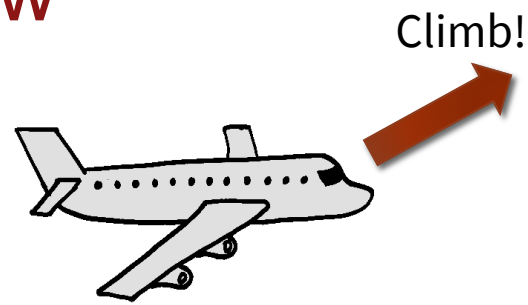
Verifying Aircraft Collision Avoidance Neural Networks through Linear Approximations of Safe Regions

Kyle D. Julian
Mykel J. Kochenderfer

Shivam Sharma
Jean-Baptiste Jeannin

AAAI SSS-19
March 26, 2019

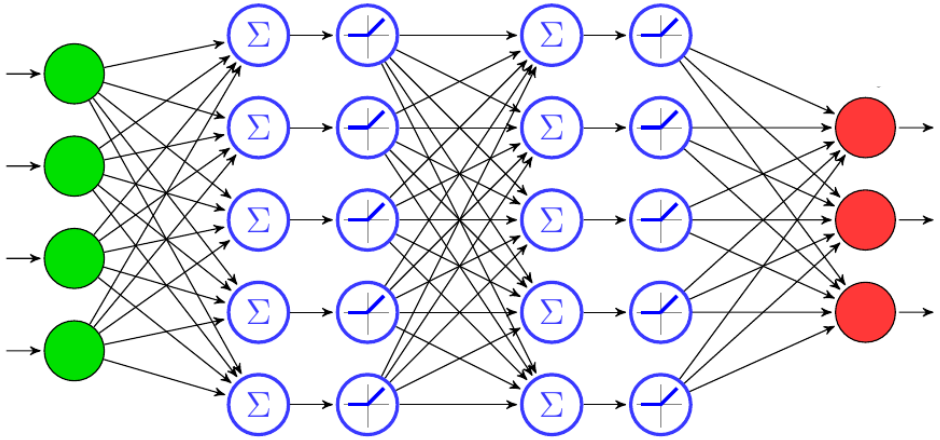
Overview



Ownship

Intruder

Inputs:
Encounter Geometry

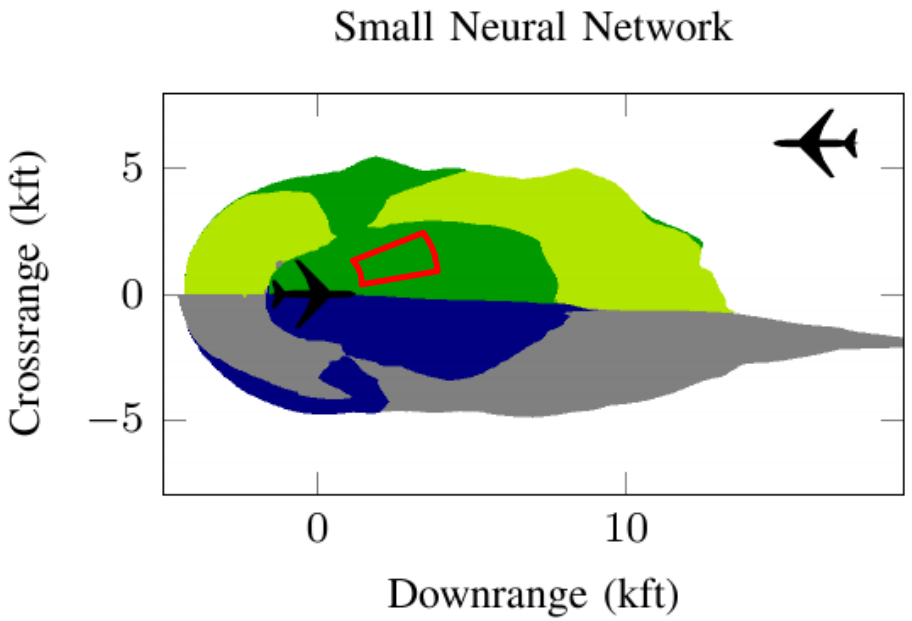


Outputs:
Advisory Scores

Is the neural network collision avoidance system safe?

Closed-Loop Verification

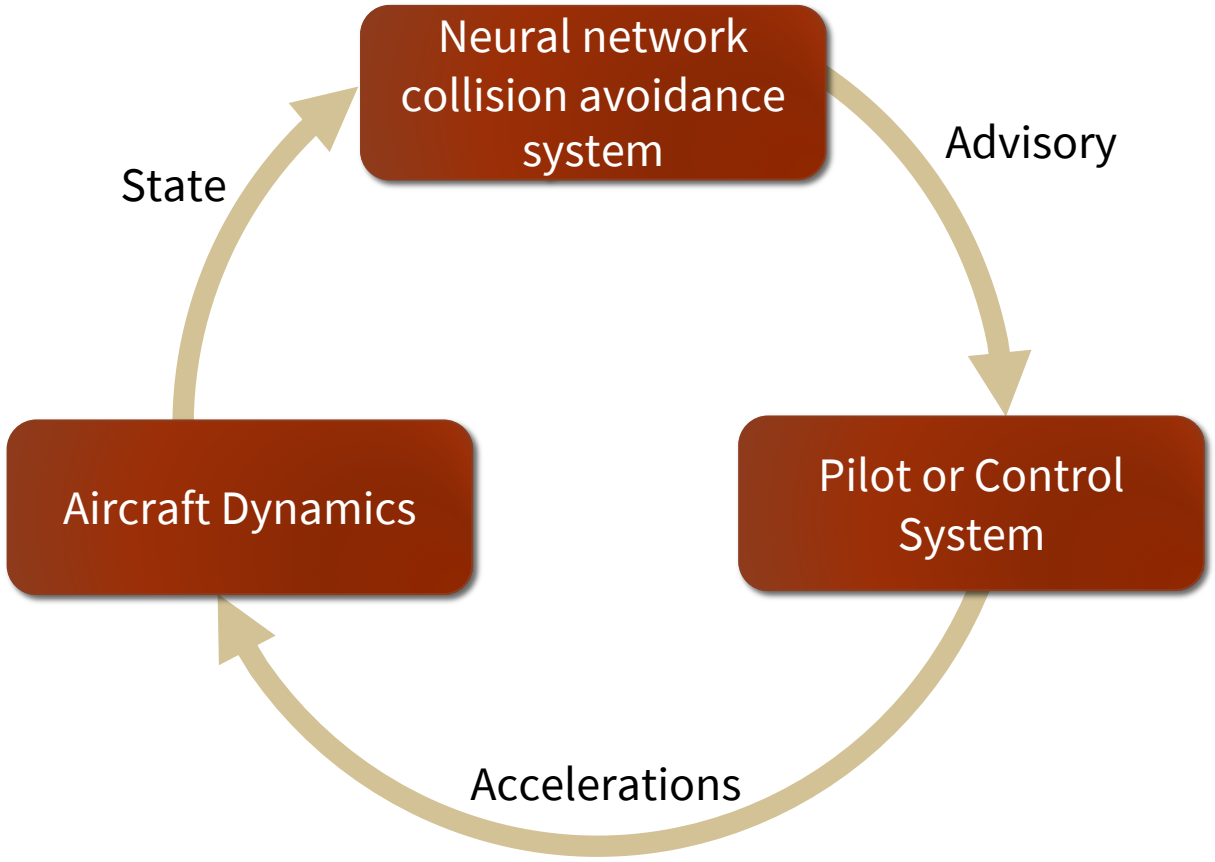
Reluplex properties [CAV'17]



✓ Verifies intuitive properties

✗ Does not guarantee safety

Closed-loop properties



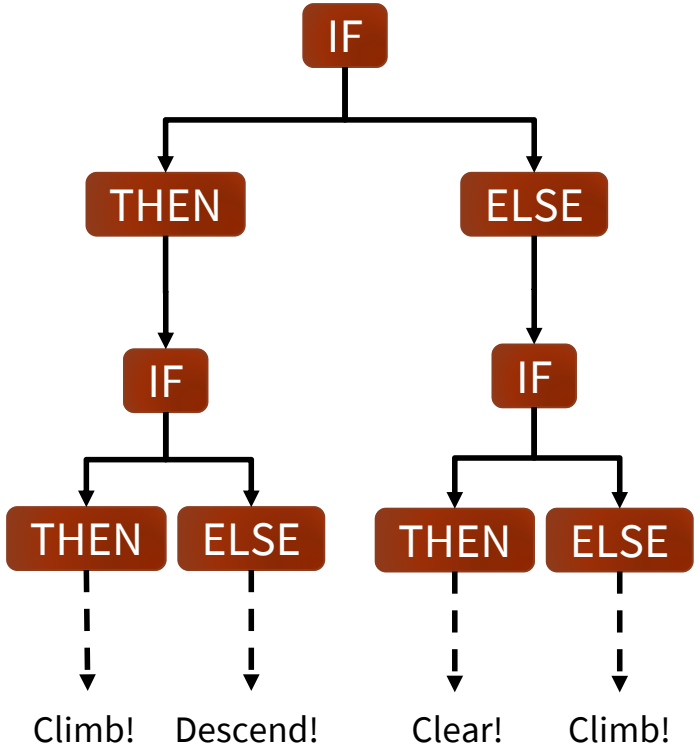
✓ Guarantees safety over time

[CAV'17] Katz et al. Reluplex: An efficient SMT solver for verifying deep neural networks.

