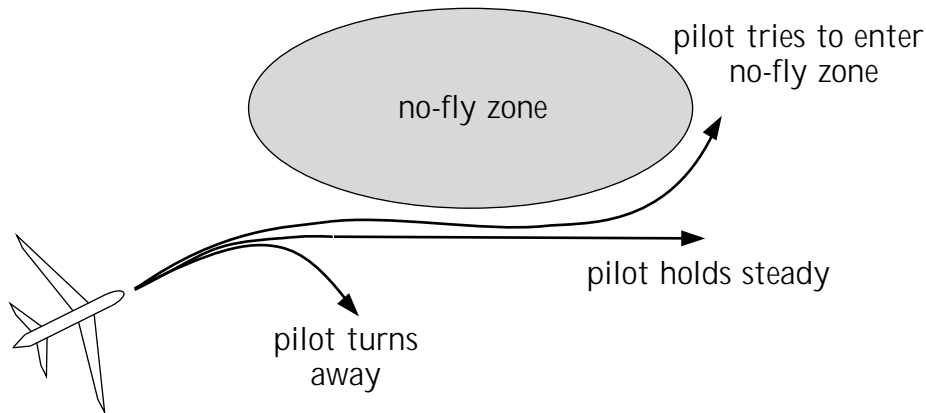


Figure 1.1: A pilot who approaches a no-fly zone and holds steady will be turned away from the no-fly zone until it is safe to let the aircraft fly straight. A pilot who chooses to turn away faster can do so. A pilot who tries to fly into the no-fly zone will be unsuccessful.

autonomous flight-path planners that keep two autonomous aircraft from colliding. This paper focuses on the collision avoidance algorithms. We consider several controllers we have developed for simulation.

## 1.2 System Model

We seek a system model that tracks the aircraft's state as it changes with time. To match reality, this model should depend on the pilot's actions as well as the actions of the control system we design. We want a general model that does not depend on the particular dynamics of the aircraft. This helps us apply our algorithms to different aircraft.

We let  $[A \rightarrow B]$  represent the set of functions mapping  $A$  to  $B$ , where  $A$  and  $B$  are sets. Then  $[A \xrightarrow{meas} B]$  represents the set of *measurable* functions from  $A$  to  $B$ . We then define a system by the tuple

$$S = (I, X, U, \mathcal{U}, D, \mathcal{D}, f, \xi, \mathcal{X}), \quad (1.1)$$

in which

- $I \subseteq \mathbb{R}$  is an interval of time, and  $t \in I$  is a particular time. We assume that  $I$  is a connected subset of  $\mathbb{R}$ .
- $X \subseteq \mathbb{R}^n$  is the state space.
- $U \subset \mathbb{R}^m$ , where  $U$  is closed and bounded, is the control space.
- $\mathcal{U} = [I \xrightarrow{meas} U]$  is set of control evolutions over the interval of time  $I$ . If  $u \in \mathcal{U}$  is a control evolution, and if  $t \in I$ , then  $u(t) \in U$  is the control input at time  $t$ .
- $D \subset \mathbb{R}^p$ , where  $D$  is closed and bounded, is the disturbance space.
- $\mathcal{D} = [I \xrightarrow{meas} D]$  is set of disturbance evolutions over the interval of time  $I$ . If  $d \in \mathcal{D}$  is a disturbance evolution, and if  $t \in I$ , then  $d(t) \in D$  is the control input at time  $t$ .
- $f : X \times U \times D \rightarrow \mathbb{R}^n$  is a bounded, Lipschitz continuous function that describes the flow of the system. That is,  $f(x(t), u(t), d(t))$  is the rate of change in state at time  $t$ .
- $C \subset 2^I$  is the set of all closed, bounded, and connected subsets of  $I$ . If  $\Omega \in C$ , then  $\exists a, b \in I$  such that  $\Omega = [a, b]$ .
- $\xi : C \times X \times \mathcal{U} \times \mathcal{D} \rightarrow X$  is a trajectory. If  $y \in X$ ,  $u \in \mathcal{U}$ ,  $d \in \mathcal{D}$ , and  $[t_1, t_2] \in C$ , with  $t_2 \geq t_1$ , then

$$\xi([t_1, t_2], y, u, d) = y + \int_{t_1}^{t_2} f(\xi([t_1, \tau], y, u, d), u(\tau), d(\tau)) d\tau. \quad (1.2)$$

- $\mathcal{X} \subset [I \rightarrow X]$  is the set of all state evolutions over the interval of time  $I$ . A state evolution  $x$  is in  $\mathcal{X}$  there exists a  $u \in \mathcal{U}$  and  $d \in \mathcal{D}$  such that  $\dot{x}(t) = f(x(t), u(t), d(t))$  for all  $t \in I$ . If  $x(t_1) = y$  for  $t_1 \in I$ , then  $x(t_2) = \xi([t_1, t_2], f, y, u, d)$  for all  $t_2 \in I$  satisfying  $t_2 \geq t_1$ .

In Soft Walls  $x \in \mathcal{X}$  is the trajectory of the aircraft.  $d \in \mathcal{D}$  is the action of the pilot, while  $u \in \mathcal{U}$  is the action of the Soft Walls controller. Note that  $U$  and  $D$  are compact sets. This means that at each time  $t \in I$ , the control input  $u(t)$  and disturbance input  $d(t)$  are bounded. These bounds are constraints on the types of inputs the system will accept. These bounds may come from safety constraints or physical limitations of the aircraft.  $f$

describes the dynamics equation of the aircraft. The constraints that  $f$  is bounded and Lipschitz continuous guarantees that for each initial condition and  $u \in \mathcal{U}$  and  $d \in D$ , the system  $S$  gives a unique trajectory  $x \in \mathcal{X}$ .

In this paper, we focus on a particular system  $S$ . Here  $X = \mathbb{R}^3$ . If  $t \in I$  we let  $x(t) = (x_{p1}(t), x_{p2}(t), x_h(t))^T$  represent the state at time  $t$ . We define  $f$  as

$$f \left( \begin{pmatrix} x_{p1}(t) \\ x_{p2}(t) \\ x_h(t) \end{pmatrix}, u(t), d(t) \right) = \begin{pmatrix} s \cos x_h(t) \\ s \sin x_h(t) \\ u(t) + d(t) \end{pmatrix}. \quad (1.3)$$

For a picture of the state space, see Figure 1.2.

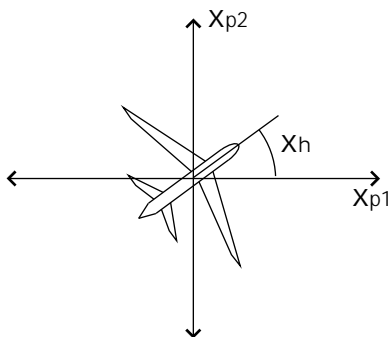


Figure 1.2: The state space for the single aircraft is the aircraft's position  $(x_{p1}(t), x_{p2}(t))$  and heading  $x_h(t)$ . We use this model for our Soft Walls-controller implementation.

Here we assume the aircraft is travelling at a fixed speed  $s > 0$ , so the aircraft can only turn or go straight. The input is the rate of change in heading angle. We assume that, due to safety constraints, the aircraft allows a maximum rate of change in heading of  $M > 0$ . We will only let the pilot give inputs in this range, so  $D = [-M, M]$ . At each  $t \in I$ , so that the control input can force the aircraft to turn away from the no-fly zone even if the pilot input commands the aircraft to turn into the no-fly zone, we let  $|u(t)| > M$ . So that even when the maximum control input is applied, the pilot still has some authority, we need  $|u(t)| < 2M$ . We choose  $U = [-1.5M, 1.5M]$ . We also limit the value of  $u(t) + d(t)$  to the range  $[-M, M]$ , but we do not include this in our dynamics model. We will explain why in Chapter 2.

# Chapter 2

## Control from an Implicit Surface Function

### 2.1 Collision-Avoidance Verification with the Hamilton-Jacobi-Isaacs Approach

In this section we present a multi-agent collision-avoidance approach for continuous systems. This is an adaptation of the work of [16] and [15]. We restrict ourselves to nonlinear continuous systems, whereas these papers solve the problem for nonlinear hybrid systems. We use a different *information pattern*, which we will explain in Section 2.1.1.

