Threshold Strategy for a Leaking Corner-Free Hamilton-Jacobi Reachability with System Decomposition Method

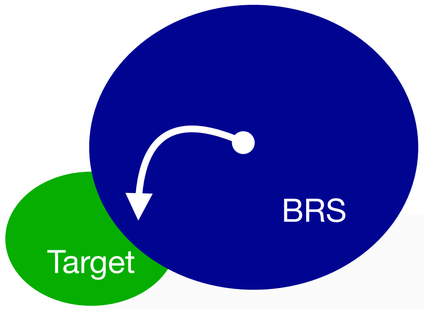
Chong He, Mugilan Mariappan, Keval Vora and Mo Chen

Chong He: chong_he@sfu.ca

2025/04/09

Background

Reachability Math



- Implicit Surface Function (0 sublevel set is the target set):

$$z \in \mathcal{T} \leftrightarrow \ell(z) \leq 0$$

- System Dynamics: $\dot{z} = f(z) + g(z) \cdot u$
- Trajectory: $\ell(\zeta(0; z, t, u(\cdot))$
- Value Functions:

Liveness problem: $V(z,t) = \min_{u(\cdot)\in\mathbb{U}} \ell(\zeta(0; z, t, u(\cdot)))$

Safety problem: $V(z,t) = \max_{u(\cdot)\in\mathbb{U}} \ell(\zeta(0; z, t, u(\cdot)))$

- Backward computation with HJ-PDE (Grid-based Dynamic Programming):

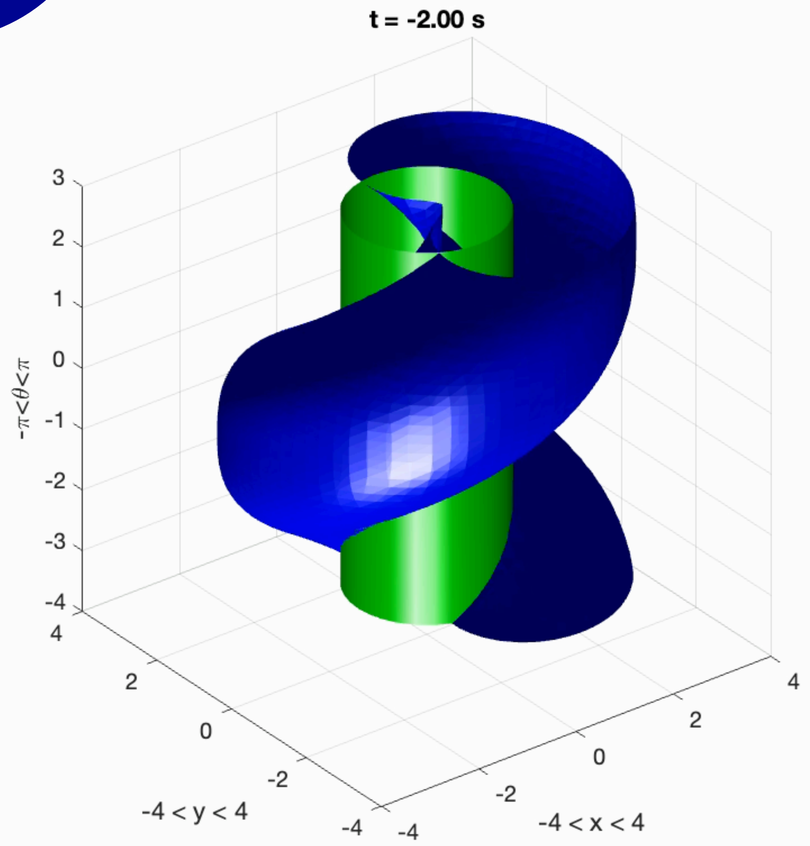
Liveness problem: $V(z, t-\delta t) = V(z,t) + \min_{\dot{z}} D_z V(z,t)\dot{z}\delta t,$

Safety problem: $V(z, t-\delta t) = V(z,t) + \max_{\dot{z}} D_z V(z,t)\dot{z}\delta t, \quad V(z,0) = l(z)$

- Backward Reachable Set (BRS):

$$z \in \mathcal{R}(t) \leftrightarrow V(z,t) \leq 0$$

- Computationally expensive due to the curse of dimensionality



Applicable method

Self-contained Subsystem Decomposition

- Full-dimensional system: 2D Single Integrator

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

- Control Input:

$u = (u_x, u_y)$; constrained by $c(u) = \|u\|_2 - \bar{u} \leq 0$

- Value Function: $V(z, t)$

- Subsystem 1:

$$\dot{x}_1 = \dot{x} = u_x$$

- Control Input:

$w_x = u_x$; constrained by $c_1(w_x) = \|u_x\|_2 - \bar{u} \leq 0$

- Value Function: $V_1(z, t) = \phi_1(x_1, t)$

- Subsystem 2:

$$\dot{x}_2 = \dot{y} = u_y$$

- Control Input:

$w_y = u_y$; constrained by $c_2(w_y) = \|u_y\|_2 - \bar{u} \leq 0$

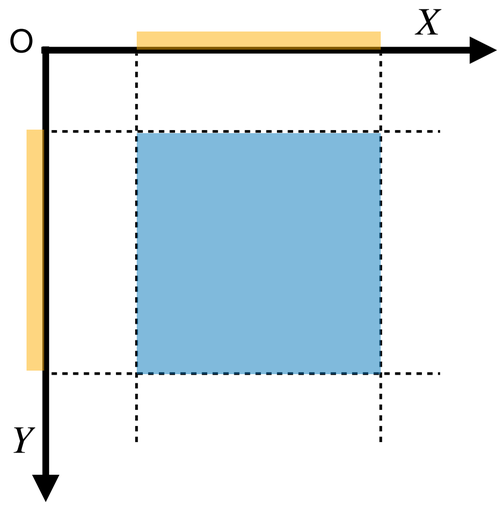
- Value Function: $V_2(z, t) = \phi_2(x_2, t)$