Aircraft Collision Avoidance Systems

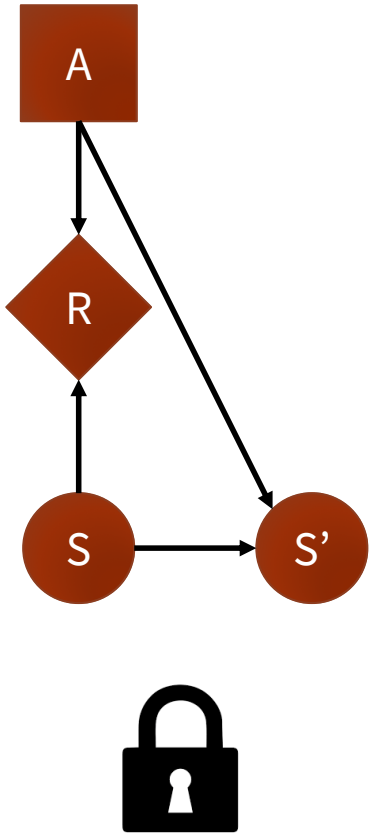
TCAS

Rule-based collision avoidance system
Mandated by FAA since 1980s



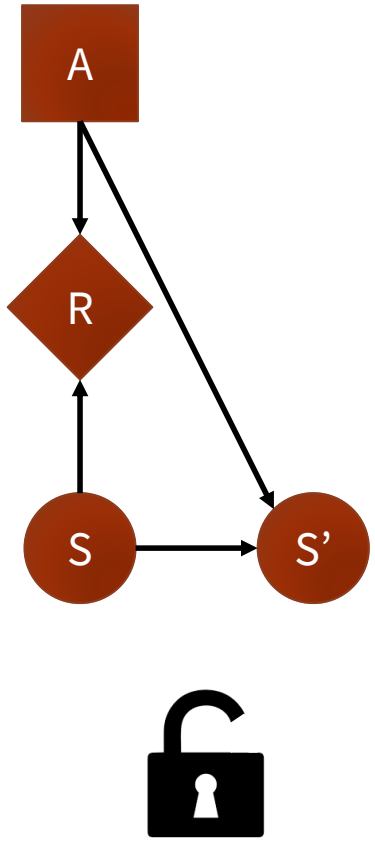
ACAS Xa

Next-gen system using Markov Decision Process
Expected deployment 2020



VerticalCAS

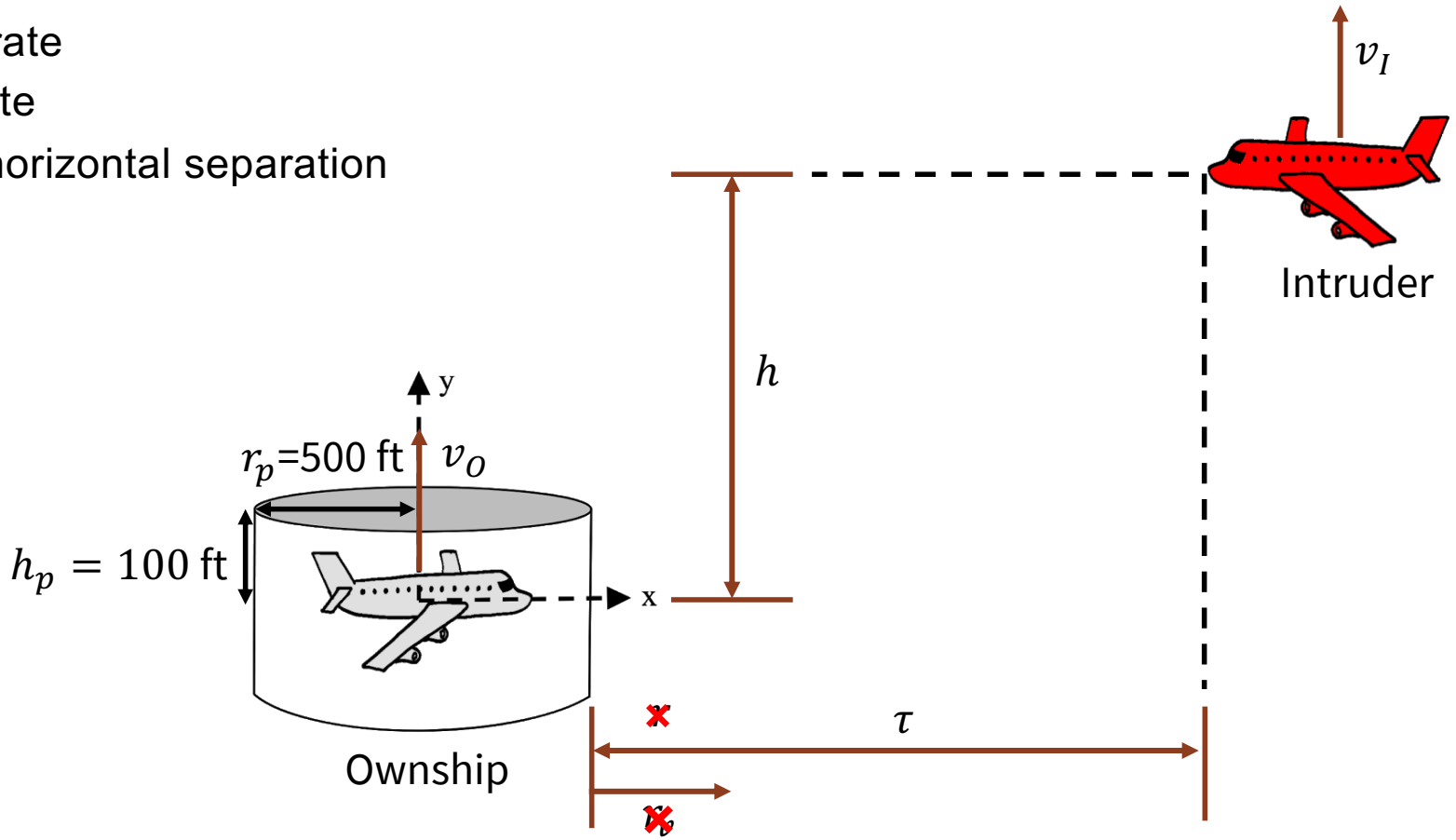
Notional System inspired by ACAS Xa
Can share widely



VerticalCAS Formulation

State variables – Describe the encounter:

1. h (ft): Relative altitude of intruder
2. v_O (ft/s): Ownship climbrate
3. v_I (ft/s): Intruder climbrate
4. τ (sec): Time to loss of horizontal separation
5. a_{prev}: Previous advisory



VerticalCAS Formulation

Actions – Advisories the system gives pilots every ϵ seconds (nominally $\epsilon = 1$):

1. **DES1500**: Descend at least 1500 ft/min
2. **CL1500**: Climb at least 1500 ft/min
3. **DNC**: Do Not Climb
4. **DND**: Do Not Descend

$$a \in [g/4, g/2]$$

5. **SDES1500**: Strengthen Descent to at least 1500 ft/min
6. **SCL1500**: Strengthen Climb to at least 1500 ft/min
7. **SDES2500**: Strengthen Descent to at least 2500 ft/min
8. **SCL2500**: Strengthen Climb to at least 2500 ft/min

$$a \in [g/3, g/2]$$

9. **COC**: Clear of Conflict

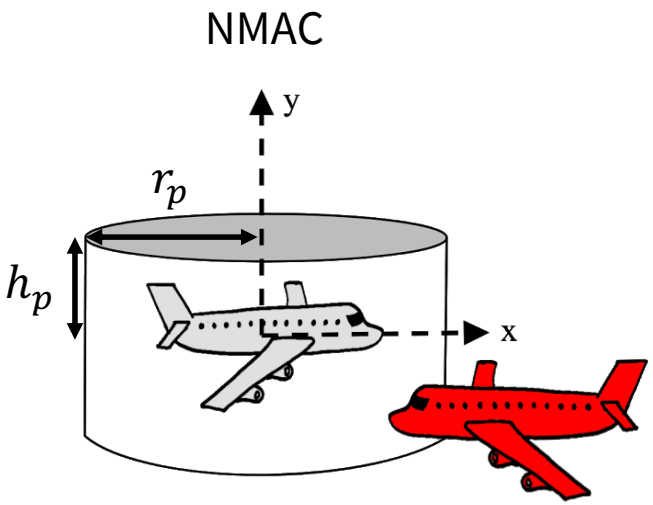
$$a \in [-g/2, g/2]$$

Vertical Speed Indicator

VerticalCAS Formulation

Rewards – Penalties to discourage bad behaviors:

- Near mid-air collisions (NMAC)
- Prohibited advisory transitions
- Reversal
- Strengthening
- Non-COC advisories



VerticalCAS Solution

Discrete Value Iteration [DMU'15]

Table: $Q(h, v_O, v_I, \tau, a_{\text{prev}}, a)$

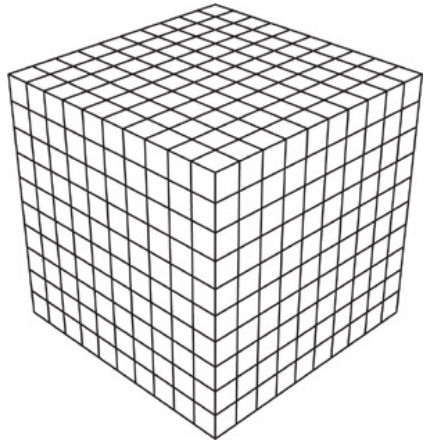
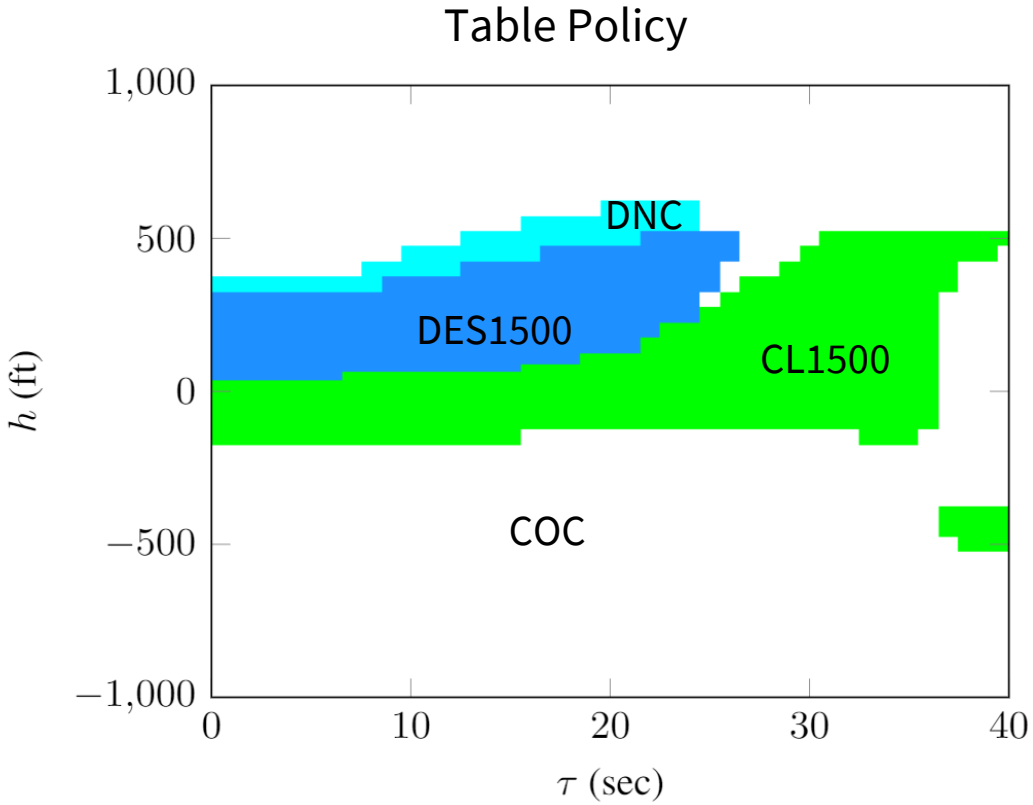
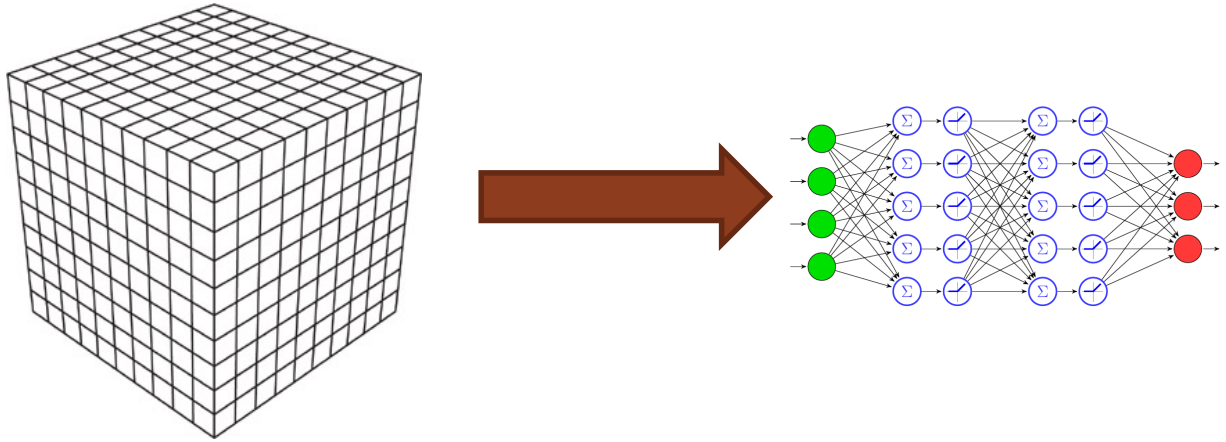