#### 2.1.1 Verification Methodology

Given our system  $S$ , we now derive a method to generate the unsafe region of the state space  $X$ . The target set  $T$  is an illegal region of the state space, and a “conflict” occurs at time  $t$  if  $x(t) \in T$ . For Soft Walls  $T$  is the no-fly zone. Here,  $T$  is a target we want to avoid; the word target comes from missile guidance problems [8]. We let  $\text{Pre} : 2^X \rightarrow 2^X$  be the *predecessor function*. This function will have the property that  $\text{Pre}(T)$ , which we call the *backwards reachable set* of  $T$ , is the unsafe region of the state space. Given  $T$ , for any control evolution and for each  $y \in \text{Pre}(T)$ , there exists a disturbance evolution which causes a trajectory starting at  $y$  to eventually enter  $T$ . Trivially, if  $y \in T$ , then  $y \in \text{Pre}(T)$ . If a trajectory starts outside this subset of the state space, then there exists a control evolution that keeps

the state outside of  $T$ . We will give a definition of Pre shortly, but we first need some more definitions.

We can describe this collision-avoidance situation as a two-player game. Here Player I chooses  $d \in \mathcal{D}$ , and Player II chooses  $u \in \mathcal{U}$ . In Soft Walls Player I is the pilot, and Player II is the Soft Walls controller. We say Player I *wins* the game if  $x(t) \in T$  for some  $t \in I$ . We say Player II wins the game if  $x(t) \notin T$  for all  $t \in I$ . The set  $\text{Pre}(T)$  is the set of states at which Player I can choose  $d$  to guarantee a collision, irrespective of the  $u$  Player II chooses. The set  $X \setminus \text{Pre}(T)$  is the set of states at which, for each  $d$  Player I chooses, Player II can choose  $u$  to avoid a collision for all future time.

We assume that at each time  $t$  Player II knows the pilot's choice of  $d(\tau)$  for all  $\tau \in I$  with  $\tau \leq t$ . Player II does not know the value of  $d(\tau)$  for  $\tau > t$ , so we call this a *nonanticipative strategy*. We also assume that each time  $t$  both players know the state  $x(\tau)$  for all  $\tau \in I$  with  $\tau \leq t$ . For  $t \in I$  we let  $I_r : I \rightarrow 2^I$  be

$$I_r(t) = \{\tau \in I \mid \tau \in (-\infty, t]\}. \quad (2.1)$$

We let the set of nonanticipative control strategies be given by

$$\begin{aligned} \Upsilon = \{ & \gamma \in [\mathcal{D} \rightarrow \mathcal{U}] \mid d_1 \in \mathcal{D}, d_2 \in \mathcal{D}, \forall s \in I_r(t), d_1(s) = d_2(s) \Rightarrow \\ & \forall s \in I_r(t), (\gamma(d_1))(s) = (\gamma(d_2))(s)\}. \end{aligned} \quad (2.2)$$

The *information pattern* tells us which information available to each player. The information pattern we choose describes the Soft Walls controller, since the system will have access to the pilot's actions, while the pilot will not be allowed to monitor the control system actions, only the systems effects on the aircraft. In [15], Player I is assumed to be an unknown disturbance, such as wind, the opposite information pattern is used. For more on information patterns, see [1]

Formally,

$$\text{Pre}(T) = \{y \in X \mid \forall \gamma \in \Upsilon, \exists d \in \mathcal{D}, \exists \tau \in (-\infty, 0], \xi([\tau, 0], y, \gamma(d), d) \in T\}. \quad (2.3)$$

We let the *avoid function*  $\text{Avoid} : 2^X \rightarrow 2^X$  output the safe region of the state space. That is,  $\text{Avoid}(T) = X \setminus \text{Pre}(T)$ .

**Lemma 2.1.** *For all  $T \subseteq X$ ,*

$$\text{Avoid}(T) = \{y \in X \mid \exists \gamma \in \Upsilon, \forall d \in \mathcal{D}, \forall \tau \in (-\infty, 0], \xi([\tau, 0], y, \gamma(d), d) \notin T\} \quad (2.4)$$



*Proof.*

$$\begin{aligned}
\text{Avoid}(T) &= \{y \in X \mid \neg(\forall \gamma \in \Upsilon, \exists d \in \mathcal{D}, \exists \tau \in (-\infty, 0], \xi([\tau, 0], y, \gamma(d), d) \in T)\} \\
&= \{y \in X \mid \exists \gamma \in \Upsilon, \neg(\exists d \in \mathcal{D}, \exists \tau \in (-\infty, 0], \xi([\tau, 0], y, \gamma(d), d) \in T)\} \\
&= \{y \in X \mid \exists \gamma \in \Upsilon, \forall d \in \mathcal{D}, \neg(\exists \tau \in (-\infty, 0], \xi([\tau, 0], y, \gamma(d), d) \in T)\} \\
&= \{y \in X \mid \exists \gamma \in \Upsilon, \forall d \in \mathcal{D}, \forall \tau \in (-\infty, 0], \neg(\xi([\tau, 0], y, \gamma(d), d) \in T)\} \\
&= \{y \in X \mid \exists \gamma \in \Upsilon, \forall d \in \mathcal{D}, \forall \tau \in (-\infty, 0], \xi([\tau, 0], y, \gamma(d), d) \notin T\}
\end{aligned}$$

Here we carried out negation of a predicate at each step.  $\square$

## 2.1.2 The Cost Function

Given our target set, we want a way to compute  $\text{Pre}(T)$ . Our first step is to rethink how we characterize  $T$ . If  $T$  is a connected, bounded, open set, and we can create a differentiable function  $l : X \rightarrow \mathbb{R}$  such that  $x \in T$  if and only if  $l(x) < 0$ , then we have one such characterization. This will be possible whenever  $T$  has a simple shape, like a hypersphere or hypercylinder. We call  $l$  the *terminal cost function*. We then define our *cost function*  $V : X \times \mathcal{U} \times \mathcal{D} \rightarrow \text{reals}$  as

$$V(y, u, d) = \inf_{\tau \in (-\infty, 0]} l(\xi([\tau, 0], y, u, d)). \quad (2.5)$$

As long as  $l$  is bounded below, the infimum will always exist.

Given a set  $A \subset \mathbb{R}^a$ , we let  $C^n(A)$  be the set of all functions in  $[A \rightarrow \mathbb{R}]$ , that has a continuous  $n^{\text{th}}$  derivative for all  $a \in A$ . That is

$$C^n(A) = \left\{ g \in [A \rightarrow \mathbb{R}] \mid \forall a \in A, \frac{d^n}{da^n} g(a) \text{ exists} \right\} \quad (2.6)$$

From this point on, we assume  $l \in C^1(X)$  and bounded below, and  $l(y) < 0$  if and only if  $y \in T$ , so that  $V$  exists. In this case,  $V$  has the following important property:

**Lemma 2.2.** *Given  $y \in X$ ,  $u \in \mathcal{U}$ ,  $d \in \mathcal{D}$ ,*

$$V(y, u, d) \geq 0 \Leftrightarrow \forall \tau \in (-\infty, 0], \xi([\tau, 0], y, u, d) \notin T \quad (2.7)$$

*Proof.*

$$\begin{aligned}
V(y, u, d) \geq 0 &\Leftrightarrow \inf_{\tau \in (-\infty, 0]} l(\xi([\tau, 0], y, u, d)) \geq 0 \\
&\Leftrightarrow \forall \tau \in (-\infty, 0], l(\xi([\tau, 0], y, u, d)) \geq 0 \\
&\Leftrightarrow \forall \tau \in (-\infty, 0], \xi([\tau, 0], y, u, d) \notin T
\end{aligned}$$

Here, we are just applying our definition of  $V$  and  $l$ . □

We use Lemma 2.2 to recharacterize  $\text{Pre}(T)$  and  $\text{Avoid}(T)$ .