$$c_{\text{joint}}(w_1, w_2) = c(u) \leq 0$$

Background

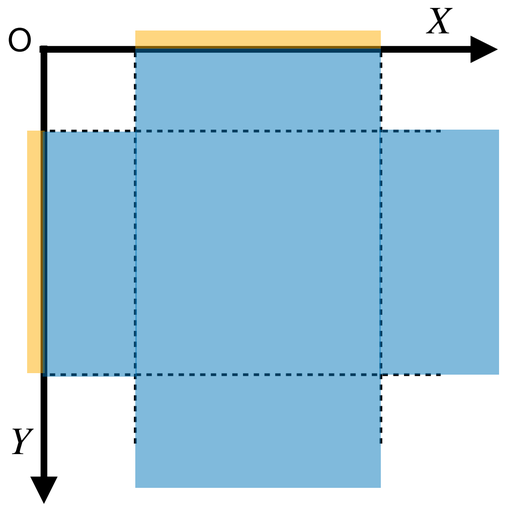
Value Function Decomposition

Intersection:



$$\ell(z) = \max\{\ell_1(x_1), \ell_2(x_2)\}$$

Union:



$$\ell(z) = \min\{\ell_1(x_1), \ell_2(x_2)\}$$

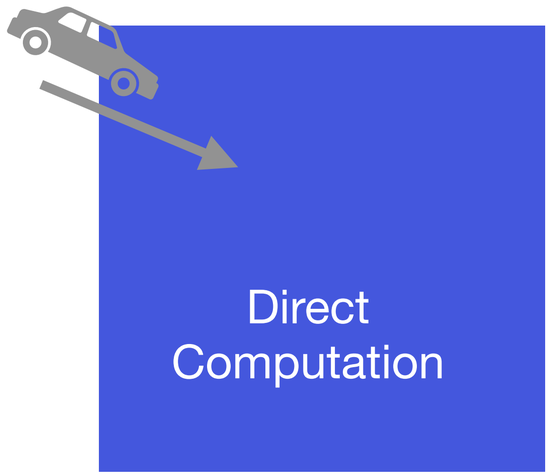
Approximated Value Function:

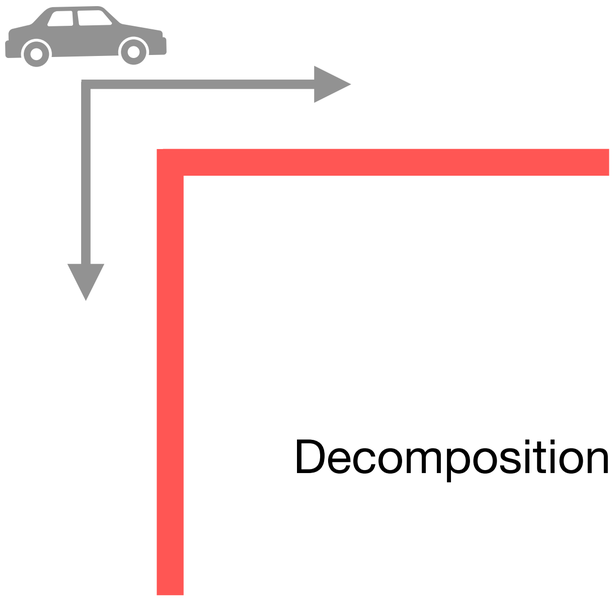
Intersection: $\hat{V}(z,t) = \max\{V_1(x_1,t), V_2(x_2,t))\}$

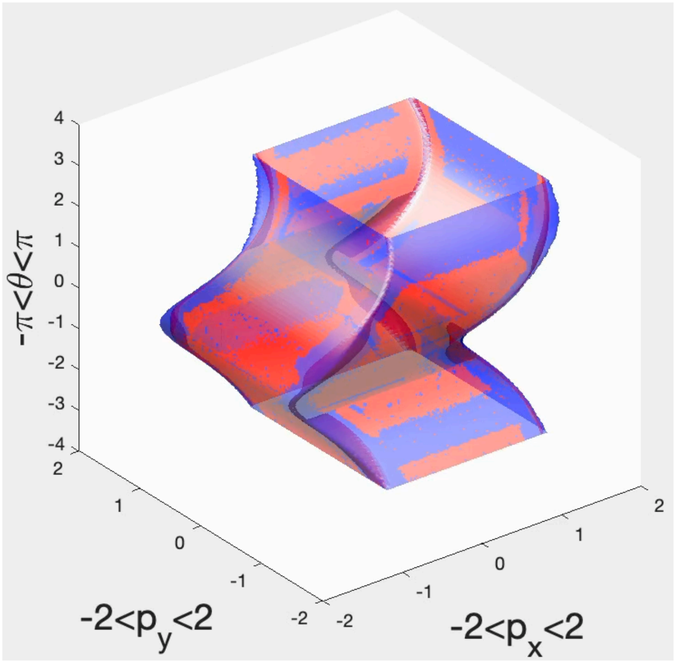
Union: $\hat{V}(z,t) = \min\{V_1(x_1,t), V_2(x_2,t))\}$