Table Policy: $\underset{a}{\text{argmax}}\, Q(h, v_O, v_I, \tau, a_{\text{prev}}, a)$

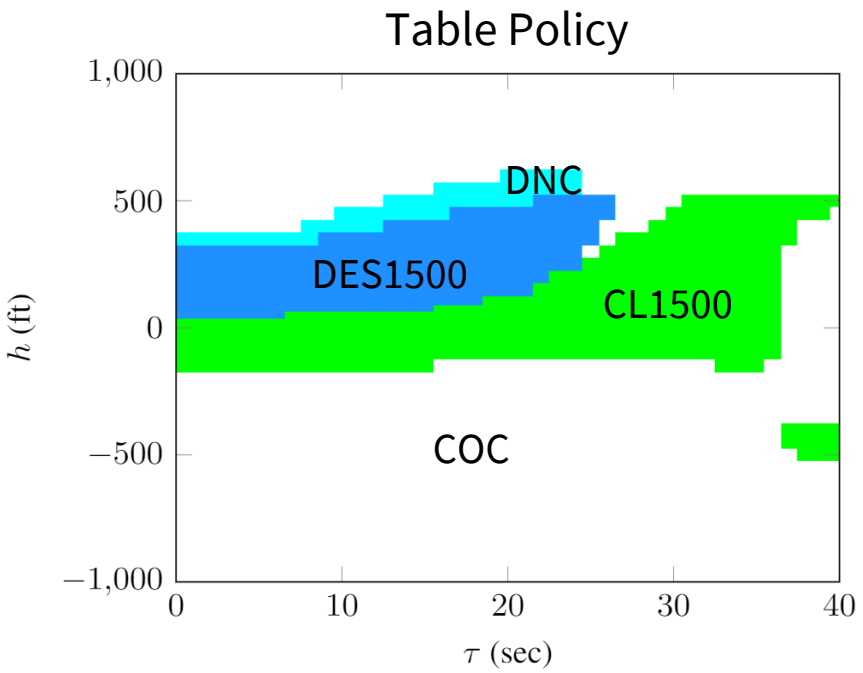


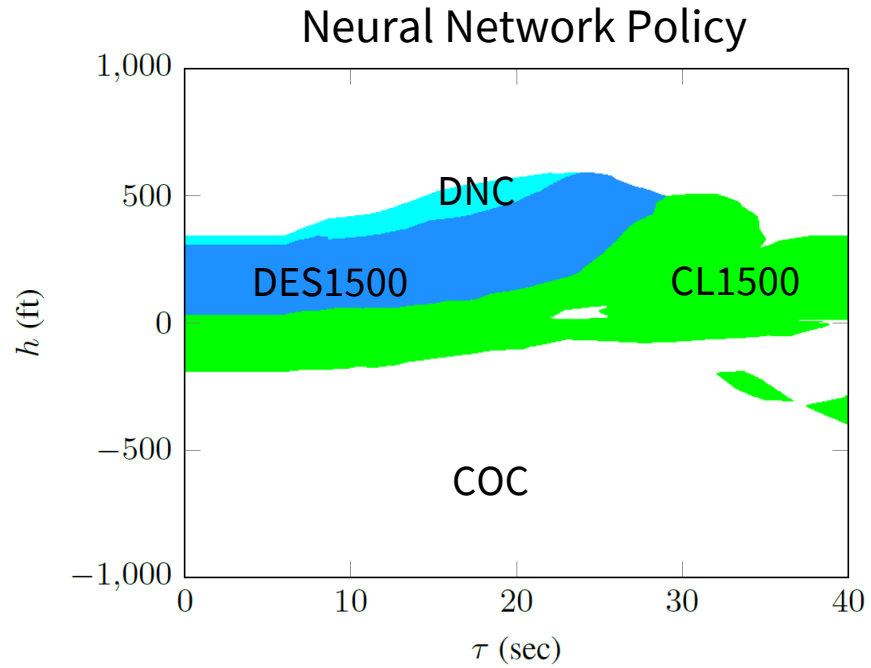
[DMU'15] Kochenderfer. Optimized airborne collision avoidance. In *Decision Making Under Uncertainty: Theory and Application.*

VerticalCAS Neural Networks



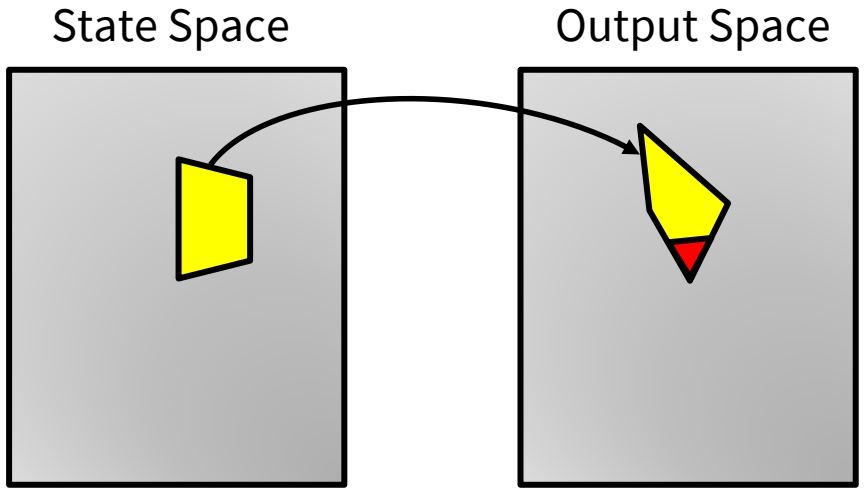
- One network per a_{prev}
- 4 inputs
- 6 hidden layers with 45 units each
- 9 outputs
- Relu activations

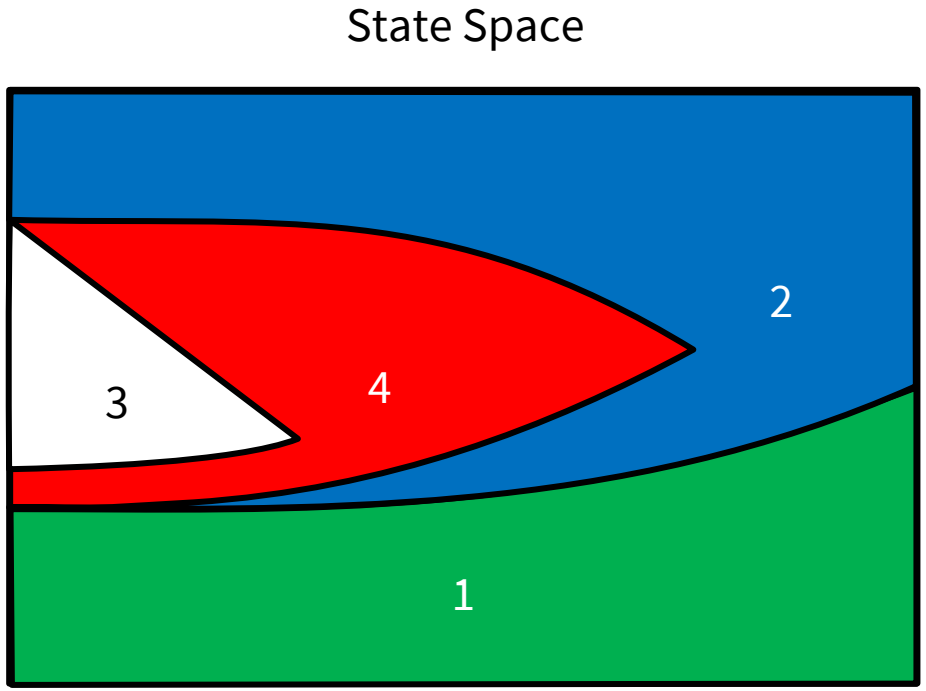




Verification Overview

1. Define **safe** region for an advisory
2. Define **safeable** region for an advisory [STTT'17]
3. Define region where no advisory is safeable
4. Compute region where an advisory is not safeable but a safeable advisory exists
5. Use neural network verification tool to verify safeable (Reluplex)