**Corrolary 2.3.**

$$\text{Pre}(T) = \{y \in X \mid \forall \gamma \in \Upsilon, \exists d \in \mathcal{D}, V(y, \gamma(d), d) < 0\} \quad (2.8)$$

$$\text{Avoid}(T) = \{y \in X \mid \exists \gamma \in \Upsilon, \forall d \in \mathcal{D}, V(y, \gamma(d), d) \geq 0\} \quad (2.9)$$

*Proof.*

$$\begin{aligned}
\text{Pre}(T) &= \{y \in X \mid \forall \gamma \in \Upsilon, \exists d \in \mathcal{D}, \exists \tau \in (-\infty, 0], \xi([\tau, 0], y, \gamma(d), d) \in T\} \\
&= \{y \in X \mid \forall \gamma \in \Upsilon, \exists d \in \mathcal{D}, \neg(\forall \tau \in (-\infty, 0], \xi([\tau, 0], y, \gamma(d), d) \notin T)\} \\
&= \{y \in X \mid \forall \gamma \in \Upsilon, \exists d \in \mathcal{D}, \neg(V(y, \gamma(d), d) \geq 0)\} \\
&= \{y \in X \mid \forall \gamma \in \Upsilon, \exists d \in \mathcal{D}, V(y, \gamma(d), d) < 0\}
\end{aligned}$$

Here we applied predicate negation and Lemma 2.2.

$$\begin{aligned}
\text{Avoid}(T) &= \{y \in X \mid \exists \gamma \in \Upsilon, \forall d \in \mathcal{D}, \forall \tau \in (-\infty, 0], \xi([\tau, 0], y, \gamma(d), d) \notin T\} \\
&= \{y \in X \mid \exists \gamma \in \Upsilon, \forall d \in \mathcal{D}, V(y, \gamma(d), d) \geq 0\}
\end{aligned}$$

Here we simply applied Lemma 2.2. □

Now recall that Player I wins the game by causing a collision. In light of Equation 2.8, at each time  $t$  Player I should choose  $d$  to minimize  $V(x(t), \gamma(d), d)$  and move the state towards the target set until he wins the game. Similarly Player II should choose  $\gamma$  to maximize  $V(x(t), \gamma(d), d)$ , so he can keep the state outside of  $\text{Pre}(T)$  and win the game. Given our information pattern, Player I will have to minimize  $V(x(t), \gamma(d), d)$  over the worst case choose of  $\gamma$ . In Soft Walls, a malicious pilot will try to minimize  $V(x(t), \gamma(d), d)$ , and our controller will try to maximize  $V(x(t), \gamma(d), d)$ .

We will show how we can use these strategies to compute  $\text{Pre}(T)$  in Theorem 2.6. First, however, let  $V^* : X \rightarrow \mathbb{R}$  be the *optimal cost function*, where

$$V^*(y) = \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} V(y, \gamma(d), d). \quad (2.10)$$

We must the existence and uniqueness of  $V^*$  before we use it.

**Lemma 2.4.**  $\forall y \in X, \min_{z \in X} l(z) \leq V^*(y) \leq l(y)$ .

*Proof.* Since  $l$  is continuous and bounded below, and the smallest value occurs when  $z$  is in the closed set  $\bar{T}$ ,  $\min_{z \in X} l(z)$  exists.

$$\begin{aligned} V^*(y) &= \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} \inf_{\tau \in (-\infty, 0]} l(\xi : ([\tau, 0], y, \gamma(d), d)). \\ &\geq \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} \min_{z \in X} l(z) \\ &= \min_{z \in X} l(z) \end{aligned}$$

Here we are saying that no trajectory can give a value of  $x$  which has lower terminal cost than the minimum terminal cost.

$$\begin{aligned} V^*(y) &= \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} \inf_{\tau \in (-\infty, 0]} l(\xi : ([\tau, 0], y, \gamma(d), d)) \\ &\leq \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} l(\xi : ([0, 0], y, \gamma(d), d)) \\ &= l(y), \end{aligned}$$

Here we applying the fact that

$$\inf_{\tau \in (-\infty, 0]} l(\xi : ([\tau, 0], y, \gamma(d), d)) \leq l(y).$$

□

**Corrolary 2.5.**  $V^*$  exists and is unique.

*Proof.* Existence follows immediately from Lemma 2.4. Uniqueness follows from the uniqueness of  $\xi$  and  $l$  and the uniqueness of the inf and sup expressions. □

Finally, we categorize  $\text{Pre}(T)$  and  $\text{Avoid}(T)$  in terms of  $V^*$ .

**Theorem 2.6.**

$$\{y \in X \mid V^*(y) < 0\} \subseteq \text{Pre}(T) \quad (2.11)$$

$$\{y \in X \mid V^*(y) > 0\} \subseteq \text{Avoid}(T) \quad (2.12)$$

*Proof.*

$$\begin{aligned} V^*(y) < 0 &\Leftrightarrow \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} V(y, \gamma(d)d) < 0 \\ &\Rightarrow \forall \gamma \in \Upsilon, \inf_{d \in \mathcal{D}} V(y, \gamma(d)d) < 0 \\ &\Leftrightarrow \forall \gamma \in \Upsilon, \exists d \in \mathcal{D}, V(y, \gamma(d)d) < 0 \end{aligned}$$

We are saying that if  $V^*(y) < 0$ , Player I can choose  $d$  to guarantee a collision. We now apply this to finish the proof of Equation 2.11.

$$\begin{aligned} \{y \in X \mid V^*(y) < 0\} &\subseteq \{y \in X \mid \forall \gamma \in \Upsilon, \exists d \in \mathcal{D}, V(y, \gamma(d)d) < 0\} \\ &= \text{Pre}(T) \end{aligned}$$

Now we prove Equation 2.12.

$$\begin{aligned} V^*(y) > 0 &\Leftrightarrow \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} V(y, \gamma(d)d) > 0 \\ &\Rightarrow \exists \gamma \in \Upsilon, \inf_{d \in \mathcal{D}} V(y, \gamma(d)d) > 0 \\ &\Leftrightarrow \exists \gamma \in \Upsilon, \forall d \in \mathcal{D}, V(y, \gamma(d)d) > 0 \end{aligned}$$

We are saying that if  $V^*(y) > 0$ , Player II can choose  $\gamma$  to avoid any future collision. We now apply this to finish the proof of Equation 2.12.

$$\begin{aligned} \{y \in X \mid V^*(y) > 0\} &\subseteq \{y \in X \mid \exists \gamma \in \Upsilon, \forall d \in \mathcal{D}, V(y, \gamma(d)d) > 0\} \\ &= \text{Avoid}(T) \end{aligned}$$

□

This theorem tells us that if  $V^*(y) > 0$ , then  $y \in \text{Avoid}(T)$ , so we can guarantee safety. If  $V^*(y) < 0$ , then  $y \in \text{Pre}(T)$ , so we cannot guarantee safety. If  $V^*(y) = 0$ , we need extra information to know if  $y$  is a safe or unsafe state. In Soft Walls, this says that if we only know  $V^*$ , we will need to control the aircraft before it reaches the boundary of the backwards-reachable set, which we call  $\partial\text{Pre}(T)$ . We now explain how to compute  $V^*$ .

### 2.1.3 Games of Finite Duration

Consider the case where our game starts at some time  $t < 0$  and terminates at time 0. Under this formulation, we will create a cost function we can compute numerically. We first need some definitions.

- $\text{Pt} : 2^X \times \mathbb{R}_- \rightarrow 2^X$  is the timed predecessor, with

$$\text{Pt}(T, s) = \{y \in X \mid \forall \gamma \in \Upsilon, \exists d \in \mathcal{D}, \exists \tau \in [t, 0], \xi([\tau, 0], y, \gamma(d), d) \in T\}. \quad (2.13)$$

- $\text{At} : 2^X \times \mathbb{R}_- \rightarrow 2^X$  is the timed avoid function, with  $\text{At}(T, s) = X \setminus \text{Pt}$ .
- $J : X \times \mathcal{U} \times \mathcal{D} \times \mathbb{R}_- \rightarrow \mathbb{R}$  is the timed cost function, with

$$J(y, u, d, t) = \min_{\tau \in [t, 0]} l(\xi([\tau, 0], y, u, d)). \quad (2.14)$$

Note that the we use  $\min$  instead of  $\inf$ , because the set  $[t, 0]$  is closed, so the minimum will exist.

- $J^* : X \times \mathbb{R}_- \rightarrow \mathbb{R}$  is the optimal timed cost function, with

$$J^*(y, t) = \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} J(y, \gamma(d), d, t). \quad (2.15)$$

By the same reasoning as in the last section, we can show that  $J^*$  exists and is unique. This gives us Proposition 2.7.

**Proposition 2.7.**

$$\{y \in X \mid J^*(y, t) < 0\} \subseteq \text{Pt}(T, t) \quad (2.16)$$

$$\{y \in X \mid J^*(y, t) > 0\} \subseteq \text{At}(T, t) \quad (2.17)$$

The proof of Proposition 2.7 is nearly identical to that of Theorem 2.6. We now show that at each  $y \in X$   $J^*(y, t)$  is decreasing as the starting time  $t$  decreases.