Leaking Corner Issue

- When the low-dimensional control are constrained with each other

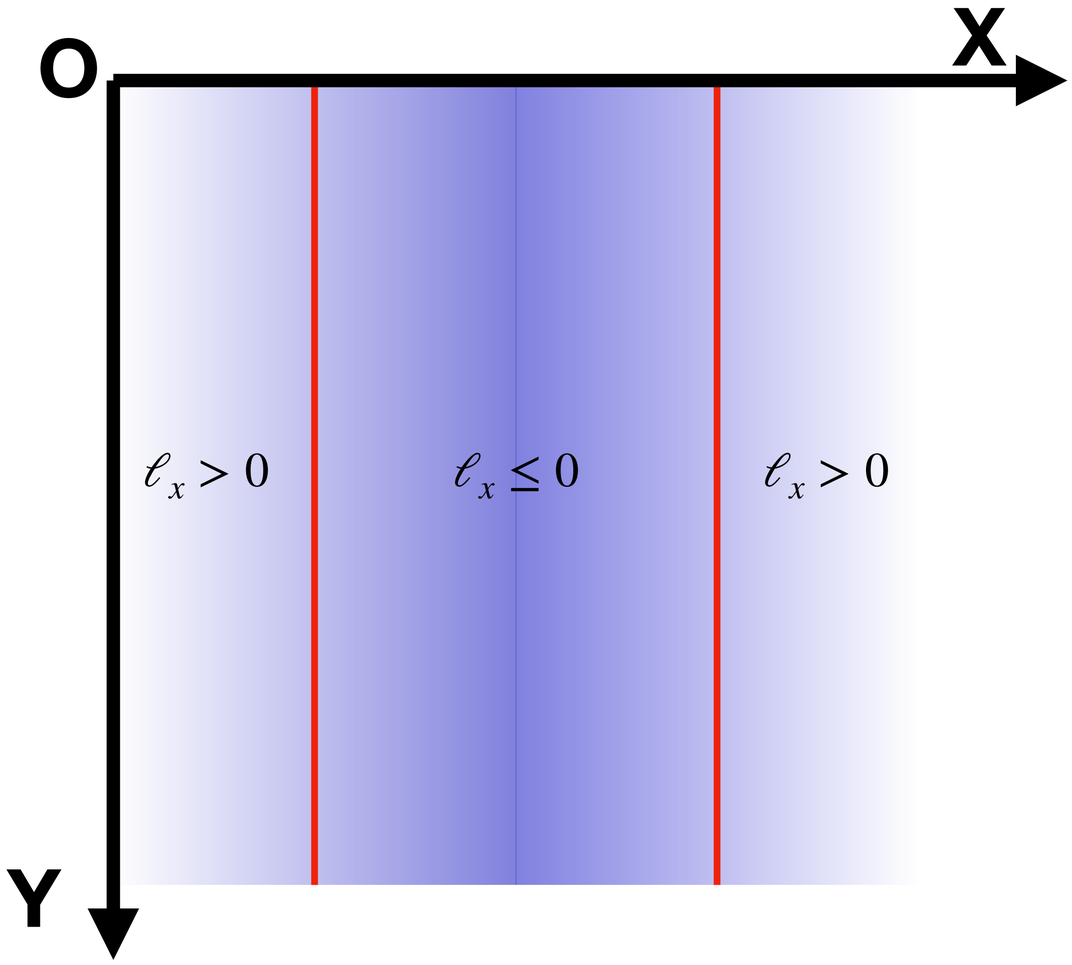


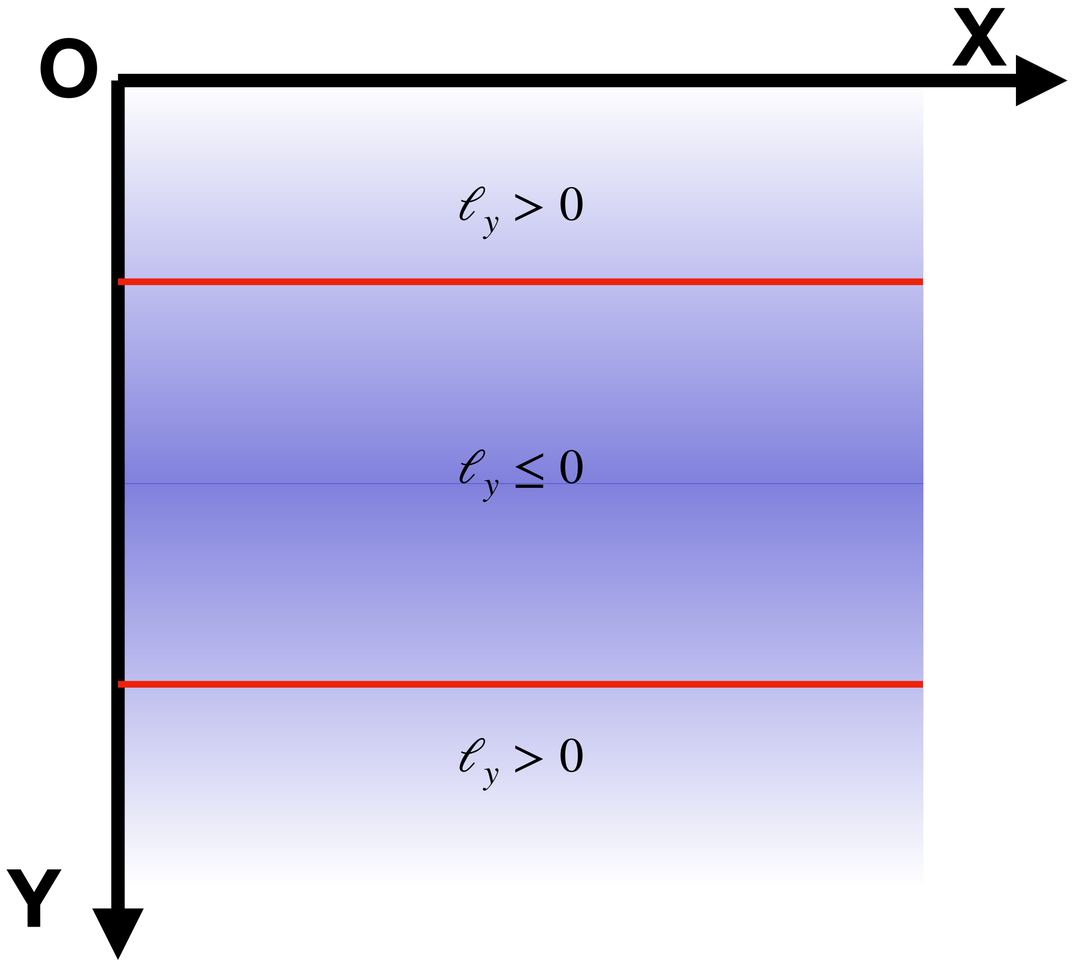




Full-dimensional sub-value functions

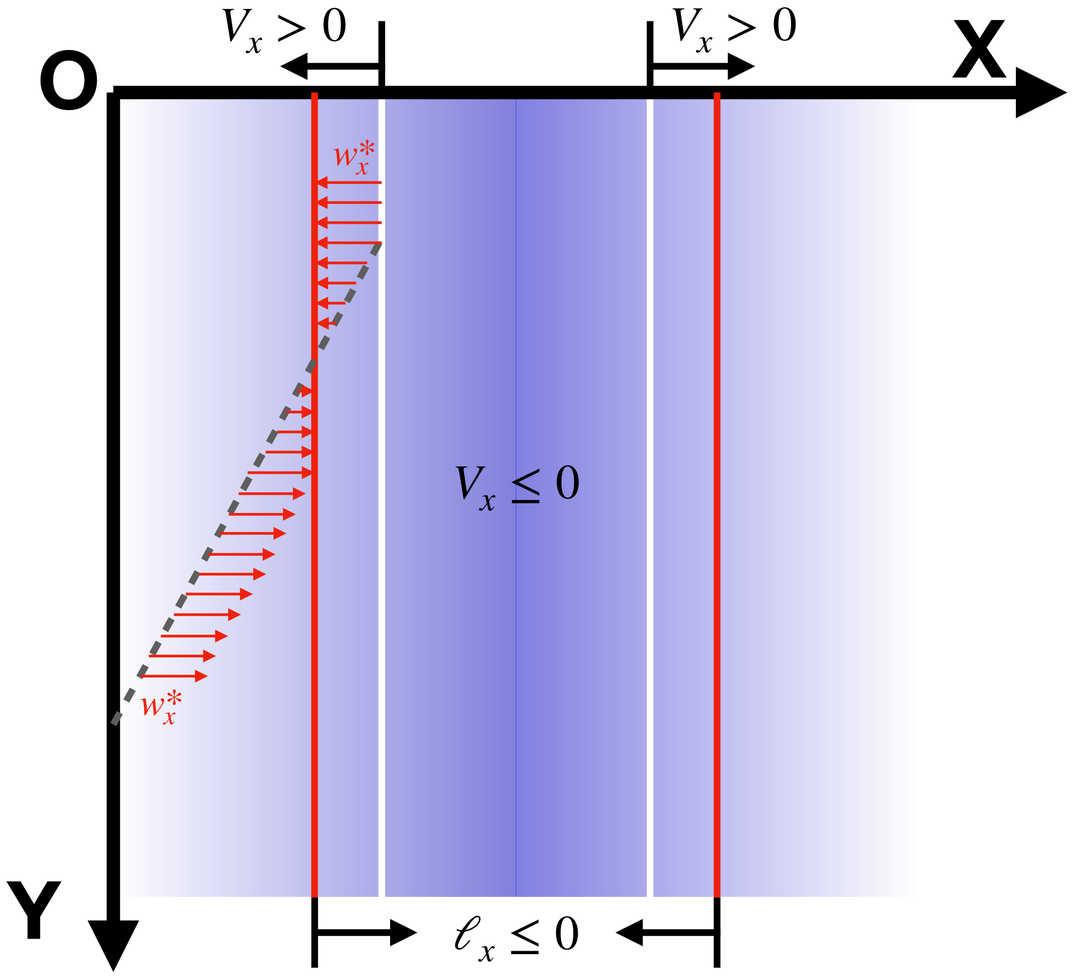
Avoiding zero sub-level set

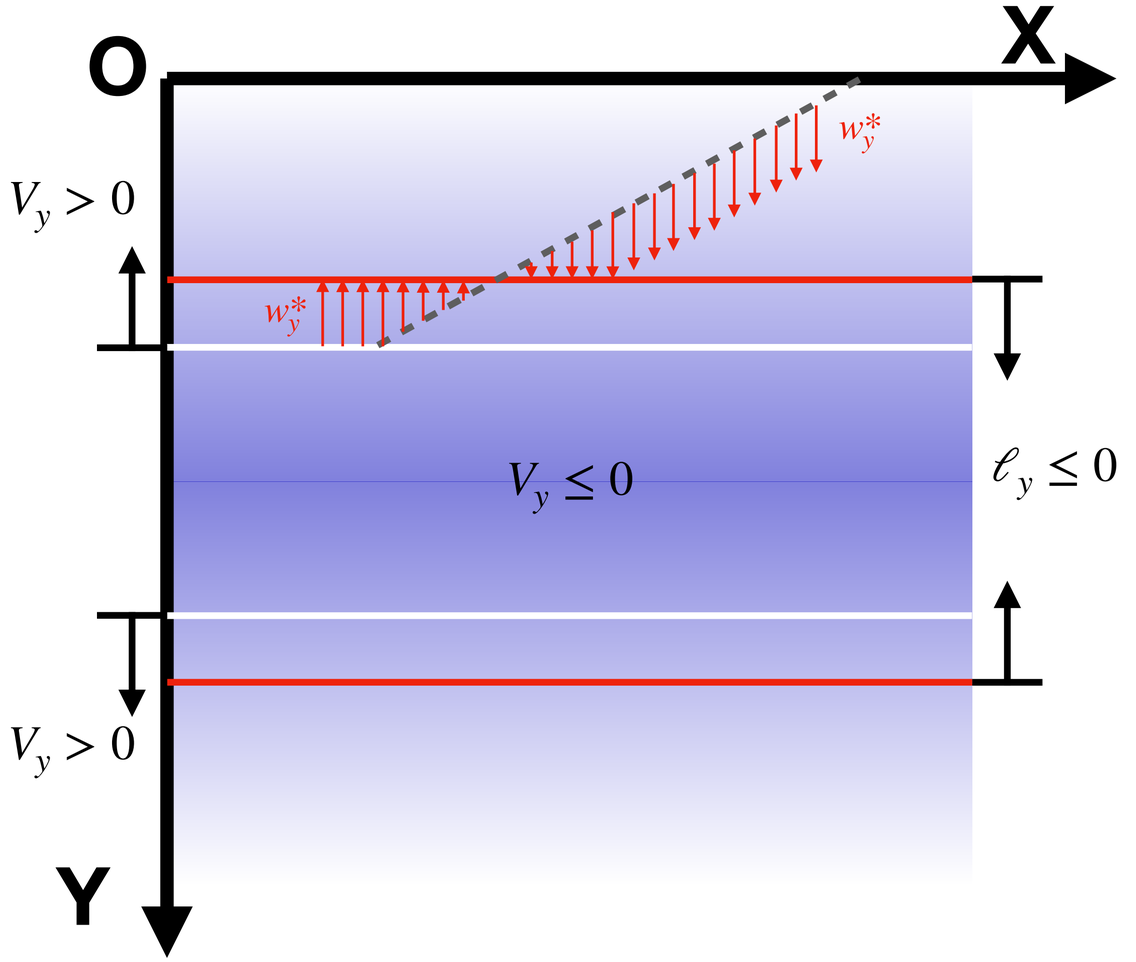




