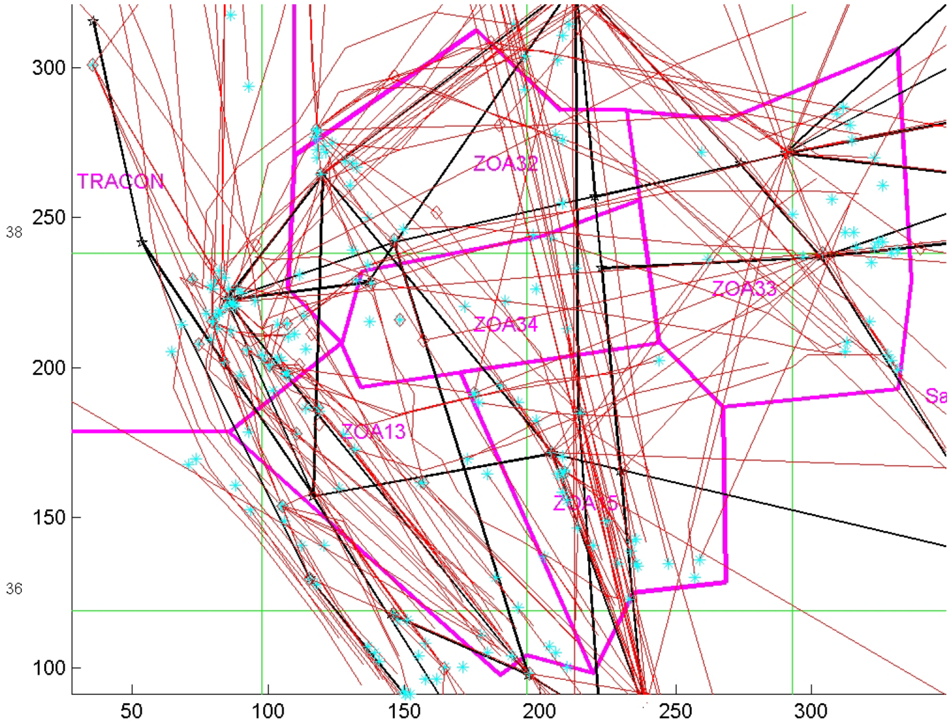
Reachability and Learning for Hybrid Systems

Kene Akametalu, Jaime Fisac, Claire Tomlin

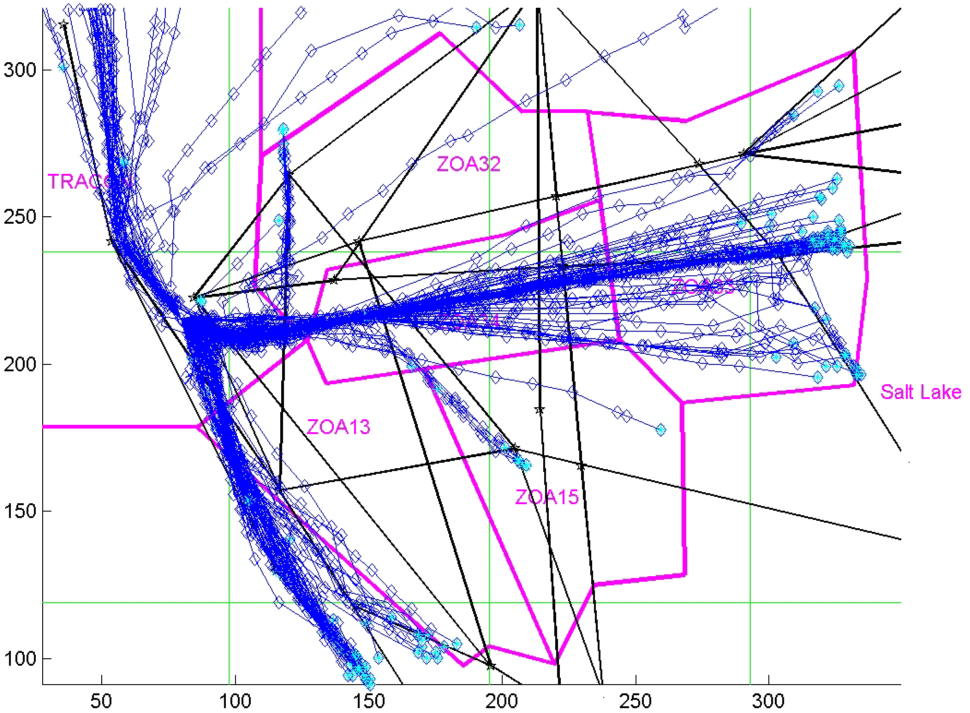
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley

June 9 2015

Air traffic in Oakland Center



cross traffic

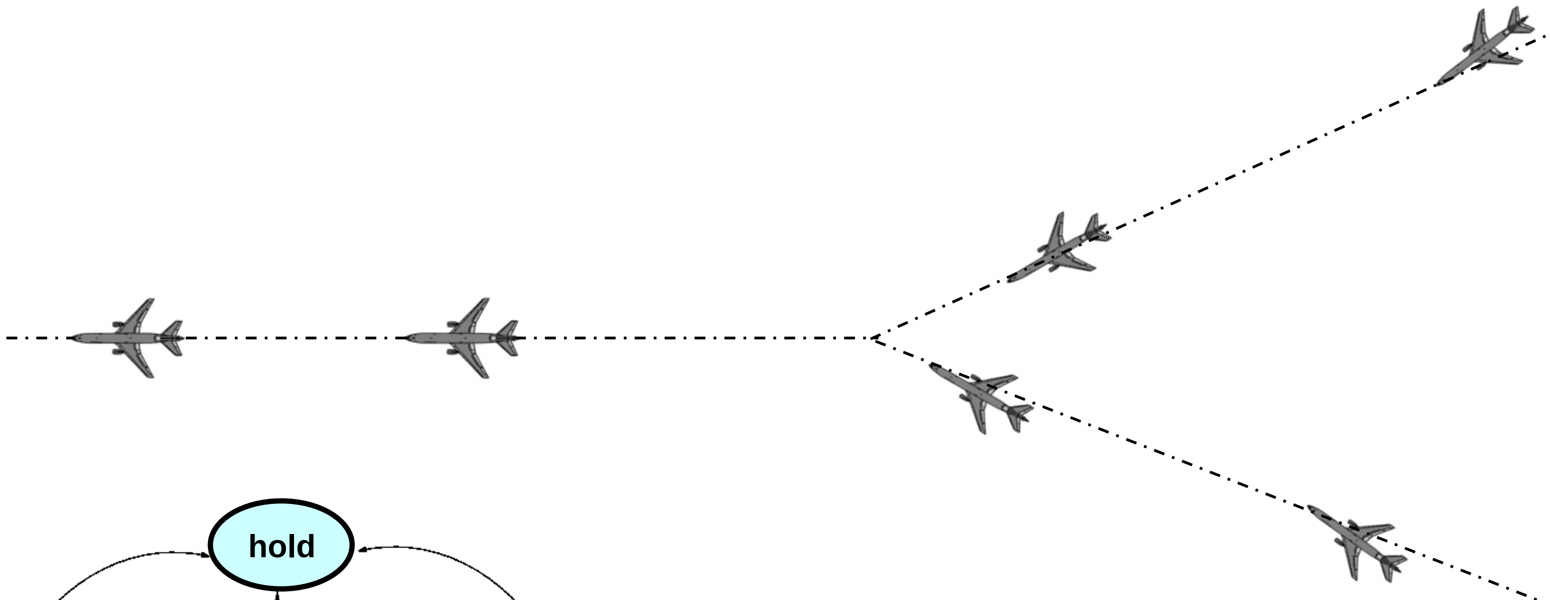


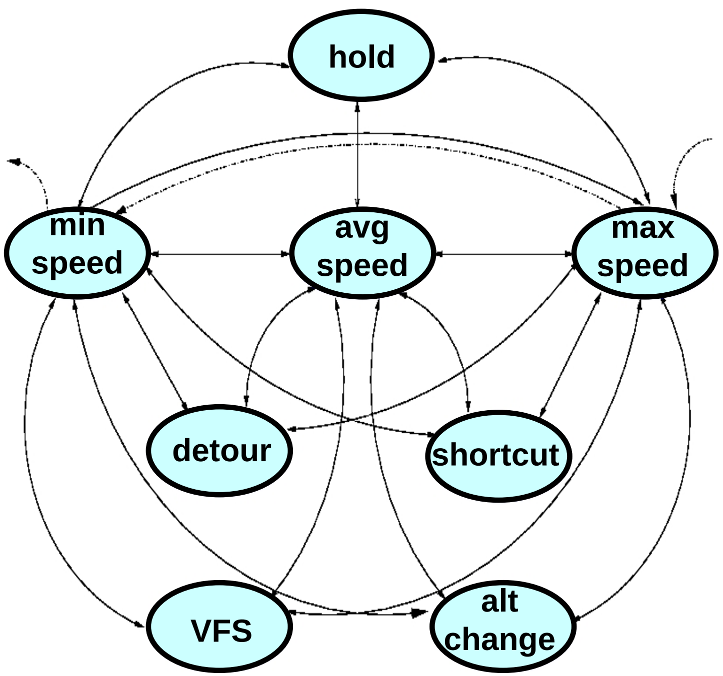
merging traffic

- Safety critical: 1000 ft, 5 nmi separation
- Standard corridors of well-travelled routes