**Lemma 2.8.** *For all  $t \in (-\infty, 0]$  and  $s \in (-\infty, 0]$ , with  $s < t$ , and for all  $y \in X$ ,*

$$J^*(y, s) \leq J^*(y, t) \quad (2.18)$$

$$V^*(y) \leq J^*(y, s) \quad (2.19)$$

*Proof.*

$$\begin{aligned}
J^*(y, s) &= \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} \min_{\tau \in [s, 0]} l(\xi([\tau, 0], y, u, d)) \\
&= \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} \min \left\{ \min_{\tau \in [s, t]} l(\xi([\tau, 0], y, u, d)), \min_{\tau \in [t, 0]} l(\xi([\tau, 0], y, u, d)) \right\} \\
&= \sup_{\gamma \in \Upsilon} \min \left\{ \inf_{d \in \mathcal{D}} \min_{\tau \in [s, t]} l(\xi([\tau, 0], y, u, d)), \inf_{d \in \mathcal{D}} \min_{\tau \in [t, 0]} l(\xi([\tau, 0], y, u, d)) \right\} \\
&= \min \left\{ \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} \min_{\tau \in [s, t]} l(\xi([\tau, 0], y, u, d)), J^*(y, t) \right\}
\end{aligned}$$

Thus  $J^*(y, s) \leq J^*(y, t)$ . Here we have broken  $[s, 0]$  into the two intervals  $[s, t]$  and  $[t, 0]$ , to prove Equation 2.18.

$$\begin{aligned}
V^*(y) &= \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} \min_{\tau \in (-\infty, 0]} l(\xi([\tau, 0], y, u, d)) \\
&= \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} \min \left\{ \min_{\tau \in (-\infty, s]} l(\xi([\tau, 0], y, u, d)), \min_{\tau \in [s, 0]} l(\xi([\tau, 0], y, u, d)) \right\} \\
&= \sup_{\gamma \in \Upsilon} \min \left\{ \inf_{d \in \mathcal{D}} \min_{\tau \in (-\infty, s]} l(\xi([\tau, 0], y, u, d)), \inf_{d \in \mathcal{D}} \min_{\tau \in [s, 0]} l(\xi([\tau, 0], y, u, d)) \right\} \\
&= \min \left\{ \sup_{\gamma \in \Upsilon} \inf_{d \in \mathcal{D}} \min_{\tau \in (-\infty, s]} l(\xi([\tau, 0], y, u, d)), J^*(y, s) \right\}
\end{aligned}$$

Thus  $V^*(y) \leq J^*(y, s)$ . Here we have broken  $(-\infty, 0]$  into the two intervals  $(-\infty, s]$  and  $[s, 0]$ , to prove Equation 2.18.  $\square$

Not only is  $V^*(y)$  a lower bound for  $J^*(y, t)$ , it is also a limit point as  $t \rightarrow -\infty$ . This is what we now show in theorem 2.9

**Theorem 2.9.** *For all  $y \in X$ ,*

$$\lim_{t \rightarrow -\infty} J^*(y, t) = V^*(y) \tag{2.20}$$

### 2.1.4 The Hamilton-Jacobi-Isaacs Approach

We now show how to compute  $J^*(y, t)$  by solving a partial differential equation (PDE). Since this  $J^*(y, t)$  converges to  $V^*(y)$  as  $t$  approaches  $-\infty$ , an approximation for  $J^*(y, t)$  for large, negative  $t$  will be a good approximation

of  $V^*(y)$ . We will then be able to approximate our backwards-reachable set,  $\text{Pre}(T)$ .

Solving  $J^*(y, t)$  from PDE requires that  $J^*(y, t)$  to be differentiable. In practice,  $J^*(y, t)$  will be continuous but may have “kinks”, or points where the function is not differentiable. These kinks correspond to points where the optimal control is not unique. At all points where  $J^*(y, t)$  is differentiable, however, it will satisfy our PDE, so we need  $J^*(y, t)$  to be a general notion of a solution to our PDE. We call this a *viscosity solution*. A good introduction to viscosity solutions can be found in [6].

Let  $h : \mathbb{R} \times \mathbb{R}_- \rightarrow \mathbb{R}$ , and let  $\nabla_y h(y, t)$  denote the gradient with respect to  $y$  at the point  $(y, t)$ . Suppose we have a function  $H : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ , which we call the *Hamiltonian*. In dynamics [12], the Hamiltonian represents the total energy of a time-invariant system if the potential energy is independent of velocity. Our notion of the Hamiltonian is more general than this. If  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  is a continuous function, then a bounded, uniformly continuous function  $h$  is a viscosity solution to the PDE

$$\frac{\partial}{\partial t} h(y, t) + H(\nabla_y h(y, t)) = 0 \quad (2.21)$$

$$h(y, 0) = g(x), \quad (2.22)$$

if for all  $v \in C^\infty(\mathbb{R} \times \mathbb{R}_-)$ ,

1. if  $h - v$  has a local maximum at  $(y_0, t_0)$ , then

$$\frac{\partial}{\partial t} v(y_0, t_0) + H(\nabla_y v(y_0, t_0)) \leq 0. \quad (2.23)$$

2. if  $h - v$  has a local minimum at  $(y_0, t_0)$ , then

$$\frac{\partial}{\partial t} v(y_0, t_0) + H(\nabla_y v(y_0, t_0)) \geq 0. \quad (2.24)$$

If a viscosity solution to Equation 2.21 exists, it is unique. At any point  $(y, t)$  where  $h(y, t)$  is differentiable, it will satisfy the PDE in the ordinary sense.

The Hamiltonian which we will use is

$$H(y, p) = \min \left\{ 0, \min_{e \in D} \max_{v \in U} p^T f(y, v, e) \right\}. \quad (2.25)$$

**Theorem 2.10.** *If  $J^*(y, t) \in C^0(X \times \mathbb{R}_-)$ , then  $J^*(y, t)$  is the unique viscosity solution of*

$$\frac{\partial}{\partial t} J^*(y, t) + H(\nabla_y J^*(y, t)) = 0 \quad (2.26)$$

$$h(y, 0) = g(x), \quad (2.27)$$

where  $H$  is defined in Equation 2.25.

This theorem is very similar to Theorem 4.1 (a) of [5], where we add the min with zero term so that  $J(y, t)$  can only decrease with time. While we may not be able to find an analytic solution to Equation 2.26, we can use the method of [13] to solve the problem numerically. This method gives high-accuracy approximations of  $J^*$  and is able to isolate kinks. This allows us to characterize the safe and unsafe regions of the state space.

### 2.1.5 Safe Controllers

A safe control strategy  $\gamma \in \Upsilon$  will ensure that if the state is near the boundary of the backwards-reachable set,  $\partial\text{Pre}(T)$ , then the state cannot enter  $\partial\text{Pre}(T)$ . The *optimal control strategy*  $\gamma^* : \mathcal{D} \times X \rightarrow \mathcal{U}$ , will always try and move the state away from the reachable set. That is

$$\gamma^*(d, y) = \arg \inf_{\gamma \in \Upsilon} V(y, \gamma(d), d) \quad (2.28)$$

We call the safe set of control strategies  $U_s$ , where

$$U_s = \{\gamma \in \Upsilon \mid \exists \epsilon > 0, \forall y \in X, V^*(y) \leq \epsilon \Rightarrow \forall d \in \mathcal{D}, \gamma(d) = \gamma^*(d, y)\}. \quad (2.29)$$

If we assume  $V$  is differentiable, a safe controller is one for which on the boundary of  $\text{Pre}(T)$ , the value of  $V$  is increasing. That is,

$$U_s = \{\gamma \in \Upsilon \mid \forall t \in I, x(t) \in \partial\text{Pre}(T) \Rightarrow \nabla V^*(x(t))^T f(x(t), (\gamma(d))(t), d(t)) \geq 0\} \quad (2.30)$$

## 2.2 Implementation

We use a two-step process to implement the Soft Walls controller. For the first step, we start with a signed distance function  $l$  such that  $\forall y \in X$ ,

$$|l(y)| = \min_{t \in \partial T} \|y - t\|. \quad (2.31)$$



We choose  $T$  such that this has an analytic solution, so no computation is involved here. Recall from Section 2.1 that  $T = \{y \in X \mid l(y) < 0\}$  and  $\partial T = \{y \in \mathbb{R}^n \mid l(y) = 0\}$ . We then sample the state space near  $T$  and approximate a viscosity solution of Equation 2.26 using the technique presented in [13, Chapter 2], which uses the level-set methods of [14]. This is a numerical technique designed to give high-resolution approximations of  $V^*(y)$  near the boundary of  $\text{Pre}(T)$ . Computation moves backwards in discrete time steps and terminates when the right side of Equation 2.26 gets close to zero. Computation of the reachable set is performed off line, and the approximate values of  $V^*(y)$ , for  $y$  in a finite set  $X_d \subset X$ , are stored in a file. We let  $\hat{V} : X \rightarrow \mathbb{R}$  represent our approximation. So  $\hat{V}(y)$  is our estimate of  $V^*(y)$  if  $y \in X_d$ . If  $y \notin X_d$ , we get  $\hat{V}(y)$  from trilinear interpolation.