Full-dimensional sub-value functions

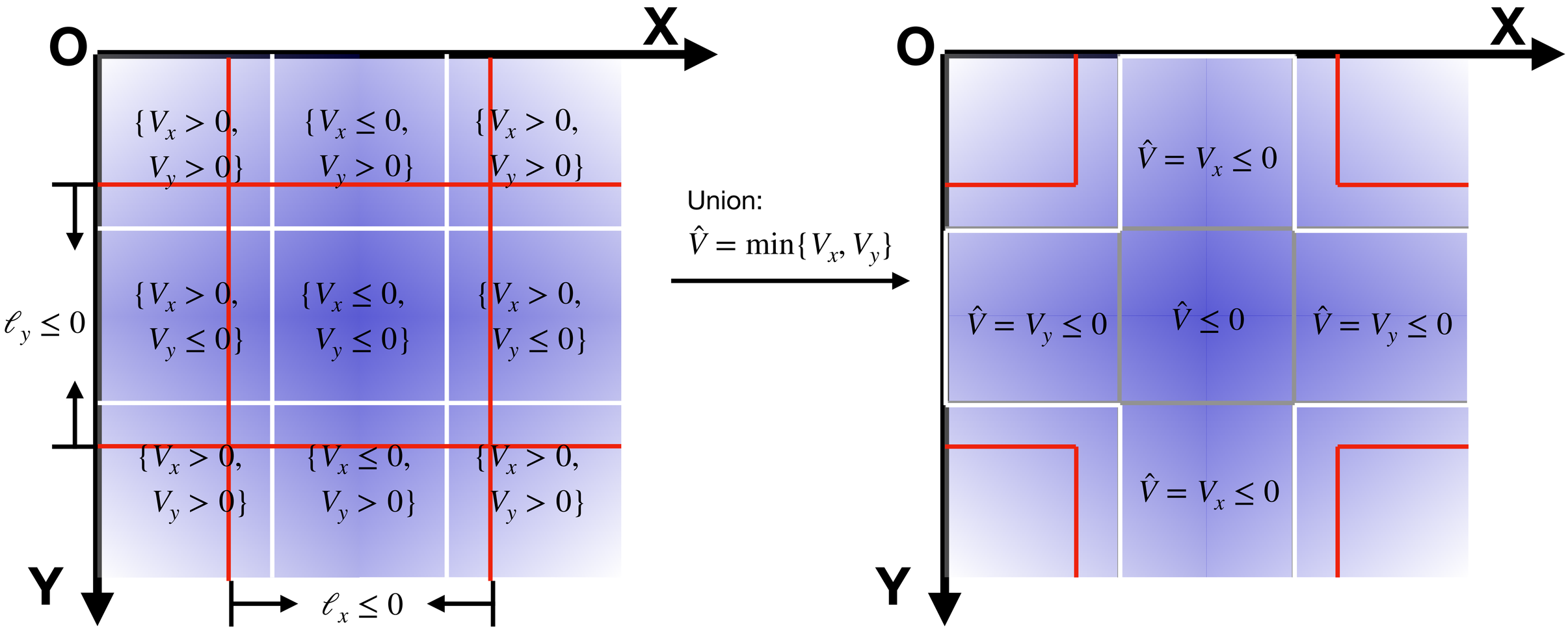
Avoiding zero sub-level set (Admissible Control)





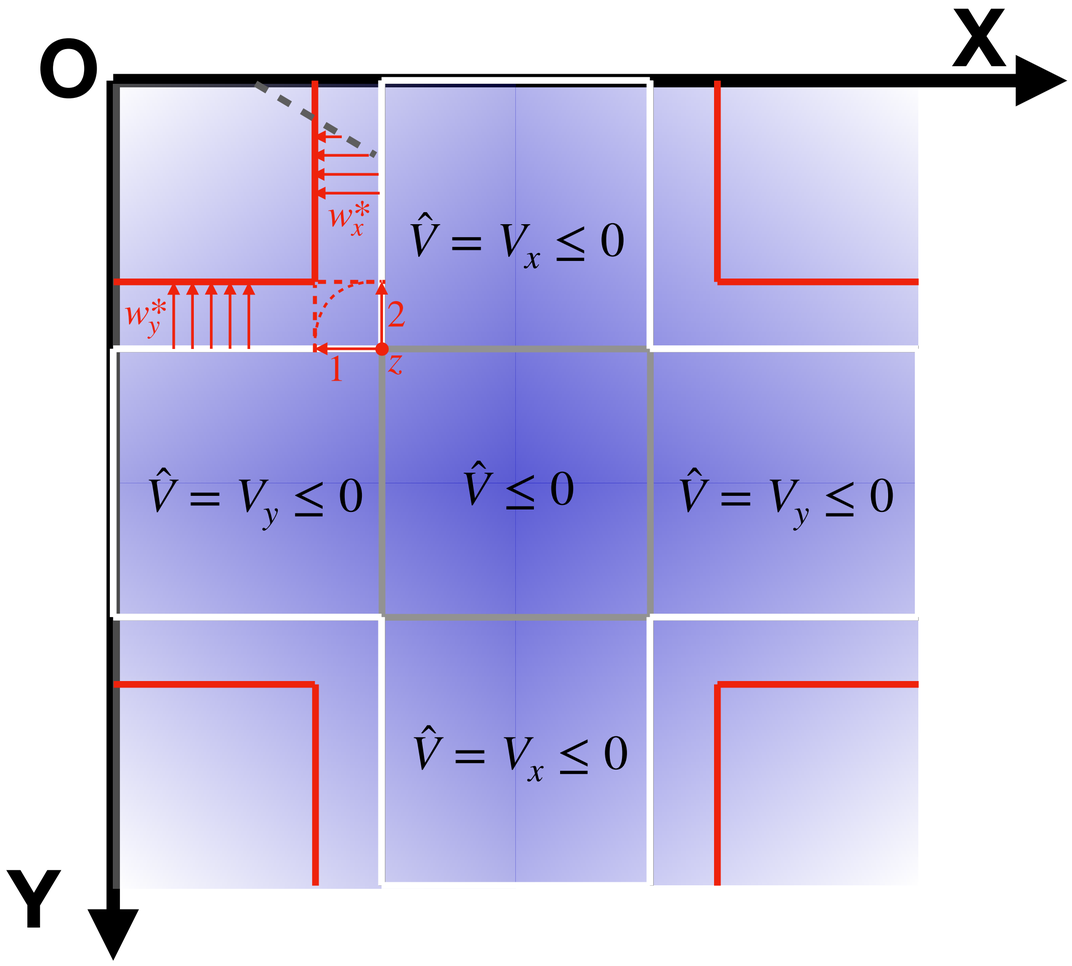
Full-dimensional approximated Value Function

Avoiding the union zero sub-level set

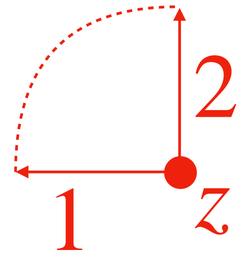


Leaking Corner Issue

Allowable Control



$$u = (u_x, u_y, u_c)$$
$$w_x = (u_x, u_c)$$
$$w_y = (u_y, u_c)$$



The state z in the corner has: $V_x(z) = V_y(z)$

(1) Each low-dimensional control input is acting optimally:

$$c_1(w_1) = c_1(w_x) = \sqrt{u_x^2 + u_c^2} - \bar{u} = 0 \qquad c_2(w_2) = c_2(w_y) = \sqrt{u_y^2 + u_c^2} - \bar{u} = 0$$
$$w_1 = w_x^* \qquad w_2 = w_y^*$$