[NASA ETMS, FACET]

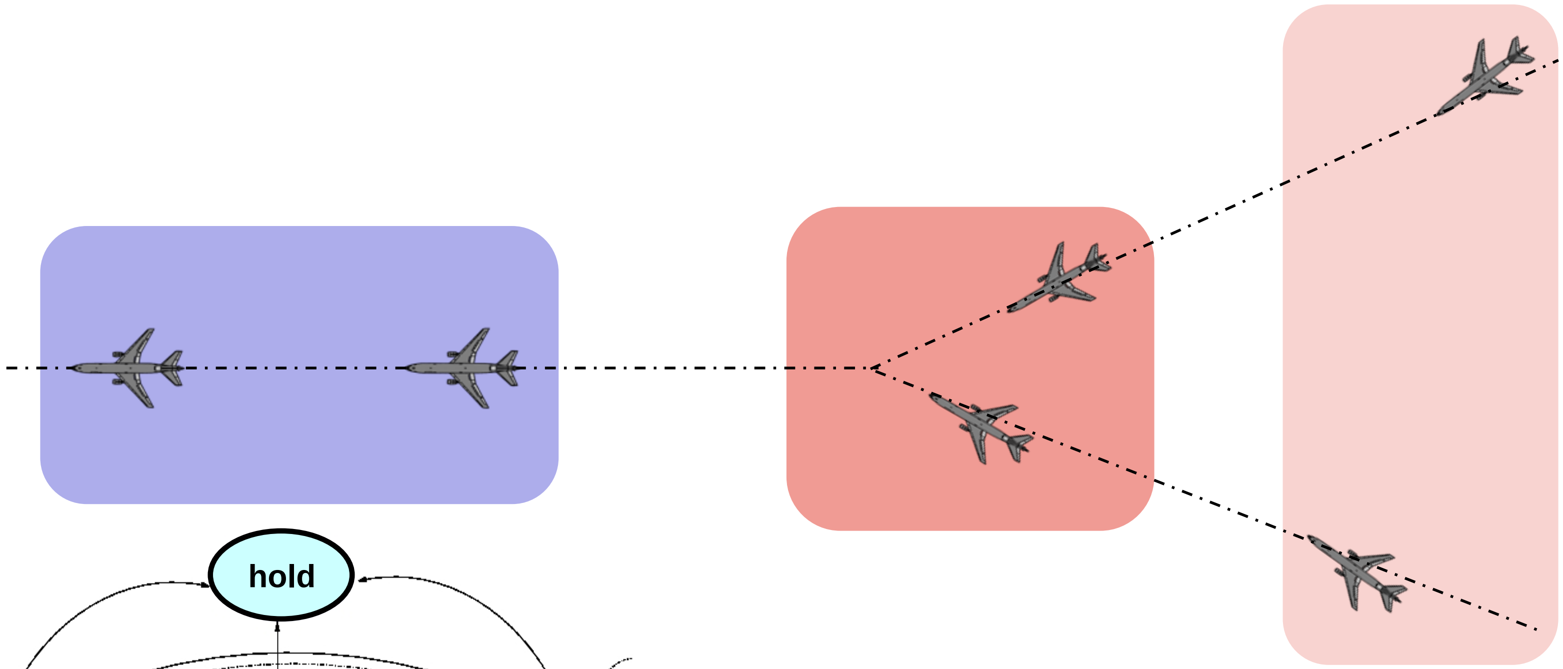
Controller must keep aircraft separated

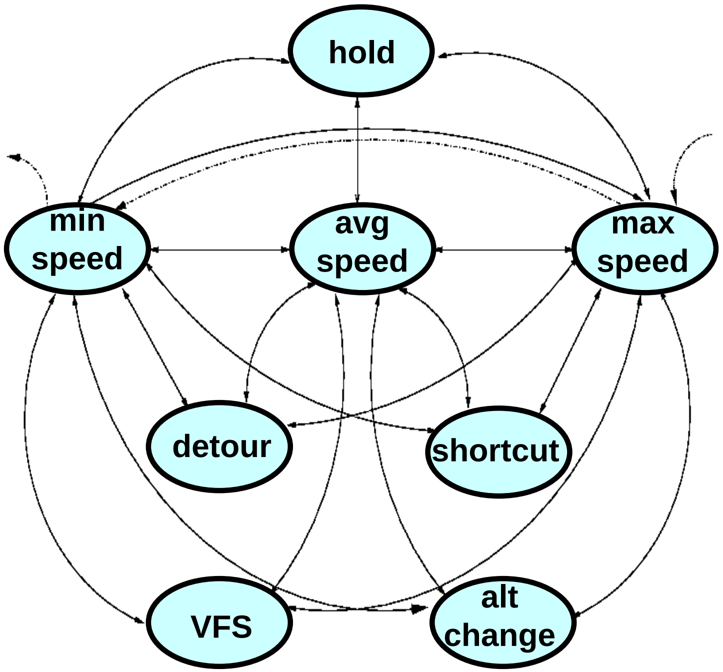




- Small set of control actions
- Infrequent deviations from nominal
- Grouping by potential conflict