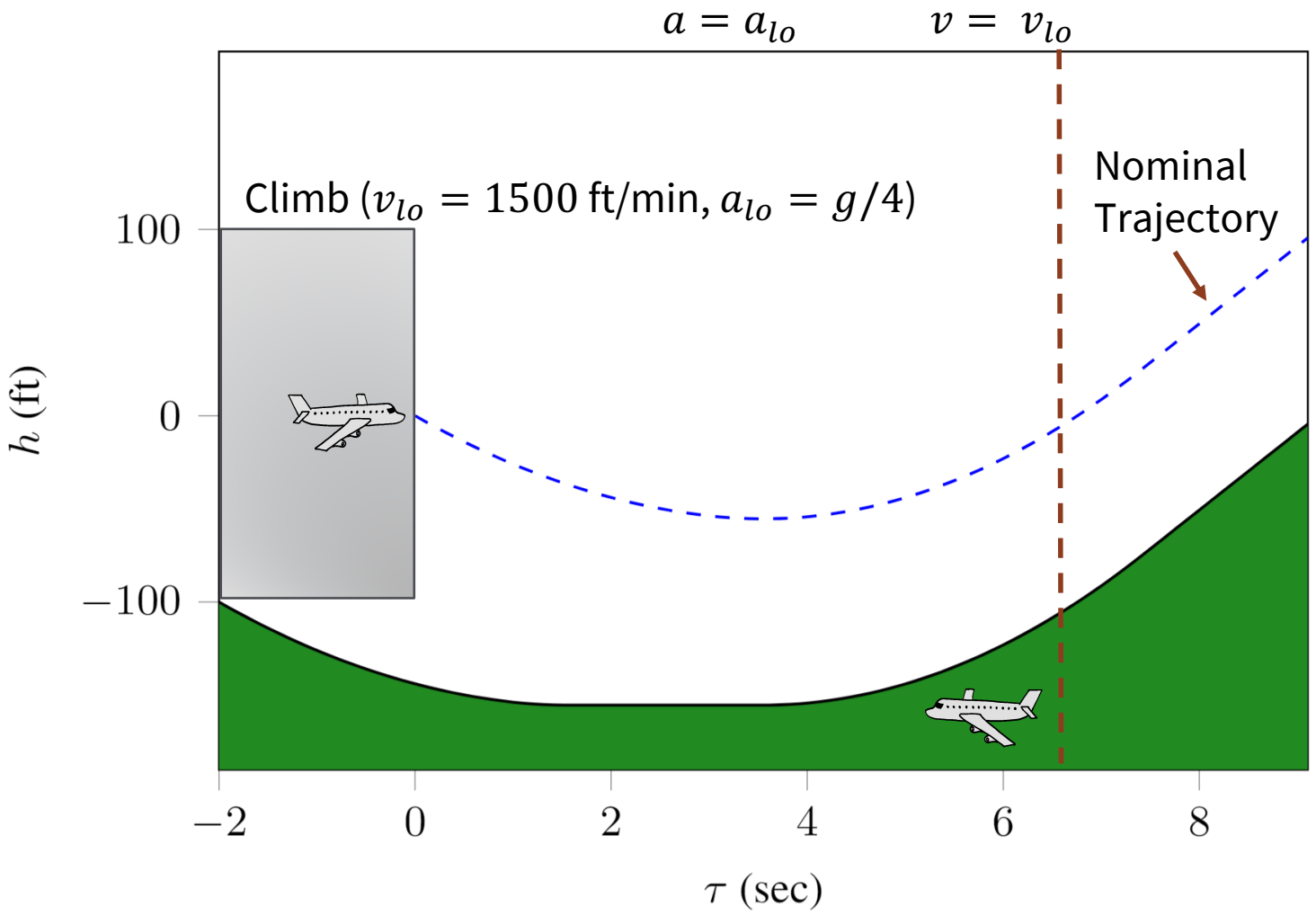




[STTT'17] Jeannin et al. A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system. *International Journal on Software Tools for Technology Transfer.*

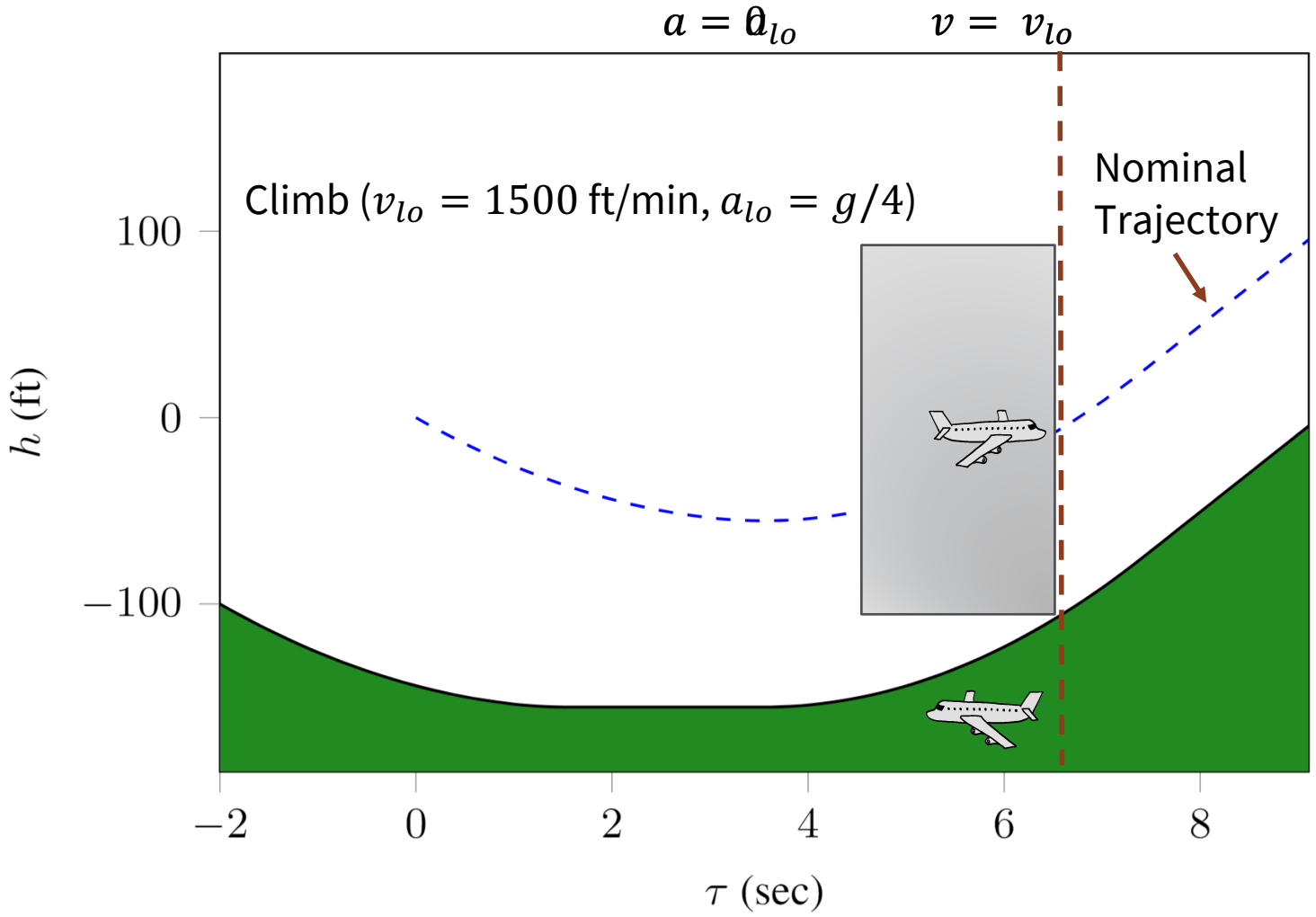
Safe Regions

- Region where an intruder aircraft is safe to exist



Safe Regions

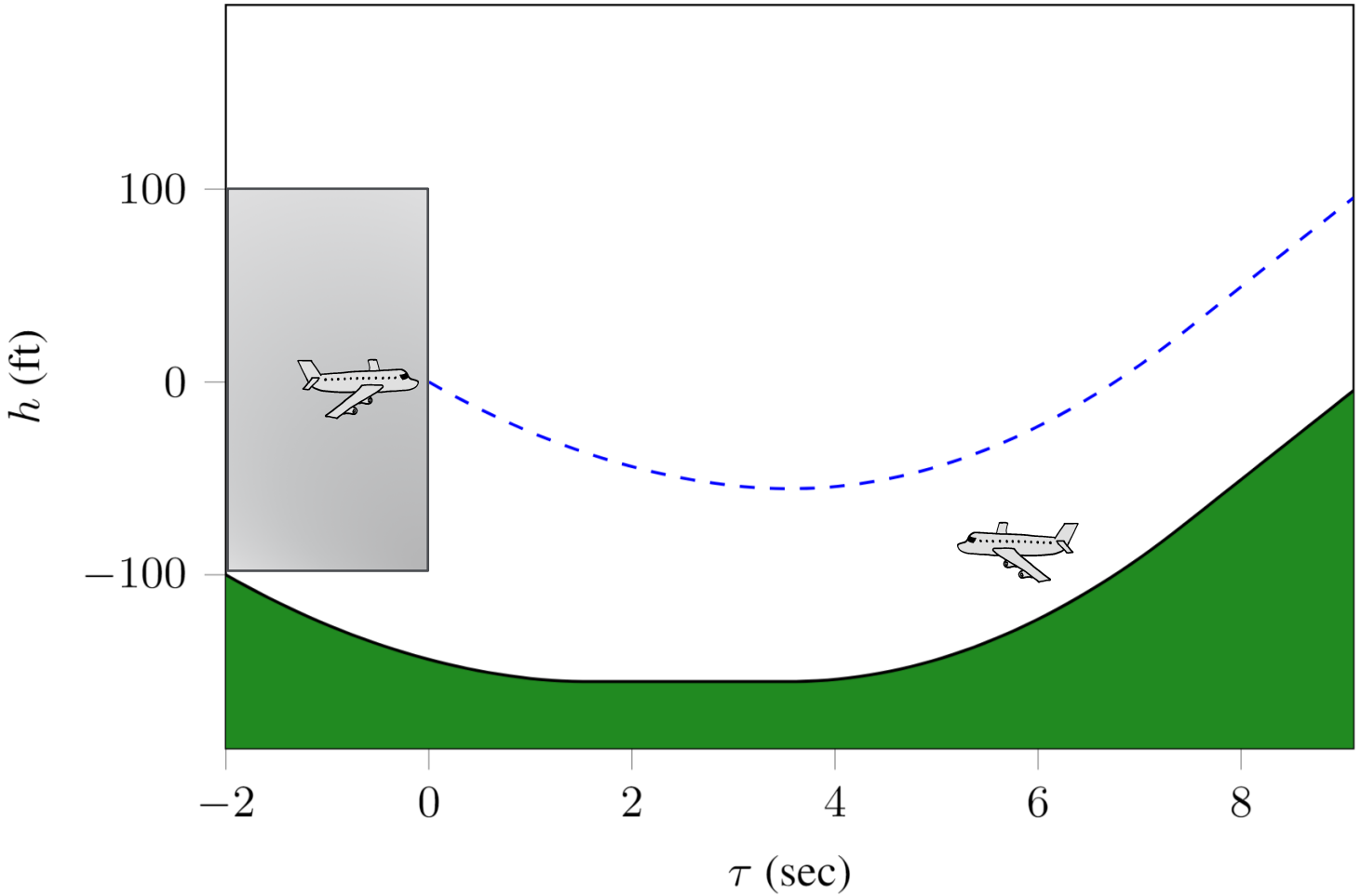
- Region where an intruder aircraft is safe to exist