(2) To form a full-dimensional control input from a pair of low-dimensional controls $(\tilde{w}_x, \tilde{w}_y)$ which could satisfy the control constraint (allowable control):

$$\begin{aligned} c_{\text{joint}}(\tilde{w}_x, \tilde{w}_y) &= \sqrt{u_x^2 + u_y^2 + u_c^2} - \bar{u} \\ &\geq \sqrt{u_x^2 + u_c^2} - \bar{u} \\ &= c_1(w_x^*) = 0 \end{aligned}$$

As a result, to form a pair of low-dimensional controls $(w_x^*, \tilde{w}_y^*)$

$$\tilde{w}_y^* = (0,\ u_c) \neq w_y^* = w_2$$

Leaking Corners

Definition: (Leaking Corners) Suppose we obtain $V(z,t)$ by solving HJ PDE, and $\hat{V}(z,t)$ by combining value functions. The set of "leaking Corners" $\mathscr{L}(t)$ is defined as

$$\mathscr{L}(t) = \{z : V(z,t) \neq \hat{V}(z,t)\}$$

2 cases will suffer from the issue:

(1) Intersection case for liveness problem: $\hat{V}_R(z,t) = \max\{V_{R,1}(x_1,t), V_{R,2}(x_2,t)\}$

(2) Union case for safety problem: $\hat{V}_A(z,t) = \min\{V_{A,1}(x_1,t), V_{A,2}(x_2,t)\}$

Allowable Control

Definition: (Allowable Control) We define a pair of low-dimensional control signals $(\tilde{w}_1, \tilde{w}_2)$ that satisfy the control constraint as "allowable controls". We also define "allowable control functions" $(\tilde{w}_1(\,\cdot\,), \tilde{w}_2(\,\cdot\,))$ as control signals that satisfy the control constraint for all the time. The corresponding low-dimensional value functions evolved under allowable control is given by

$$\tilde{V}_i(x_i, t) = \ell_i(\xi_i(0; x_i, t, \tilde{w}_i(\,\cdot\,))).$$

Allowable Control

Lemma 1: Suppose $\tilde{w}_1(t) = w_1^*(t) = (u_1(t), u_c(t))$ for all t and $(\tilde{w}_1, \tilde{w}_2)$ is a pair of allowable controls. Then, for all t, $u_2(t) = \mathbf{0}$, $\tilde{w}_1(t)$ and $\tilde{w}_2(t)$ share the same $u_c(t)$.

Allowable Control

its relation to the leaking corner

Lemma 2: A state z is not in the leaking corner, $z \notin \mathscr{L}(t)$, if and only if there exists a pair of allowable control functions $\tilde{w}_1(\cdot)$ and $\tilde{w}_2(\cdot)$at every time step such that they satisfy the following case-based conditions :

(1) Intersetion for liveness problem: $\max\{\tilde{V}_{R,1}(x_1,t), \tilde{V}_{R,2}(x_2,t)\} = \hat{V}_R(z,t),$

(2) Union for safety problem: $\min\{\tilde{V}_{A,1}(x_1,t), \tilde{V}_{A,2}(x_2,t)\} = \hat{V}_A(z,t),$

Threshold Strategy

Theorem 1: We can find the set of leaking corners $\mathscr{L}(t)$ by comparing the (full-dimensional) sub-value functions:

$$\mathscr{L}(t) = \{z : |V_1 - V_2| < \Delta\}.$$

The value of Δ is

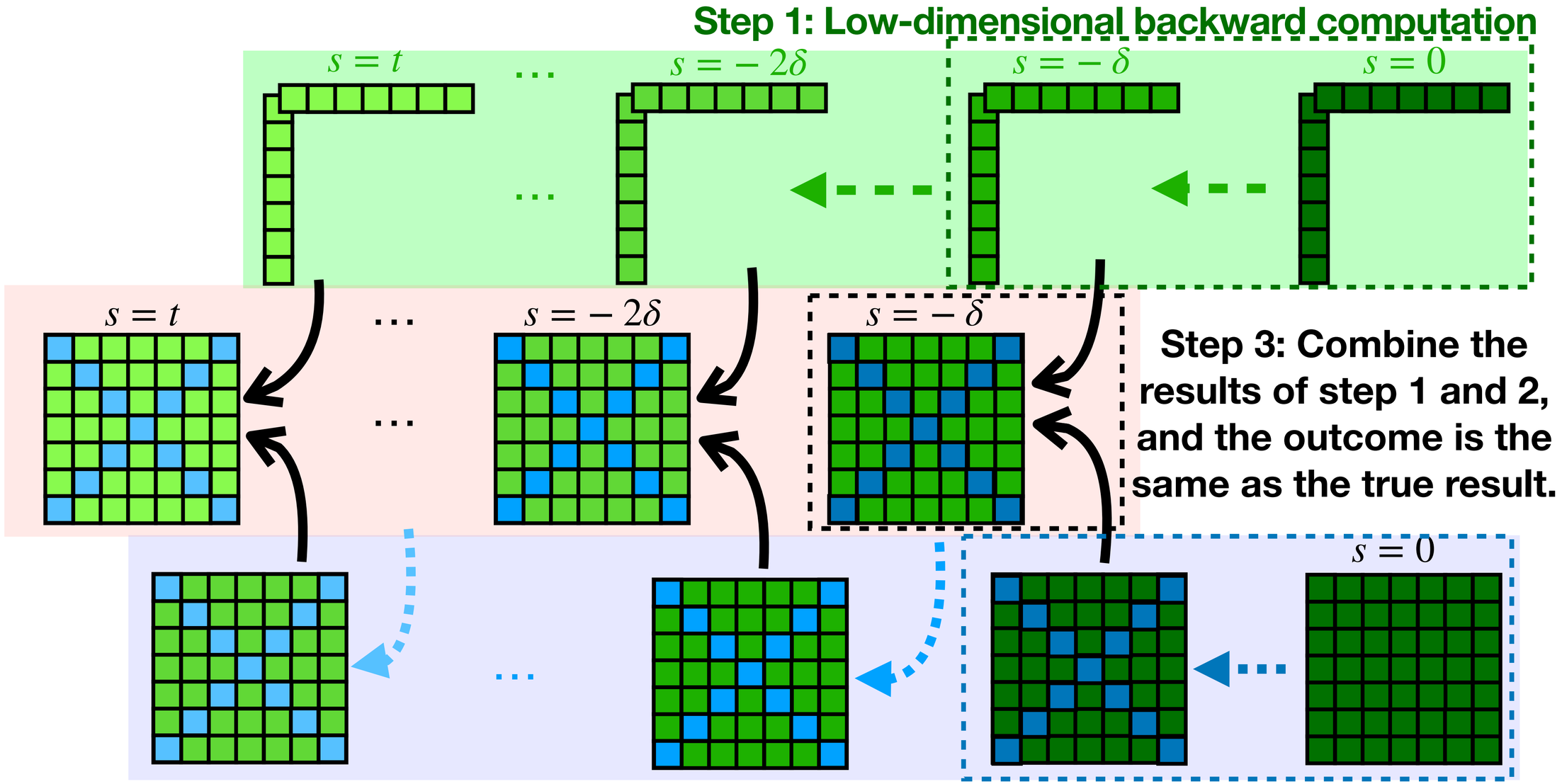
$$\Delta = \begin{cases} |\tilde{V}_1^* - V_1|, \text{if } V_{R,2} \geq V_{R,1} \text{ or } V_{A,1} \geq V_{A,2} \\ |\tilde{V}_2^* - V_2|, \text{if } V_{R,1} \geq V_{R,2} \text{ or } V_{A,2} \geq V_{A,1}. \end{cases}$$