We then simulate our controller using Ptolemy II [3]. Ptolemy II is a tool for heterogeneous modelling, simulation, and design of concurrent systems. We use the ODE solvers in Ptolemy II to simulate the aircraft dynamics. We also use Ptolemy II for graphics rendering and interfacing with a user who can give pilot input to the aircraft through a keyboard. This lets us simulate our controller against a pilot. The system marches time along while receiving data from a user, reading  $\hat{V}$  from a file and drawing the graphics on the screen. We can also make plots of the state and save our data in real time.

In our simulations, we use a forward-Euler ODE solver to approximate our dynamics. This solver is the only ODE solver in Ptolemy II which explicitly solves each new integration value while marching time along at a constant step size. This reduces the computation time that would be required to converge to an implicit solution or to converge to the next value of time. We let  $I_d \subset I$  represent our set of sampling times. We let  $x_d : I_d \rightarrow X$  represent our sampled state values. We similarly define  $u_d$  and  $d_d$ . Our discrete dynamics model becomes

$$x_d(t_{k+1}) = x_d(t_k) + h * f(x_d(t_k), u_d(t_k), d_d(t_k)) \quad (2.32)$$

where

$$h = t_{k-1} - t_k \quad (2.33)$$

Here  $h$  is a constant for all  $k$ .

In Soft Walls the disturbance may help us move away from the reachable set if the pilot is trying to move away from the no-fly zone. We choose a control strategy  $\gamma$  which satisfies  $\gamma \in U_s$ . Using Equation 2.30, we ensure

that  $\forall t_k \in I_d$ , if  $x_d(t_k) \in \partial\text{Pre}(T)$ , then

$$\nabla V^*(x_d(t_k))^T f(x_d(t_k), (\gamma(d_d))(t_k), d_d(t_k)) \geq 0. \quad (2.34)$$

We let  $u_d^\dagger : X \times D \rightarrow U$  be a control strategy which always moves the state away from the reachable set. In particular,

$$u_d^\dagger(x_d(t_k), d_d(t_k)) = \min_{v \in U} \{\|v\| \mid \nabla V(x_d(t_k))^T f(x_d(t_k), v, d_d(t_k)) \geq 0\}. \quad (2.35)$$

We can calculate this minimum analytically, using our dynamics equation and input constraints. We omit this equation for  $u_d$  because it is long and not particularly revealing.

In our control input, we wish to decrease the bias to zero as we move away from the reachable set. For some  $\delta > 0$ , we choose

$$u_d(t_k) = \begin{cases} u^\dagger(x_d(t_k), d_d(t_k)) & \text{if } \hat{V}(x_d(t_k)) = 0 \\ (\delta - \hat{V}(x_d(t_k)))u^\dagger(x_d(t_k), d_d(t_k))/\delta & \text{if } \hat{V}(x_d(t_k)) \in (0, \delta) \\ 0 & \text{if } \hat{V}(x_d(t_k)) \geq \delta. \end{cases} \quad (2.36)$$

The theory behind the modified Hamilton-Jacobi-Isaacs equation allows us to verify the safety of a continuous-time controller. It assumes a continuous-time system as well as a continuous state space. In our implementation, we discretize the state space to estimate  $V$ . We also discretize time in Ptolemy II to simulate the system. In addition we suffer round-off error effects from using the computer. Under these conditions, we can no longer verify safety; we can only simulate the system to check for unsafe behaviors.

The level set method of [13] provides high-resolution estimates of  $V$ , which helps ensure the safety guarantee is generally met. The forward Euler integration method of Ptolemy II suffers errors in estimating the dynamics. When the step size  $h$  is large, the integration suffers from truncation error; when  $h$  is small the integration suffers from roundoff error [7]. In our simulations we choose an ad-hoc value of  $h$  to be  $0.33ms$ , corresponding to a  $30Hz$  refresh rate of the images on the screen.

In spite of these numerical limitations, through simulation, we have not been able to steer the aircraft through the no-fly zone. Figure 2.1 shows some actual trajectories observed when the user tried different strategies.

Trajectories of the algorithm which uses knowledge of the pilot input are shown in Figure 2.1. This trajectory assumes knowledge of the pilot input, and the control is computed by Equation 2.36.

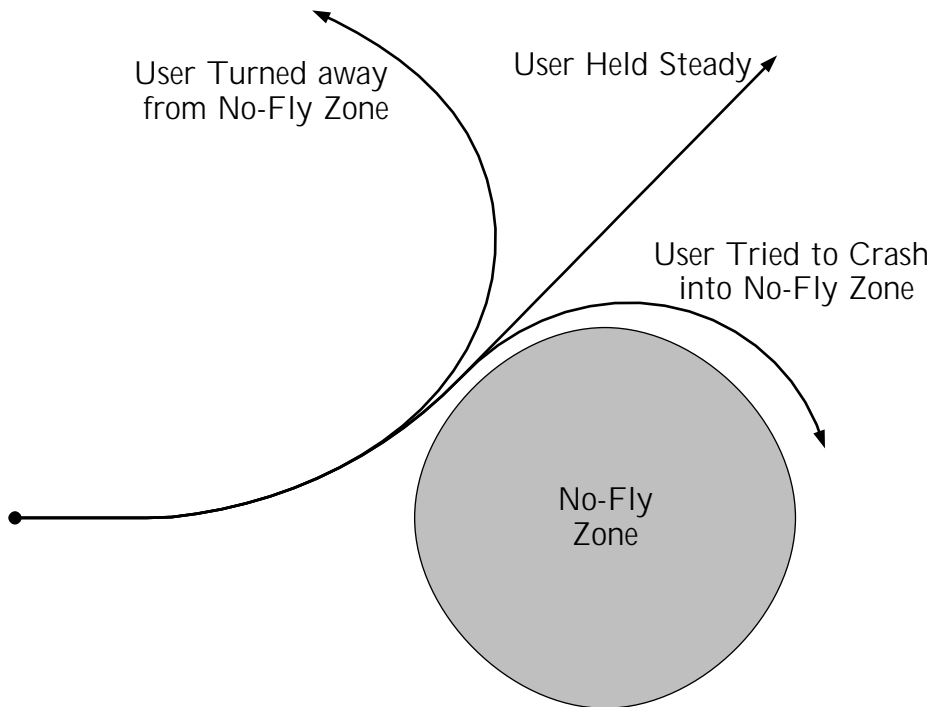


Figure 2.1: A user tried the algorithm in Ptolemy II, and created these trajectories, which depended on his input. This behavior is consistent with that of Figure 1.1. This trajectory assumes knowledge of the pilot input, and the control is computed by Equation 2.36.

### 2.2.1 A Note on Our Dynamics Equation

Recall our simple model of the aircraft dynamics:

$$f \left( \begin{pmatrix} x_{p1}(t) \\ x_{p2}(t) \\ x_h(t) \end{pmatrix}, u(t), d(t) \right) = \begin{pmatrix} s \cos x_h(t) \\ s \sin x_h(t) \\ u(t) + d(t) \end{pmatrix}. \quad (2.37)$$

We mentioned in Chapter 1 that also limit the value of  $u(t) + d(t)$  to the range  $[-M, M]$ , but we do not include this in our dynamics model. We will now see why this is okay.

From differential game theory, an optimal control input will maximize the Hamiltonian while the optimal disturbance will minimize the Hamiltonian

[11]. In our case an optimal control input is one which tries to move the state maximally away from the reachable set, while the optimal disturbance input tries to move it maximally into the reachable set. If we let

$$(V_{p1}(x(t)) \quad V_{p2}(x(t)) \quad V_h(x(t))) = \nabla V^*(x(t))^T, \quad (2.38)$$

then the Hamiltonian is

$$V_{p1}(x(t))s \cos x_h(t) + V_{p2}(t)s \sin x_h(t) + V_h(t)(u(t) + d(t)). \quad (2.39)$$

At each time  $t$ , the optimal control and disturbance inputs are then

$$u^*(x(t)) = 1.5M \operatorname{sgn} V_h(t), \quad (2.40)$$

$$d^*(x(t)) = -M \operatorname{sgn} V_h(t), \quad (2.41)$$

and the optimal Hamiltonian is

$$V_{p1}(t)s \cos x_h(t) + V_{p2}(t)s \sin x_h(t) + 0.5M|V_h(t)|. \quad (2.42)$$

In our computation of  $V^*$ , we must assume  $u^*$  and  $d^*$  are applied [13]. At each  $t$  then  $u(t) + d(t) = \pm 0.5M$ . Because  $|u(t) + d(t)|$  will always be less than  $M$ , we can use the dynamics in Equation 1.3 to calculate  $V^*$  without explicitly including the fact that we limit  $u(t) + d(t)$  to the range  $[-M, M]$  for safety reasons.