Safe Regions

- Region where an intruder aircraft is safe to exist

$$\Omega_{unsafe} = (\Omega_{safe})^c$$

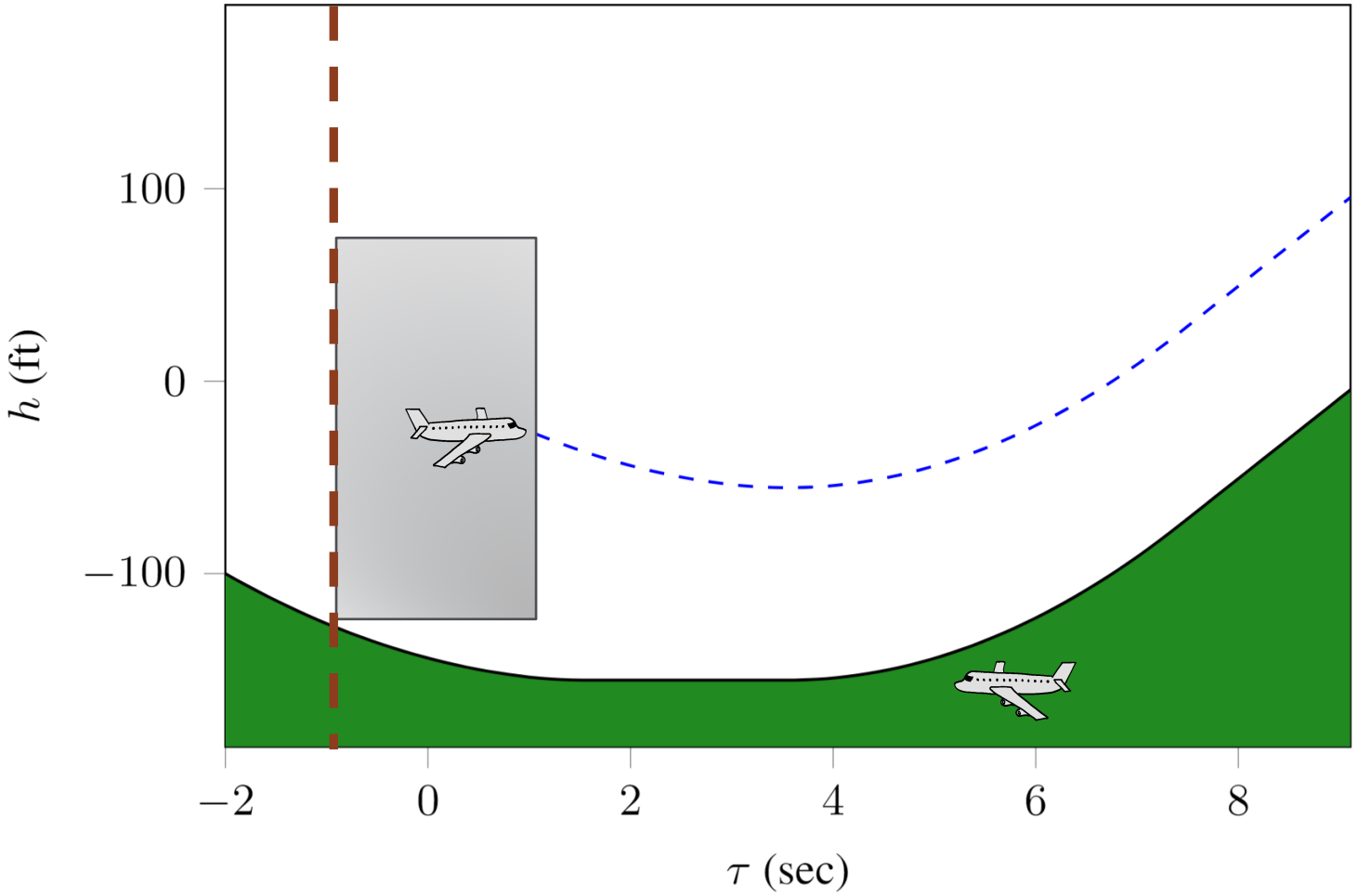


Safe Regions – Revisiting τ

- Relative velocity, r_v, dictates width of ownship puck given by

$$\frac{2r_p}{r_v}$$

- However, we don't know r_v

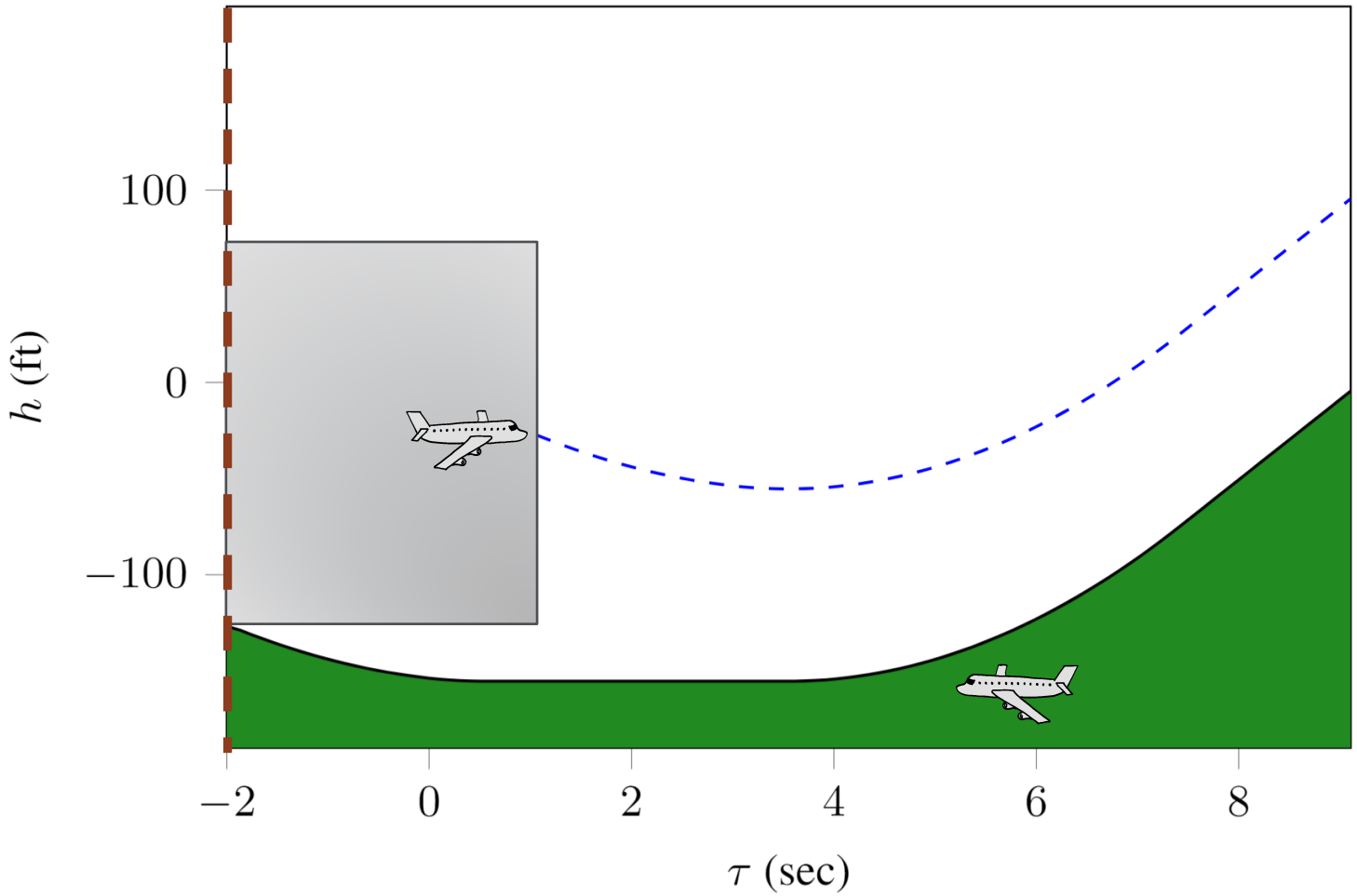


Safe Regions – Revisiting τ

- Relative velocity, r_v, dictates width of ownship puck given by

$$\frac{2r_p}{r_v}$$

- However, we don't know r_v
- As r_v decreases, the puck gets wider