Definition: (Island) The ``leaking corner" set $\mathscr{L}(t)$ consists of a finite union of k connected sets, referred to as islands, denoted by $\mathscr{I}_\kappa(t)$ for $\kappa \in \{1,2,3,...,k\}$. Each $\mathscr{I}_\kappa(t)$ is a connected component of $\mathscr{L}(t)$.

Assume that for all $\kappa \in \{1,2,3,...,k\}$, there exists at least one state $z^* \in \mathscr{I}_\kappa(t)$ such that

$$V_1(z^*, t) = V_2(z^*, t)$$

Updating Method



Algorithm 1: Local updating procedure

Data: $\hat{V}(\cdot,\cdot), \hat{\mathcal{L}}(\cdot), Z, t_{\text{list}} = [t, t+\delta..., 0]$
Result: $\check{V}(\cdot,\cdot)$
$s \leftarrow 0$; ▷ Backward Computation
$\check{V}(\cdot, 0) \leftarrow \hat{V}(\cdot, 0)$;
Frontier $\leftarrow$ nextFrontier $\leftarrow$ visited $\leftarrow \{\}$;
while $s > t$ **do**
  for $z \in Z$ **do**
    if $z \in \hat{\mathcal{L}}(s)$ **then**
      updateValue$(z, s, \delta,$ Frontier$)$
    else
      $\check{V}(z, s-\delta) \leftarrow \hat{V}(z, s-\delta)$;
    end
  end
  visited $\leftarrow \hat{\mathcal{L}}(s)$
  Frontier $\leftarrow$ Frontier $\setminus$ visited
  while *Frontier* $\neq \emptyset$ **do**
    for z *in Frontier* **do**
      updateValue$(z, s, \delta,$ nextFrontier$)$
    end
    visited $\leftarrow$ visited $\cup$ Frontier
    Frontier $\leftarrow$ nextFrontier $\setminus$ visited
    nextFrontier $\leftarrow \{\}$
  end
  $s \leftarrow s - \delta$;
end
def *updateValue*$(z, s, \delta,$ *Frontier*$)$**:**
  $\check{V}(z, s-\delta) \leftarrow$ HJ Update$(\check{V}(z, s))$ ▷ Equation 5
  if $\check{V}(z, s-\delta) \neq \hat{V}(z, s-\delta)$ **then**
    Frontier $\leftarrow$ Frontier $\cup$ neighbor(z)
  end

- The algorithm keep checking the neighboring states of the leaking corners detected for every time step.

- It makes sure that all the leaking corners are covered with the updating procedure.

Results

2D Single Integrator

TABLE I: 2D Accuracy Comparison for One Step

Metric	Before	After
Number of grid points with different values from the ground truth	200	0
Average absolute difference from ground truth	1.2×10^{-4}	9.51×10^{-18}
Maximum absolute difference from ground truth	2×10^{-2}	2.22×10^{-16}

TABLE III: 2D Accuracy Comparison for 10 Steps

Metric	Before	After
Number of states with different values from the ground truth	1344	0
Average absolute difference from ground truth	2.5×10^{-3}	1×10^{-9}
Maximum absolute difference from ground truth	7.39×10^{-2}	2.44×10^{-8}