# Chapter 3

## Other Soft Walls Control Approaches

In this section we present some results on two other Soft Walls control strategies we have considered before using the method of Chapter 2.

### 3.1 Solution for Half-Space No-Fly Zone

This chapter describes a Soft Walls controller designed by Xiaojun Liu [4]. This controller worked well in simulations, and we prove its safety here. We will consider a particular no-fly zone in this chapter, a half-space in  $\mathbb{R}^2$ . We let  $T = \{x(t) \in X \mid x_{p1}(t) > 0\}$ . See Figure 3.1. Here  $x(0) \in \text{Init} = \{x \in X \mid x_{p1}(t) < -2s/M\}$ . This choice of Init is important in verifying our controller.

One parameter we use in this controller design is  $\tau : X \rightarrow \mathbb{R}_+$ .  $\tau(x(t))$  is, the minimum time it takes for the pilot to hit the no-fly zone from state  $x(t)$ , given zero control input. That is

$$\tau(x(t)) = \min_{d \in \mathcal{D}} \{ \hat{t} \mid x(t + \hat{t}) \in \partial T \}, \quad (3.1)$$

where

$$x(t + \hat{t}) = \int_t^{t+\hat{t}} f(x(s), 0, d(s)) ds. \quad (3.2)$$

Finding  $\tau(x(t))$  is equivalent to minimizing the cost function

$$h(x(t + \hat{t})) + \int_t^{t+\hat{t}} g(x(s), d(s)) ds \quad (3.3)$$

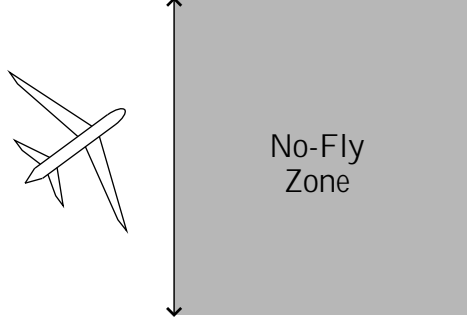


Figure 3.1: The no-fly zone

where for all  $t \in I$ ,  $h : X \rightarrow \mathbb{R}$  and  $g : X \times D \rightarrow \mathbb{R}$  are

$$h(x(t)) = 0 \quad (3.4)$$

$$g(x(t), d(t)) = 1 \quad (3.5)$$

For notational convenience, let us assume  $x_h(t) \in (-\pi, \pi]$ . We define  $d^* : X \rightarrow D$  as the optimal disturbance at state  $x$ . That is, if  $d(t) = d^*(x(t))$ ,  $d$  will minimize  $\tau(x(t))$  at each  $t \in I$ .

**Lemma 3.1.** *For all  $t \in I$ ,*

$$\tau(x(t)) = \begin{cases} \frac{|x_h(t)| - |\sin x_h(t)|}{M} - \frac{x_{p1}(t)}{s} & \text{if } x_{p1}(t) \leq \frac{-s|\sin x_h(t)|}{M} \\ \frac{|x_h(t)|}{M} - \frac{1}{M} \arcsin \left( |\sin x_h(t)| + \frac{Mx_{p1}(t)}{s} \right) & \text{if } x_{p1}(t) > \frac{-s|\sin x_h(t)|}{M} \end{cases} \quad (3.6)$$

with

$$d^*(x(t)) = \begin{cases} M & \text{if } x_h(t) < 0 \\ 0 & \text{if } x_h(t) = 0 \\ -M & \text{if } x_h(t) > 0 \end{cases} \quad (3.7)$$

*Proof.* Along the optimal trajectory, the optimal cost-to-go  $J^* : X \times I \rightarrow \mathbb{R}$  will satisfy the Hamilton-Jacobi-Bellman equation [2]:

$$g(x(t), d^*(x(t))) + \nabla_t J^*(x(t), t) \nabla_{x(t)} J^*(x(t), t)^T f(x(t), 0, d^*(x(t))) = 0, \quad (3.8)$$

$$J^*(x(t), t + \tau(x(t))) = h(x). \quad (3.9)$$

Here  $J^*(x(t), t) = \tau(x(t))$ . We prove the lemma by proving that Equation 3.8 is satisfied for all  $t$  using  $d^*(x(t))$  and  $\tau(x(t))$ .

**Case 1.** ( $x_h(t) = 0$ ) Since  $|\sin x_h(t)| = 0$ ,  $x_{p1}(t) \leq -s|\sin x_h(t)|/M$  is trivially satisfied, and  $J^*(x(t), t) = -x_{p1}(t)/s$ . Here  $d^*(x(t)) = 0$ . Since  $d^*(x(t)) = 0$ ,  $x_h(\tilde{t}) = 0$  will remain true for all times  $\tilde{t} \geq t$ . We see that  $\nabla_{x(t)} J^*(x(t), t) = (-1/s, 0, 0)^T$  and  $\nabla_t J^*(x(t), t) = 0$ . Then

$$\begin{aligned} g(x(t), d^*(x(t))) + \nabla_t J^*(x(t), t) \nabla_{x(t)} J^*(x(t), t)^T f(x(t), 0, d^*(x(t))) = \\ 1 + \frac{-1}{s} s \cos x_h(t) = 1 - \cos 0 = 0. \end{aligned} \quad (3.10)$$

**Case 2.** ( $(x_h(t) > 0) \wedge (x_{p1}(t) \leq \frac{-s|\sin x_h(t)|}{M})$ ) Here  $x_h(t)$  will be decreasing to zero, since  $d^*(x(t)) = -M$ . If  $x_h(t + \delta t)$  becomes zero at some time  $t + \delta t$ , then  $d^*(x(t)) = 0$ . When  $x_h(t) > 0$ ,  $|\sin x_h(t)| = \sin x_h(t)$ , and  $J^*(x(t), t) = (x_h(t) - \sin x_h(t))/M - x_{p1}(t)/s$ . We see that  $\nabla_{x(t)} J^*(x(t), t) = (-1/s, 0, (1 - \cos x_h(t))/M)^T$  and  $\nabla_t J^*(x(t), t) = 0$ . Then

$$\begin{aligned} g(x(t), d^*(x(t))) + \nabla_t J^*(x(t), t) \nabla_x J^*(x(t), t)^T f(x(t), 0, d^*(x(t))) = \\ 1 - \frac{1}{s} s \cos x_h(t) - M \frac{1 - \cos x_h(t)}{M} = 0. \end{aligned}$$

The time required for  $x_h(t)$  to go to zero is  $\delta t = x_h(t)/M$ . From our dynamics,

$$\begin{aligned} x_{p1}(t + \delta t) &= x_{p1}(t) + \int_t^{t+x_h(t)/M} s \cos x_h(\tilde{t}) d\tilde{t} \\ &= x_{p1}(t) + \int_t^{t+x_h(t)/M} s \cos(x_h(t) + M(t - \tilde{t})) d\tilde{t} \\ &= x_{p1}(t) - \frac{s}{M} \sin 0 + \frac{s}{M} \sin x_h(t) \\ &= x_{p1}(t) + \frac{s}{M} \sin x_h(t) \end{aligned} \quad (3.11)$$

Since we require that  $x_{p1}(t) < -s \sin x_h(t)/M$ ,  $x_h(t)$  will reach zero before the target is reached. Then  $d^*(x(t + \delta t)) = 0$  and the time remaining before collision is now  $\tau(x(t)) - x_h(t)/M = -\sin x_h(t)/M - x_{p1}(t)/s = -x_{p1}(t + \delta t)/s$ . We have already verified the solution is optimal for any time  $\tilde{t}$  when  $d^*(x(\tilde{t})) = 0$  in Case 1.

**Case 3.**  $((x_h(t) < 0) \wedge (x_{p1}(t) \leq \frac{-s|\sin x_h(t)|}{M}))$  This proof is nearly identical to the proof of Case 2, so we omit it here.