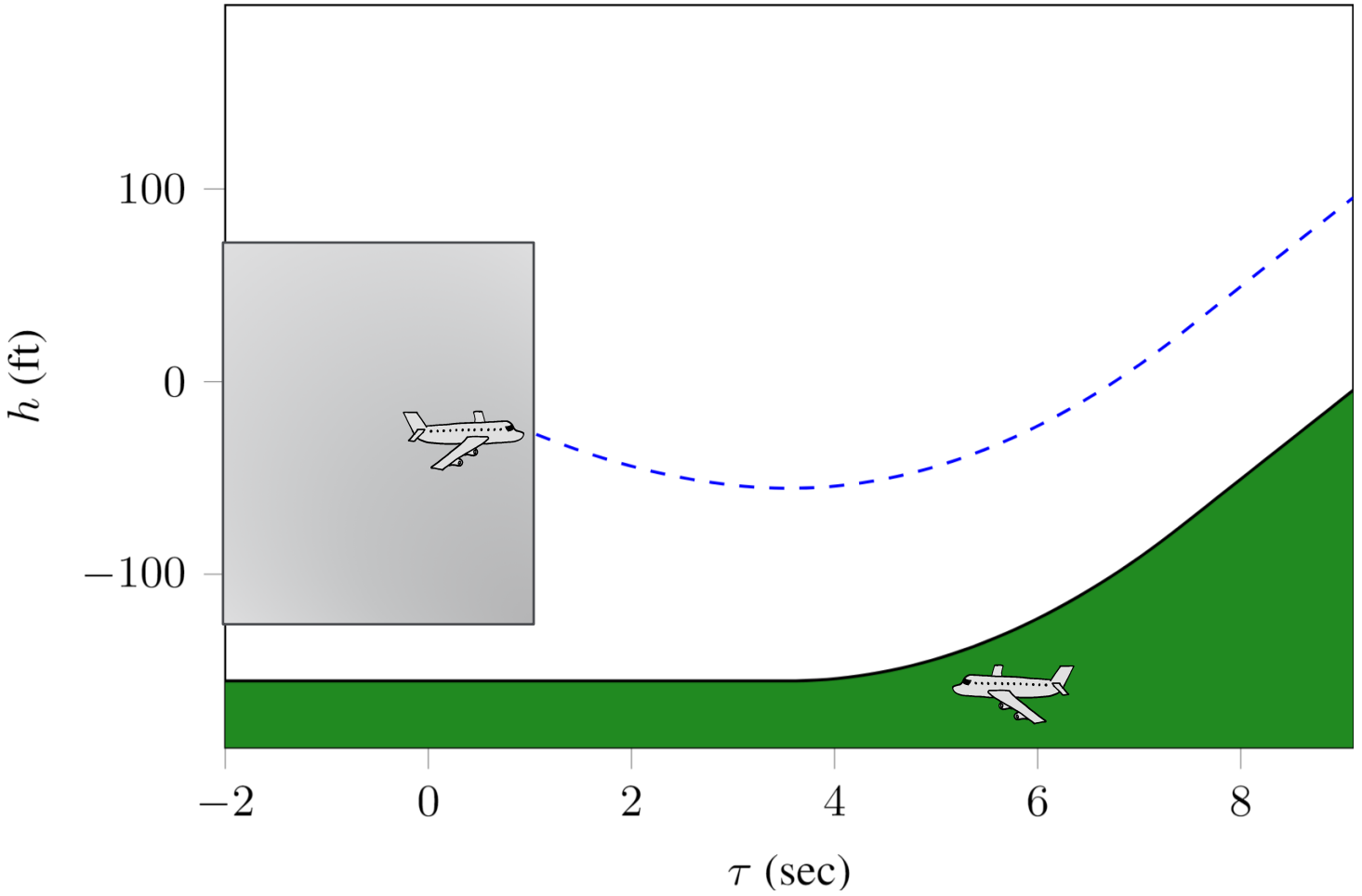


Safe Regions – Revisiting τ

- Relative velocity, r_v, dictates width of ownship puck given by

$$\frac{2r_p}{r_v}$$

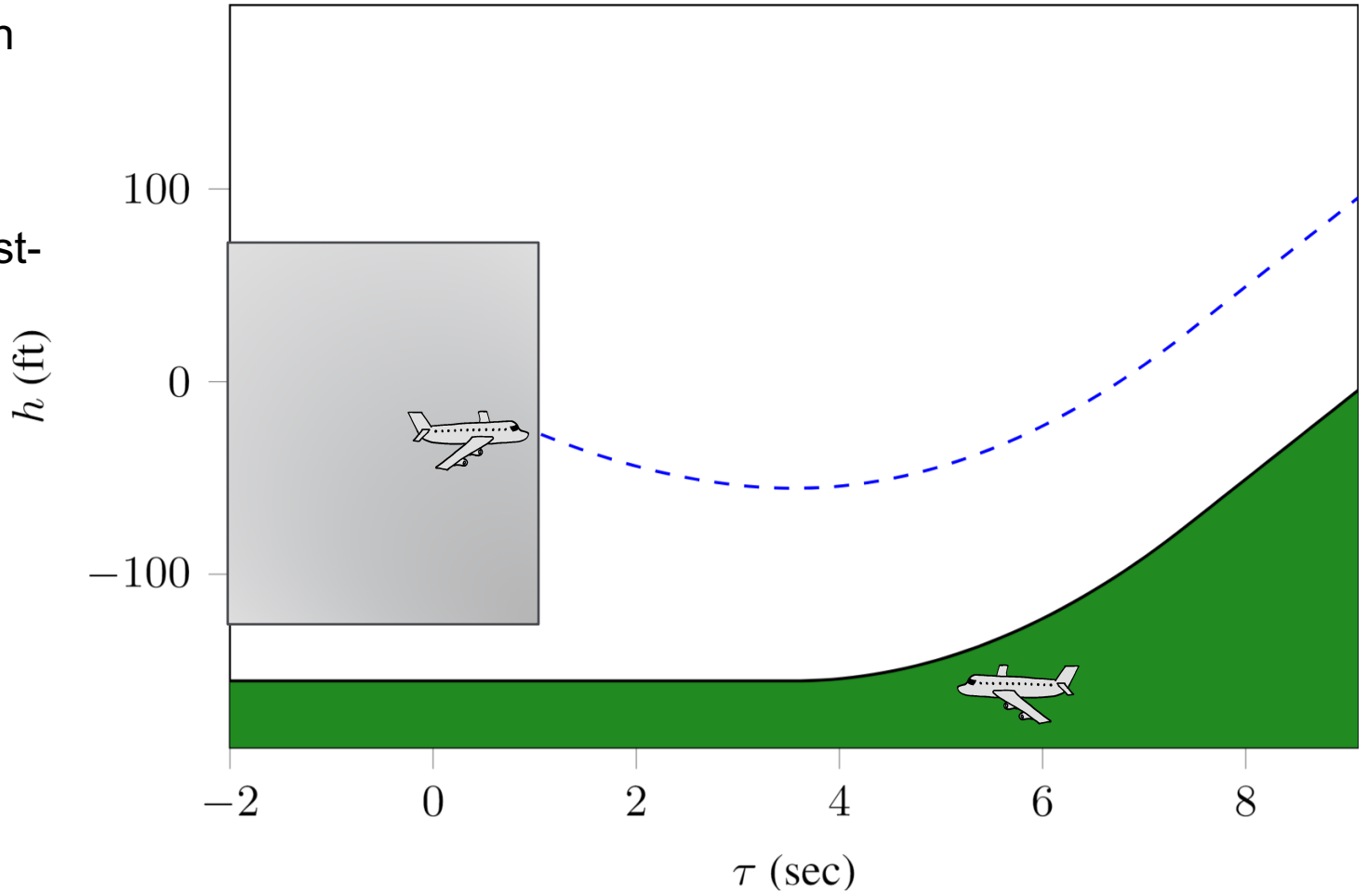
- However, we don't know r_v
- As r_v decreases, the puck gets wider
- As $r_v \to 0$, $width \to \infty$



Worst-Case Approach

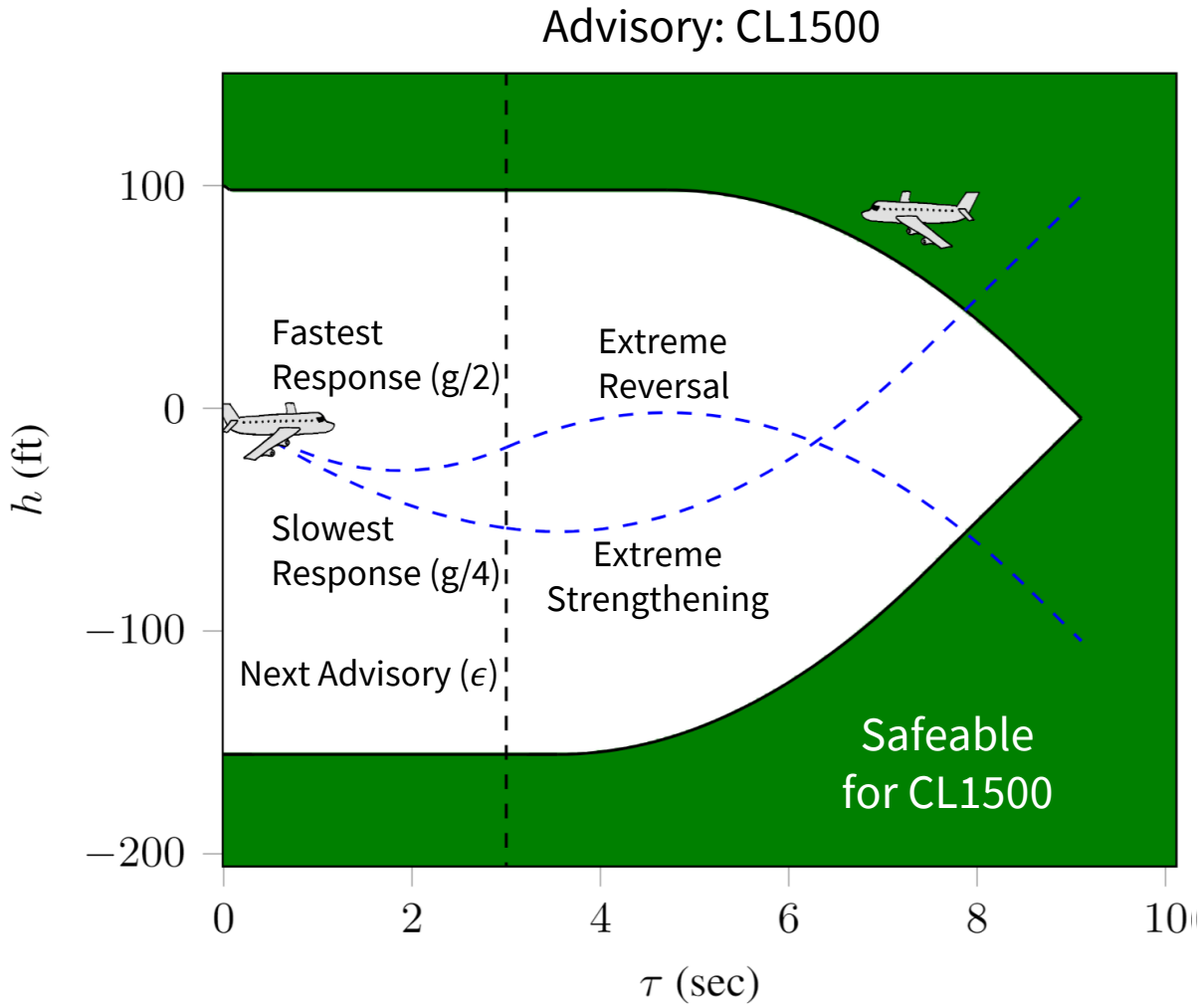
- For $r_v \to 0$ the unsafe region includes all other unsafe regions
- Using this, we define a worst-case safe region where,