Controller must keep aircraft separated



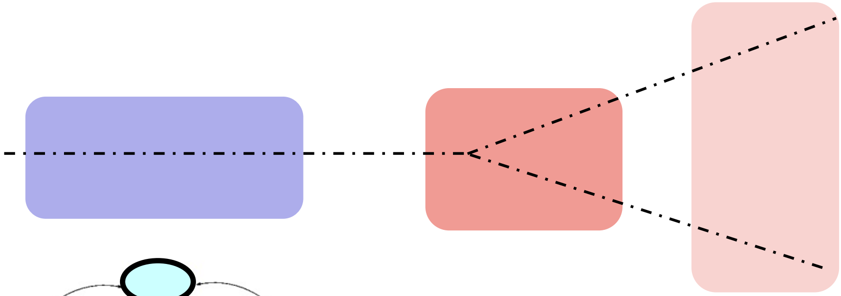


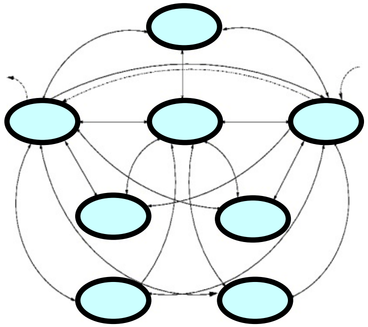
- Small set of control actions
- Infrequent deviations from nominal
- Grouping by potential conflict

Growing numbers of UAV applications

[Amazon]

[Google]