**Case 4.**  $((x_h(t) > 0) \wedge (x_{p1}(t) > \frac{-s|\sin x_h(t)|}{M}))$  Here  $x_h(t)$  will be decreasing, but as we saw in Case 2, it will never reach zero, so the optimal disturbance  $d^*(x(\tilde{t})) = -M$  will remain constant for all  $\tilde{t}$  until collision. Here  $J^*(x(t), t) = \frac{x_h(t)}{M} - \frac{1}{M} \arcsin\left(\sin x_h(t) + \frac{Mx_{p1}(t)}{s}\right)$ . Again  $\nabla_t J^*(x(t), t) = 0$  and

$$\nabla_{x(t)} J^*(x(t), t) = \begin{pmatrix} \frac{-1}{s} \left[1 - \left(\sin x_h(t) + \frac{Mx_{p1}(t)}{s}\right)\right]^{-1} \\ 0 \\ \frac{1/M}{-M} \cos x_h(t) \left[1 - \left(\sin x_h(t) + \frac{Mx_{p1}(t)}{s}\right)\right]^{-1} \end{pmatrix}. \quad (3.12)$$

Then

$$\begin{aligned} g(x(t), d^*(x(t))) + \nabla_t J^*(x(t), t) \nabla_{x(t)} J^*(x(t), t)^T f(x(t), 0, d^*(x(t))) = \\ 1 - \cos x_h(t) \left[1 - \left(\sin x_h(t) + \frac{Mx_{p1}(t)}{s}\right)\right]^{-1} \\ - 1 + \cos x_h(t) \left[1 - \left(\sin x_h(t) + \frac{Mx_{p1}(t)}{s}\right)\right]^{-1} = 0. \end{aligned} \quad (3.13)$$

**Case 5.**  $((x_h(t) < 0) \wedge (x_{p1}(t) > \frac{-s|\sin x_h(t)|}{M}))$  This proof is nearly identical to the proof of Case 4, so we omit it here. □

**Lemma 3.2.**  $(\tau(x(t)))$  is continuous with respect to  $t$ .

*Proof.* We know  $x(t)$  is continuous with respect to  $t$ , because  $f(x, u, d)$  is continuous and  $u(t)$  and  $d(t)$  are bounded. It is easy to see that  $\tau(x(t))$  is continuous with respect to  $x(t)$  whenever  $x_h(t) \neq 0$  and  $x_{p1}(t) \neq -s|\sin x_h(t)|/M$ . If  $x_h(t) = 0$ ,

$$\begin{aligned} \frac{x_h(t) - \sin x_h(t)}{M} - \frac{x_{p1}(t)}{s} &= -\frac{x_{p1}(t)}{s} \\ &= \frac{-x_h(t) + \sin x_h(t)}{M} - \frac{x_{p1}(t)}{s}. \end{aligned} \quad (3.14)$$



and

$$\begin{aligned} \frac{x_h(t)}{M} - \frac{1}{M} \arcsin \left( \sin x_h(t) + \frac{Mx_{p1}(t)}{s} \right) &= \frac{1}{M} \arcsin \left( \frac{Mx_{p1}(t)}{s} \right) = \\ &= \frac{-x_h(t)}{M} - \frac{1}{M} \arcsin \left( -\sin x_h(t) + \frac{Mx_{p1}(t)}{s} \right) \end{aligned} \quad (3.15)$$

If  $x_{p1}(t) \neq -s|\sin x_h(t)|/M$ ,

$$\begin{aligned} \frac{|x_h(t)|}{M} - \frac{1}{M} \arcsin \left( |\sin x_h(t)| + \frac{Mx_{p1}(t)}{s} \right) &= \frac{|x_h(t)|}{M} = \\ &= \frac{|x_h(t)| - |\sin x_h(t)|}{M} - \frac{x_{p1}(t)}{s}. \end{aligned} \quad (3.16)$$

Because the composition of continuous functions is continuous functions is continuous [17],  $\tau(x(t))$  is continuous with respect to  $t$ .  $\square$

We can now use  $\tau(x(t))$  to synthesize a safe controller.

**Theorem 3.3.** *Let  $u^\dagger : X \rightarrow U$  be*

$$u^\dagger x(t) = \begin{cases} 1.5M \widetilde{\text{sgn}} x_h(t) & \text{if } \tau(x(t)) \leq \frac{2}{M} \\ \frac{3M}{2\pi-4} (\tau(x(t)) - \frac{\pi}{M}) \widetilde{\text{sgn}} x_h(t) & \text{if } \tau(x(t)) \in \left( \frac{2}{M}, \frac{\pi}{M} \right] \\ 0 & \text{if } \tau(x(t)) > \frac{\pi}{M}. \end{cases} \quad (3.17)$$

where  $\text{sgn} : \mathbb{R} \rightarrow \mathbb{R}$  is

$$\widetilde{\text{sgn}} x = \begin{cases} \text{sgn } a & \text{if } a \neq 0 \\ 1 & \text{if } a = 0. \end{cases} \quad (3.18)$$

If  $x(0) \in \text{Init}$ , then for all  $d \in \mathcal{D}$  and all  $t \geq 0$ ,  $x(t) \notin T$ .

*Proof.* Consider the states where  $\tau(x(t)) = 2/M$ . From Equation 3.6, we see that if  $x_{p1}(t) \leq -s|\sin x_h(t)|/M$ , then  $\tau(x(t)) = 2/M$  if

$$x_{p1}(t) = \frac{s}{M} (|x_h(t)| - |\sin x_h(t)| - 2). \quad (3.19)$$

For these values of  $x(t)$ , the condition that  $x_{p1}(t) \leq -s|\sin x_h(t)|/M$  is equivalent to the condition that  $|x_h(t)| \leq 2$ . Clearly, for all  $|x_h(t)| \leq 2$ ,  $x_{p1}(t) \leq 0$  if  $\tau(x(t)) = 2/M$ .

If  $x_{p1}(t) > -s|\sin x_h(t)|/M$ , then  $\tau(x(t)) = 2/M$  if

$$x_{p1}(t) = \frac{s}{M} [\sin(|x_h(t)| - 2) - |\sin x_h(t)|] \quad (3.20)$$

For these values of  $x(t)$ , the condition that  $x_{p1}(t) > -s|\sin x_h(t)|/M$  is equivalent to the condition that  $|x_h(t)| > 2$ . Here  $x_{p1}(t) \leq 0$  only if  $|x_h(t)| \leq \pi/2 + 1$ , since for values greater than this,  $\sin(|x_h(t)| - 2) - |\sin x_h(t)| > 0$ .

For our initial condition  $x(0) \in \text{Init}$  and from Equation 3.6, the minimum possible value of  $\tau(x(0))$  is  $2/M$ , which occurs when  $x_h(t) = 0$  and  $x_{p1}(t) = -2s/M$ . Using Lemma 3.2, this means that if at any time  $\tilde{t}$ ,  $x(\tilde{t}) \in T$ ,  $\tau(x(t)) = 2/M$ , for some  $t < \tilde{t}$ .

If  $\tau(x(t)) = 2/M$ , at time  $t$ , and  $x_h(t) \geq 0$ , then  $u^\dagger(x(t)) = 1.5M$ . This value will decrease only if  $\tau(x(\tilde{t})) > 2/M$  for some  $\tilde{t} > t$ .

Whenever the  $u^\dagger x(t) = 1.5M$  and  $x_h(t) \in [0, \pi)$ , the pilot input that maximizes  $x_{p1}(t)$  is  $d(x(t)) = -M$ . To see this, we let,  $g(x(t), t) = s \cos(\int_0^t [u^\dagger x(s) + d(x(s))] ds)$ . Then  $x_{p1}(t) = \int_0^t g(x(s), s) ds$ . The pilot input which maximizes  $g(x(t), t)$  will minimize  $\int_0^t [u^\dagger x(s) + d(x(s))] ds$ , assuming  $\int_0^t [u^\dagger x(s) + d(x(s))] ds \in [0, \pi)$ . By the Hamilton-Jacobi-Bellman equation,  $d(x(t)) = -M$ , will minimize this function.

For  $\tau(x(t)) = 2/M$  and  $x_h(t) \in [0, \pi/2)$ ,  $x_{p1}(t)$  will increase until time  $\tilde{t}$ , when  $\int_t^{\tilde{t}} [u^\dagger x(s) + d(x(s))] = \pi/2$ . If  $x_h(t) \geq \pi/2$ ,  $x_{p1}(t)$  will be decreasing. If  $x_h(t) \in [0, \pi/2)$ , given the pilot input  $d(x(t)) = -M$  which maximizes  $x_{p1}(t)$  in for times in the range  $[t, \tilde{t}]$ , then  $\tilde{t} = t + 2(\pi/2 - x_h(t))/M$ . At that time

$$\begin{aligned} x_{p1}(\tilde{t}) &= x_{p1}(t) + \int_t^{\tilde{t}} s \cos \left( x_h(t) + \frac{M(\tau - t)}{2} \right) \\ &= x_{p1}(t) + \frac{2s}{M} \sin \left( x_h(t) + \frac{M(\tilde{t} - t)}{2} \right) x_{p1}(t) + \\ &= x_{p1}(t) + \frac{2s}{M} \sin \frac{\pi}{2} \\ &= x_{p1}(t) + \frac{2s}{M} \end{aligned} \quad (3.21)$$

Since  $x_{p1}(t) = \frac{s}{M}(x_h(t) - \sin x_h(t) - 2)$  in this case,  $x_{p1}(\tilde{t}) = \frac{s}{M}(x_h(t) - \sin x_h(t)) \leq 0$ , since  $\forall x > 0, x > \sin x$ .