$\Omega_{unsafe} \subset \Omega_{unsafe\,(worstcase)}$



Safeable Regions

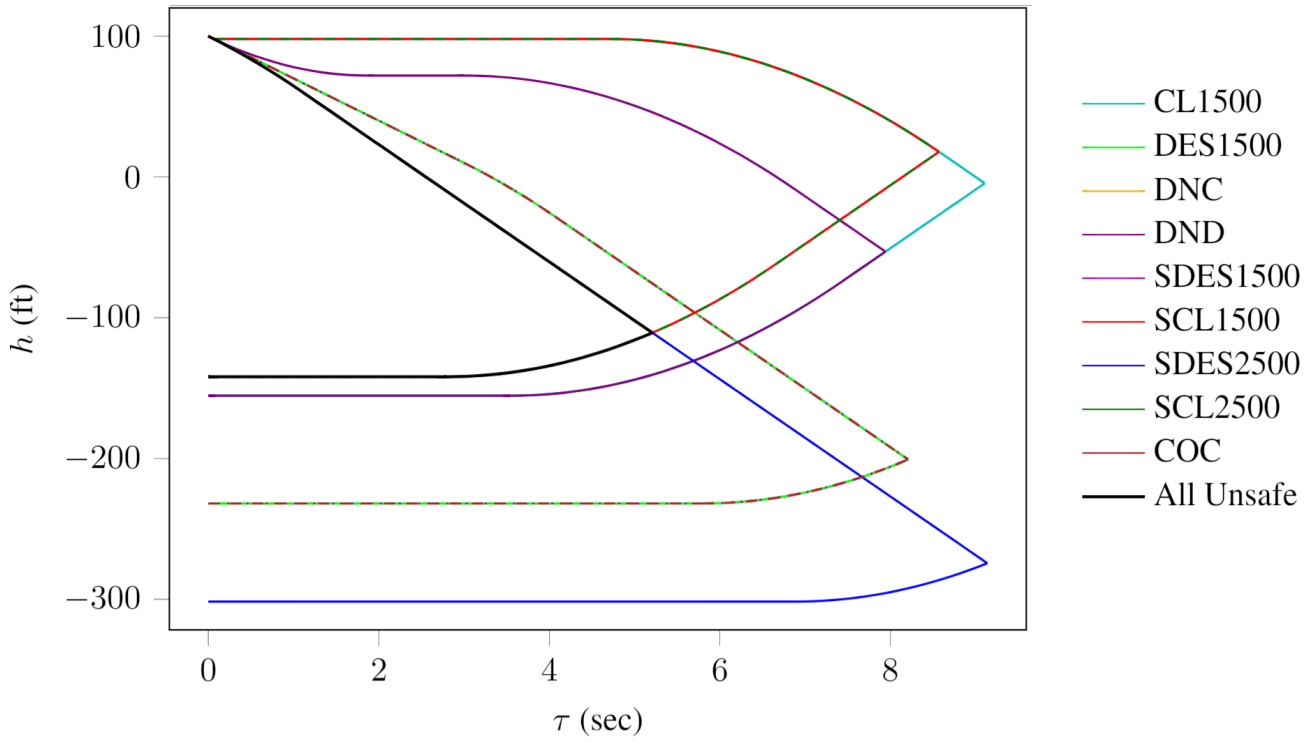
- Safeable addresses extends safe regions to address limitations
- Allow system to change advisory issued every ϵ seconds
- Consider two most extreme positions after ϵ seconds
- If system always issues safeable advisories, then an intruder beginning in a safeable region will remain in one, and safety is guaranteed [STTT '17]



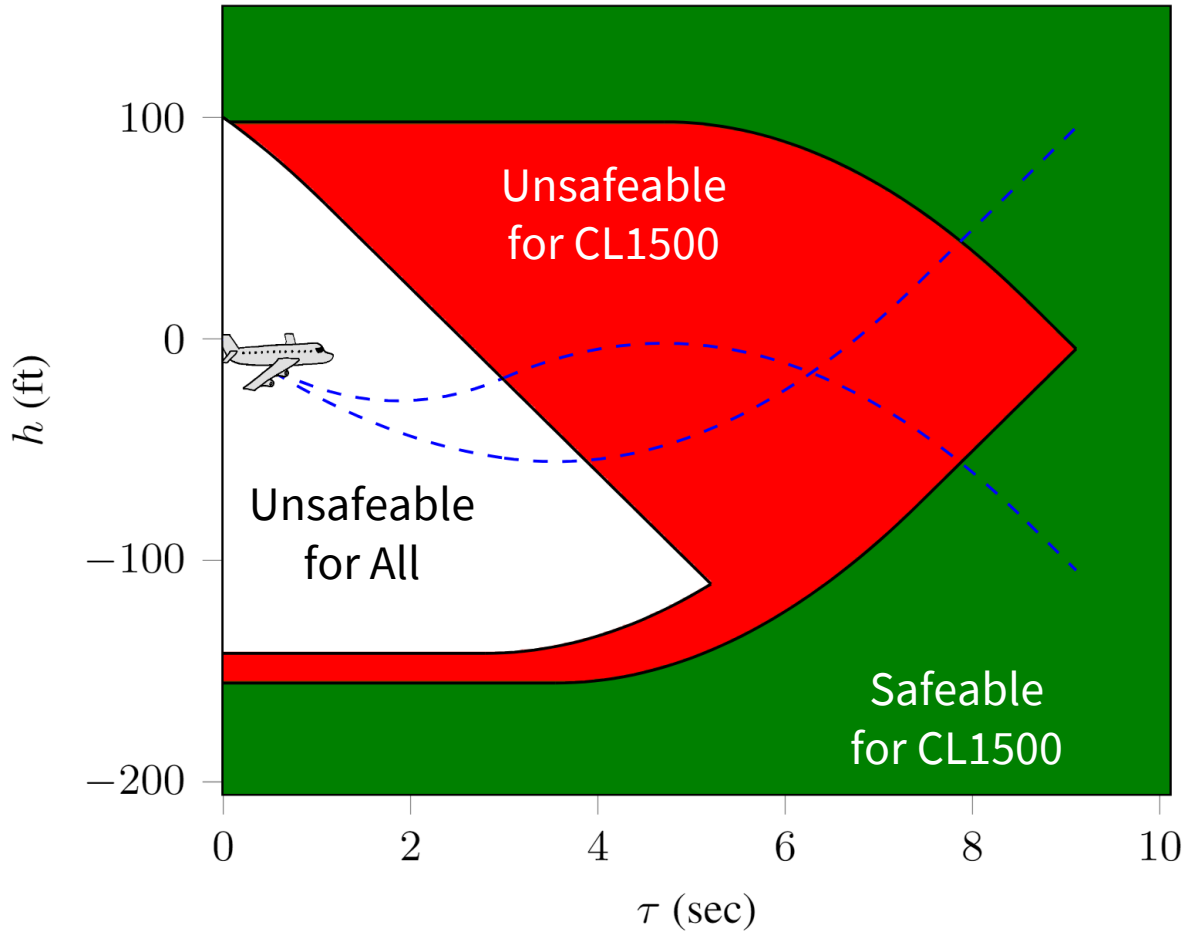
[STTT '17] Jeannin, et. al. A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system.

Search Region

- Want to search network in regions where advisory is not safeable, but is still issued
- But we want to omit regions that are unsafeable for all advisories