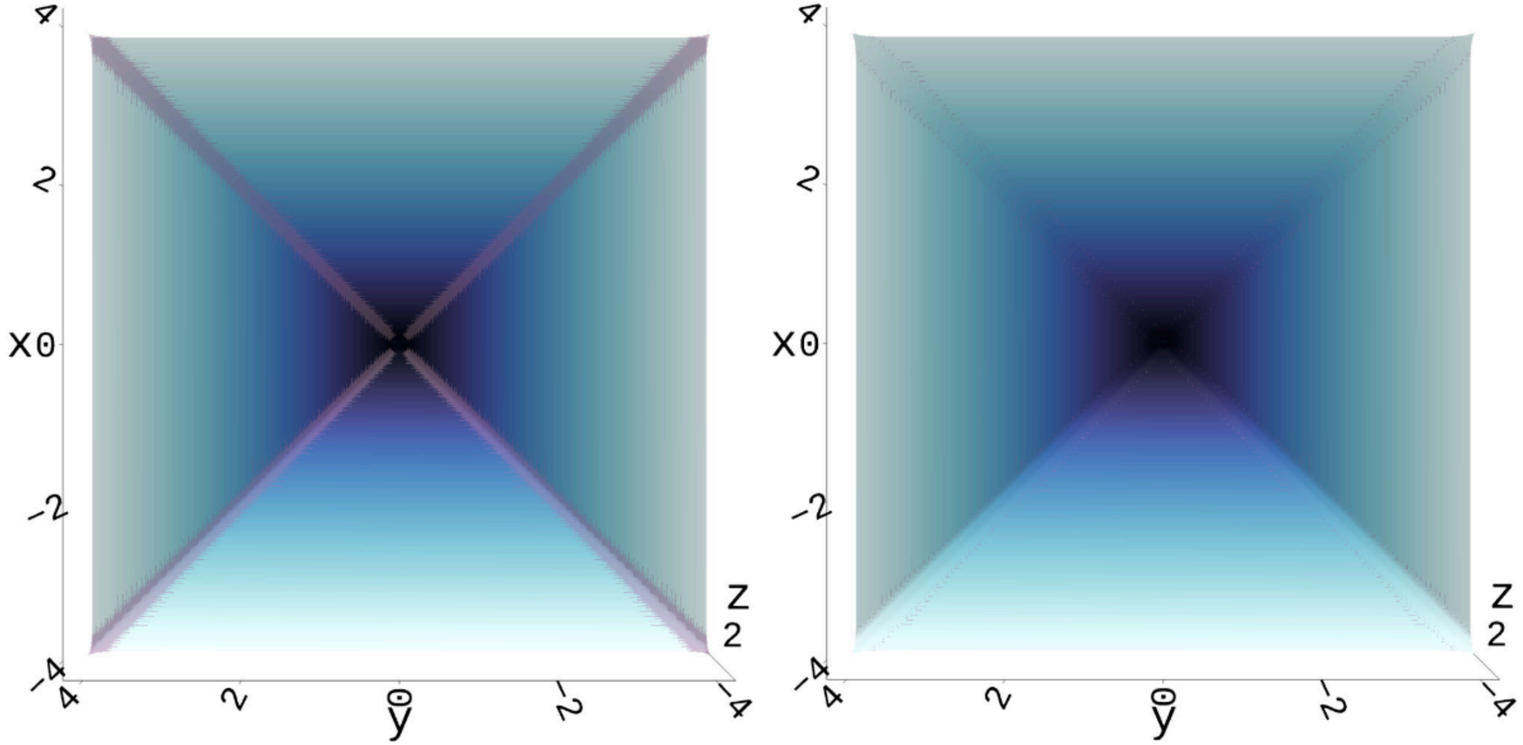


TABLE II: 2D Time Comparison for One Step

Process	Time (seconds)
Direct computation	3.3×10^{-2}
SCSD computation + HJ local update computation	$7 \times 10^{-4} + 1.3 \times 10^{-3} = 2.0 \times 10^{-3}$

TABLE IV: 2D Time Comparison for 10 Steps

Process	Time (seconds)
Direct computation	3.36×10^{-1}
SCSD computation + HJ local update computation	$4.72 \times 10^{-3} + 1.61 \times 10^{-1} = 1.65 \times 10^{-1}$

Results

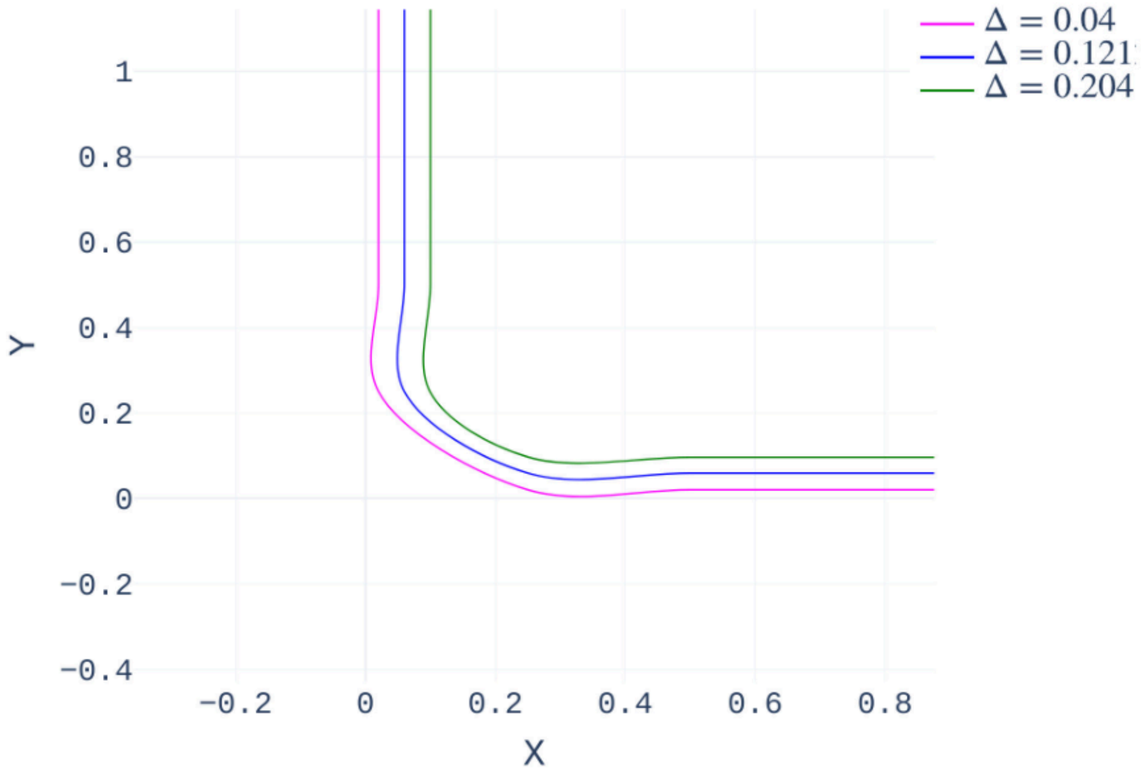
6D Planar Quadrotor

TABLE V: 6D Accuracy Comparison for One Step

Metric	**Before**	**After**
Number of grid points with different values from the ground truth	3.89×10^{6}	0
Average absolute difference from ground truth	6.97×10^{-4}	0.0
Maximum absolute difference from ground truth	4×10^{-2}	0.0

TABLE VI: Computation Time and Delta Value for ts

t **(s)**	Δ	**Decomposition Time + Local Updating Time (seconds)**
-0.02	0.04	2.447 + 47.1078 = 49.5548
-0.06	0.1212	6.769 + 157.3528 = 164.1218
-0.1	0.204	11.8038 + 250.7859 = 262.5897



Conclusion

1. Propose a threshold-based method to detect the leaking corners.
2. Introduce a local updating method that ensures accuracy while maintaining computational efficiency.
3. Validate the method with 2D Single Integrator system and 6D Planar Quadrotor system.

Future Works

1. Extend the work to the cases when there are disturbance
2. Generalize the method to solve the issue with different types of the control constriant
3. Test the method with other computational acceleration techniques
4. Parallelize the local updating procedure for faster computation
5. Explore machine learning or other techniques for new value updating methods.
6. Apply the method to real robots