Whenever  $x_h(t) = \pi$ , we can see that  $\tau(x(t)) \leq \pi/M$  from Equation 3.6, so the bias will eventually decrease as  $x_h(t)$  increases by the continuity of

$\tau(x(t))$ . Therefore, if  $x_h(t) \geq 0$ , we know that the bias will prevent  $x$  from entering the target set. The proof is similar if  $x_h(t) < 0$ , so we omit it here.  $\square$

We have proved  $u^\dagger$  is a safe controller. See [4] for results from this control strategy.

## 3.2 A Geometric Approach

This is a solution first proposed by Ashwin Ganesan. I present it here as a method we have considered but not implemented. This solution is based on the geometry of aircraft trajectories. I

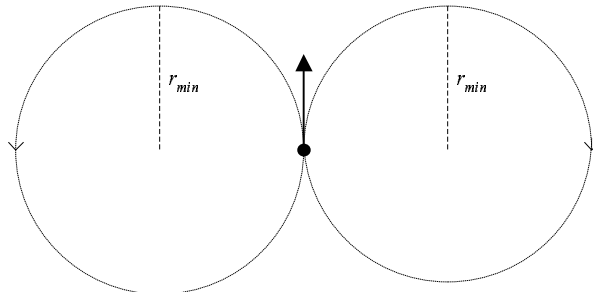


Figure 3.2: The dot represents the aircraft, which is moving along the center trajectory. The circle trajectories represent turns at  $r_{min}$ , which in this case equals  $2s/M$ .

If the aircraft banks left or right with with rate of change in heading  $\pm M/2$ , it will follow one of the trajectories in figure 3.2. Now suppose we wish to prevent the aircraft from entering the no-fly zone. As the aircraft approaches the no-fly zone boundary, one of the minimum-turning-radius trajectories from figure 3.2 will intersect the no-fly zone boundary  $\partial(T)$ . As long as the other trajectory has not yet intersected the boundary, the aircraft can still avoid entering the interior of the no-fly zone by moving along this trajectory. At the instant the second trajectory intersects the boundary, we can force the aircraft to fly along the second trajectory by applying  $u(t) = \pm 1.5M$ . Then even if the pilot applies  $d(t) = \mp M$ , the aircraft will turn away from the no-fly. This situation is depicted in figure 3.3. We can change the bias to zero at the point when (and if) the other trajectory no longer intersects the no-fly zone or its boundary.

We would like the control bias  $u$  to increase and decrease from zero gradually, so as not to shock the pilot. If we use this method there will be discontinuities in  $u$ . We abandoned this strategy when we could not develop a good way to increase and decrease the bias gradually and still guarantee safety.

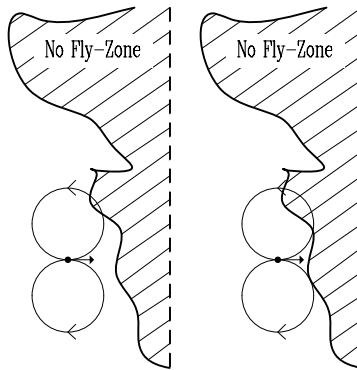


Figure 3.3: On the left picture, the aircraft can still avoid the no-fly zone by banking right. On the right picture, the aircraft must be forced to bank right at a rate greater than or equal to  $-M/2$ .

# Chapter 4

## Conclusions and Future Work

Soft Walls is a new approach to aviation security. From a technical standpoint, Soft Walls is a control system strategy. Under this framework, it is a collision avoidance problem.

When our dynamics equation is Lipschitz continuous, we can guarantee collision avoidance by describing the unsafe set of states through a cost function. We can then model our problem as a game between controllable and uncontrollable inputs. With this formulation we estimate a verifiably safe controller using the Hamilton-Jacobi-Isaacs equation. Simulation validates this strategy.

One drawback of this strategy is that solving the game requires us to solve a problem over the entire state space. The computation and storage required to solve this problem grows exponentially with the number of states [13].

The approach of Section 3.1 uses a particular dynamics equation for the aircraft and a particular no-fly zone geometry. In this case, we were able to derive a safe controller which guarantees the aircraft will always avoid the no-fly zone. This controller is based on the minimum time required for the aircraft to reach the no-fly zone. The advantage of this approach is that it requires only a small amount of online computation.

This approach has two disadvantages. The first is that it depends heavily on the dynamics equation we choose. Changing the dynamics would require resolving the minimum-time-to-reach function and the control input function. This may be difficult or impossible to do. Changing the shape of the no-fly zone would have the same consequence.

The approach of Section 3.2 uses a particular dynamics equation, but it

puts only very limited restrictions on the shape of the no-fly zone. It works by analyzing the trajectories of the aircraft if it is forced to turn right or left, and then forces a turn when necessary.

While this approach removes the restrictions on the no-fly zone, it still was solved for a particular dynamics. It also uses no blending to gradually increase the bias from zero. It is unclear how to do this or how to extend this method under different dynamics.

For these reasons, the game formulation of Chapter 2 is the most promising of these control strategies, because it does not depend on the system dynamics or the no-fly zone shape. Research on appropriate discretizations, of both time and space may reduce the computational complexity of this approach. The research question here is how can you discretize the problem to get useful results? Since the system will be implemented in discrete-time, the formulation should be extended to discrete-time.

Another research topic is that of robustness. In particular, how can you make the system safe with respect to real-life factors such as uncertainty in state information and the timing jitter in a discrete-time realization of the system? This is ongoing research, and we will address these questions. The answers, we hope, will have applications far beyond Soft Walls.

# Bibliography

- [1] Martino Bardi. Some applications of viscosity solutions to optimal control and differential games. In I. Capuzzo Dolcetta and P. L. Lions, editors, *Viscosity Solutions and Applications*. Springer, 1995.
- [2] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1, chapter 3. Athena Scientific, 2nd edition, 2000.
- [3] Shuvra S. Bhattacharyya, Elaine Cheong, John Davis II, Mudit Goel, Christopher Hylands, Bart Kienhuis, Edward A. Lee, Jie Liu, Xiaojun Liu, Lukito Muliadi, Steve Neuendorffer, John Reekie, Neil Smyth, Jeff Tsay, Brian Vogel, Winthrop Williams, Yuhong Xiong, and Haiyang Zheng. Ptolemy II: Heterogeneous concurrent modeling and design in Java. Technical report, University of California, Berkeley.
- [4] J. Adam Cataldo, Edward A. Lee, and Xiaojun Liu. Preliminary version of a two-dimensional technical specification for softwalls. Technical Memorandum UCB/ERL M02/9, University of California, Berkeley, April 2002.
- [5] L. C. Evans and P. E. Souganidis. Differential games and representation formulas for solutions of hamilton-jacobi-isaacs equations. *Indiana University Mathematics Journal*, 33(5):773–797, 1984.
- [6] Lawrence C. Evans. *Partial Differential Equations*, chapter 10. American Mathematical Society, 1998.
- [7] C. William Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Series in Automatic Computation. Prentice-Hall Inc., 1971.



- [8] Rufus Isaacs. *Differential Games: A Mathematical Theory With Applications to Warfare and Pursuit, Control and Optimization*. Krieger Publication Company, 2 edition, 1975.
- [9] Edward A. Lee. Soft Walls—modifying flight systems to limit the flight space of commercial aircraft. Technical Memorandum UCB/ERL M01/31, UC Berkeley, September 2001.
- [10] Edward A. Lee. Soft Walls—frequently asked questions. Technical Memorandum UCB/ERL M03/31, UC Berkeley, July 2003.
- [11] Joseph Lewin. *Differential Games: Theory and Methods for Solving Game Problems with Singular Surfaces*. Springer-Verlag, 1994.
- [12] Jerry B. Marion and Stephen T. Thornton. *Classical Dynamics of Particles and Systems*, chapter 7. Harcourt College Publishers, 4 edition, 1995.
- [13] Ian Mitchell. *Application of Level Set Methods to Control and Reachability Problems in Continuous and Hybrid Systems*. PhD thesis, Stanford University, 2002.
- [14] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2002.
- [15] Claire Tomlin, John Lygeros, and Shankar Sastry. A game theoretic approach to controller design for hybrid systems. In *Proceedings of the IEEE*, volume 88, July 2000.
- [16] Claire Tomlin, George Pappas, and Shankar Sastry. Conflict resolution in air traffic management: A study in multi-agent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, April 1998.
- [17] Neil A. Watson. *Mathematical Analysis Explained*, chapter 3. World Scientific, 1993.