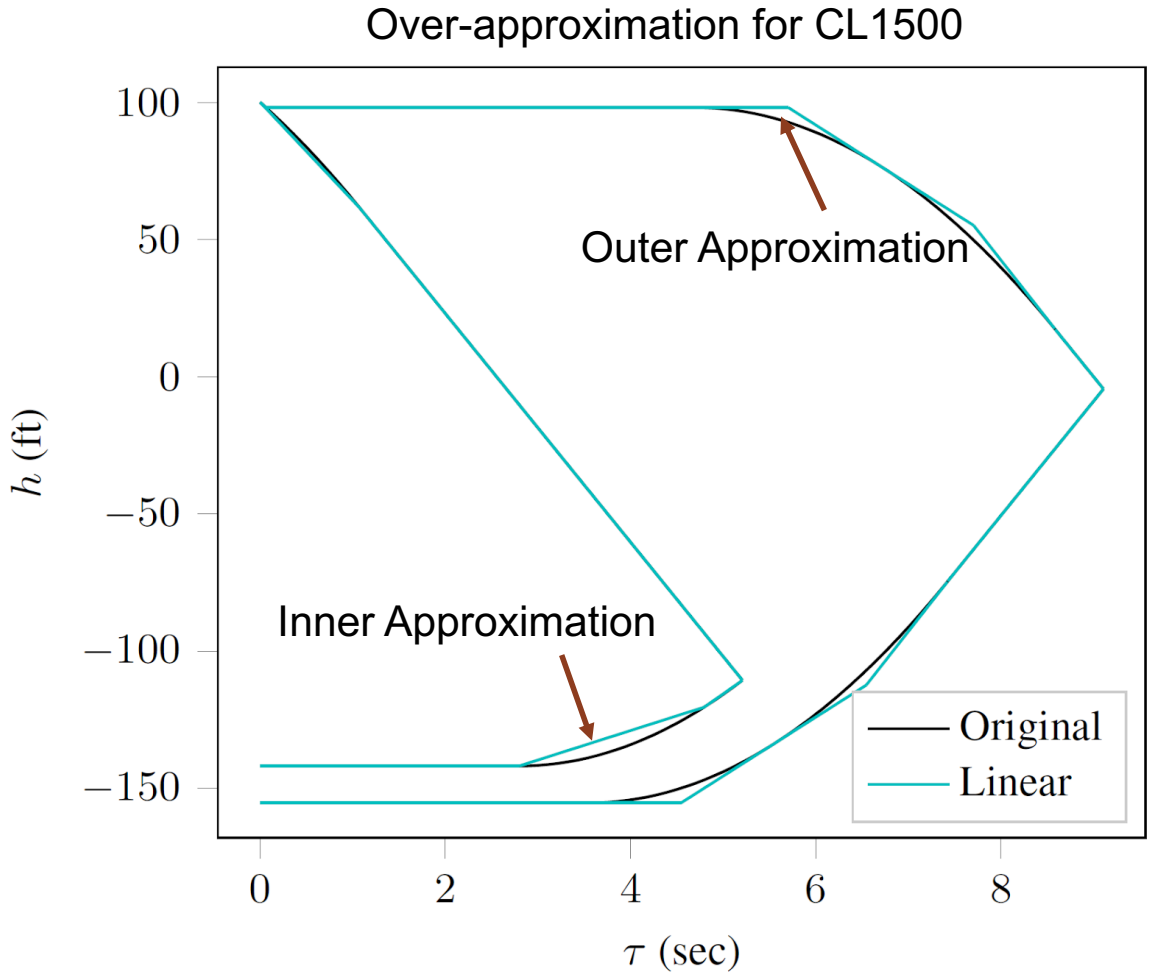


Search Region



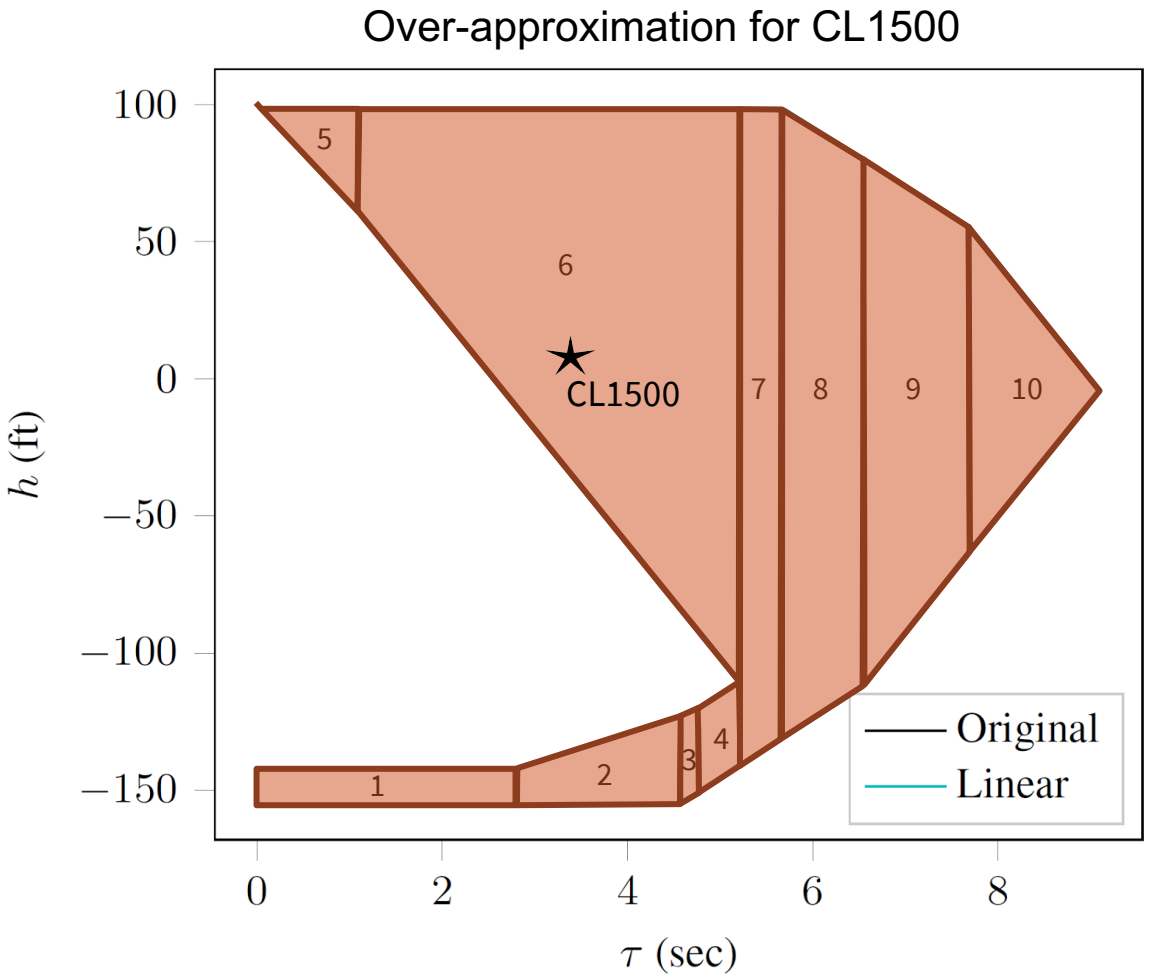
Linearization

- Search region needs to be linear for Reluplex
- If over-approximated region is verified, actual region is also verified (no false negatives)
- If under-approximated region is verified, all counter-examples found are real (no false positives)



Using Reluplex

- Non-convex search region is split into smaller convex regions
- Each small region is checked by Reluplex as a separate query
- Network inputs producing an unsafeable advisory when a safeable advisory exists is a counter-example to safe behavior



Results

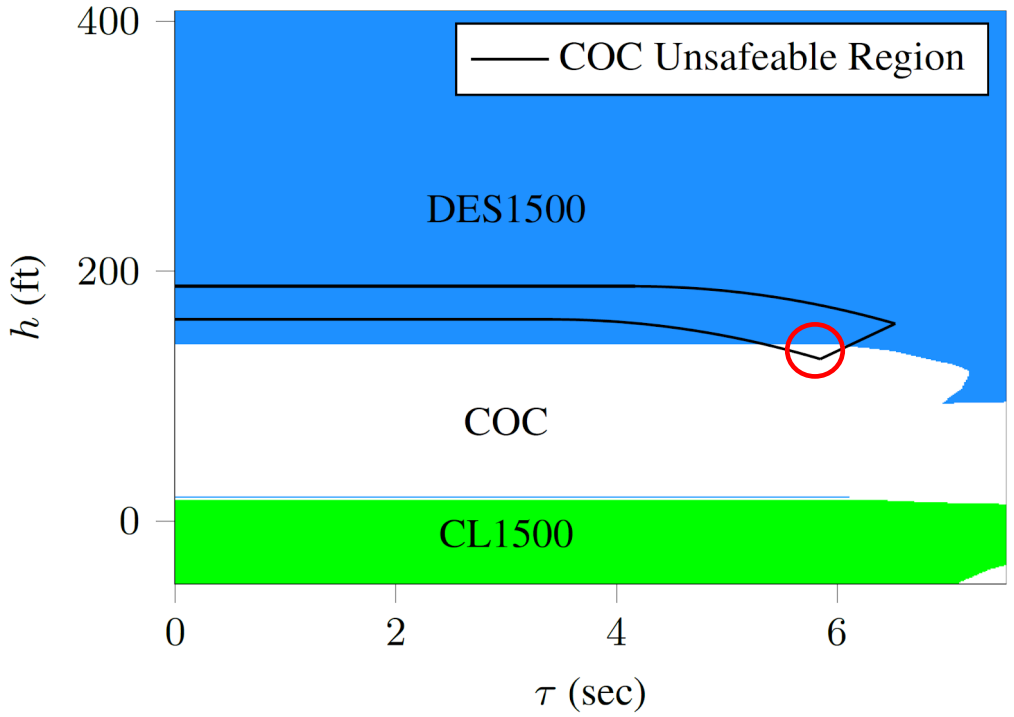
- Out of 42,032 queries, 9.14% resulted in a counter-example
- The degree of linearization had no effect on number of counter-examples found

Table: Number of counterexamples

	Current Advisory								
Previous Advisory	COC	DNC	DND	DES1500	CL1500	SDES1500	SCL1500	SDES2500	SCL2500
COC	359	28	0	48	21	N/A	N/A	N/A	N/A
DNC	438	30	0	40	47	N/A	N/A	N/A	N/A
DND	249	0	17	133	50	N/A	N/A	N/A	N/A
DES1500	284	0	1	1	0	65	76	N/A	N/A
CL1500	223	0	0	0	0	117	21	N/A	N/A
SDES1500	281	0	0	0	0	26	6	32	65
SCL1500	238	0	3	0	0	53	66	43	51
SDES2500	324	0	0	0	0	12	1	89	25
SCL2500	209	0	12	0	0	52	15	48	58

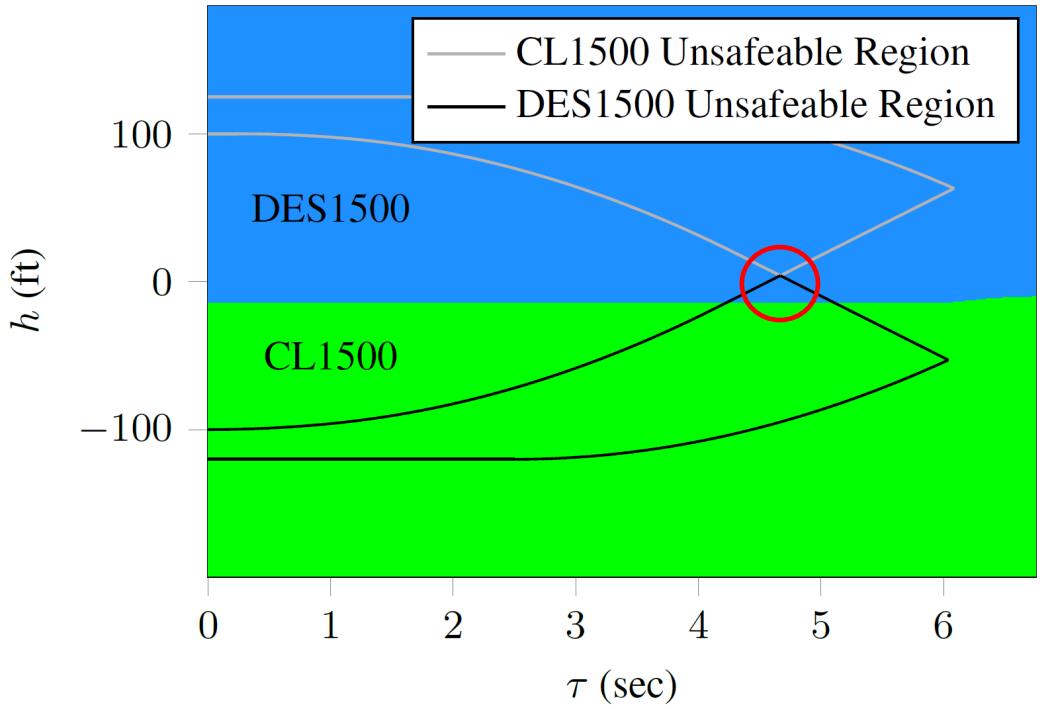
Investigating Counterexamples

- Informative counter-examples reveal problems in the network
- COC advisory issued in its unsafeable region



Investigating Counterexamples

- Many counter-examples are found at the boundary of regions
- To be completely safeable, the alerting boundary would have to pass exactly through the intersection of safeable regions
- Not possible for neural networks to achieve this precision



Future Work

- Address the limitations of the approach that force networks to be highly precise
- Explore the idea of safeable2 that considers a region verified only if it is safeable for at least 2 advisories
- Use the counter-examples to improve the network