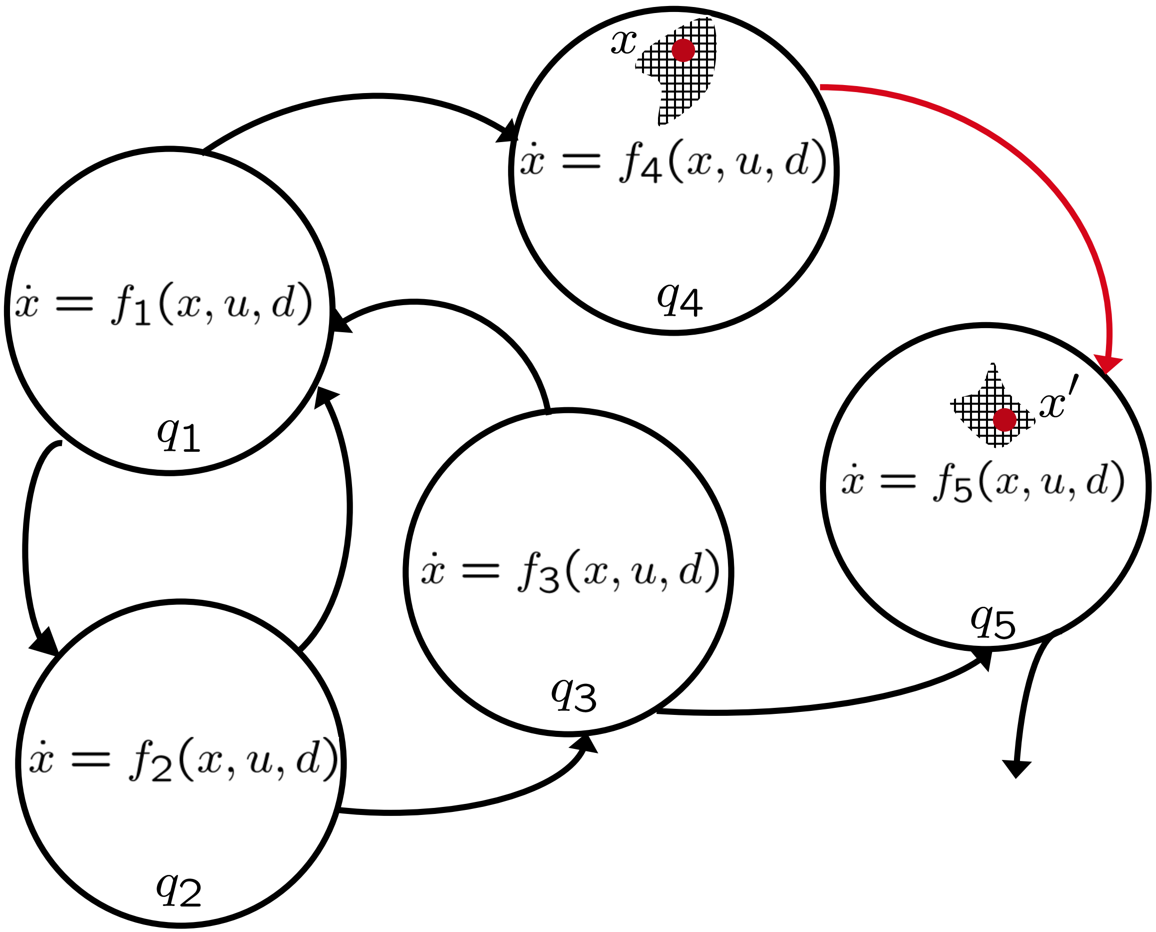


1. Safety
2. Simplicity
3. Ability to adapt to new information

[NASA]

- Collision avoidance system
- Forced landing system

Hybrid System Model

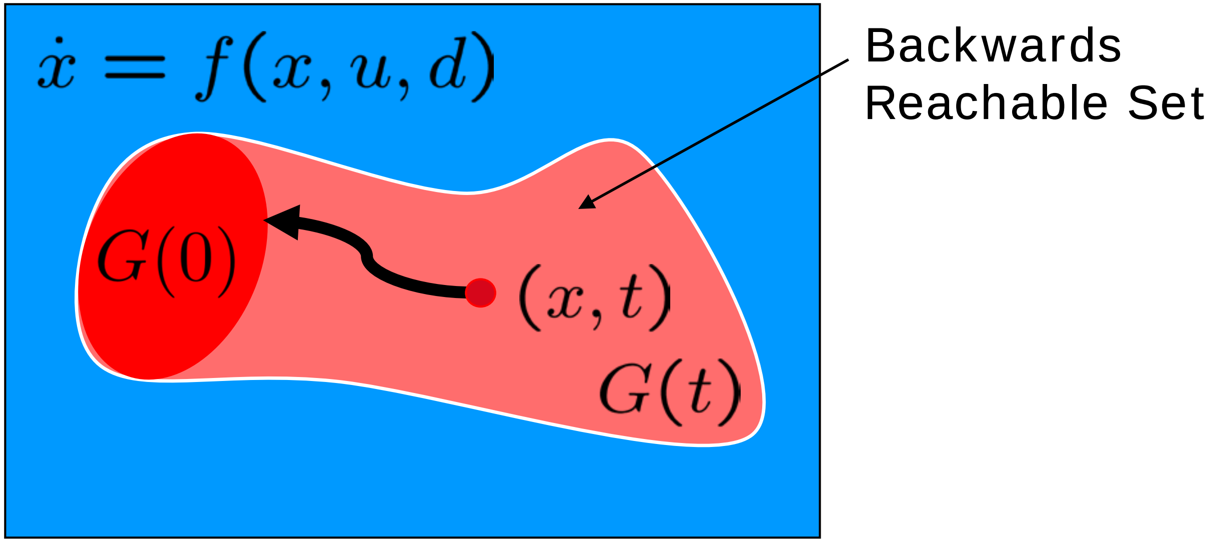


Outline

- Reachable sets for hybrid systems
  - Overview
  - Examples:
    - Collision avoidance
    - Mode sequencing
- Learning dynamic behavior safely
  - Overview
  - Examples:
    - Learning to fly

Backwards Reachable Set

All states for which, for all possible control actions, there is a disturbance action which can drive the system state into a region $G(0)$ in time t



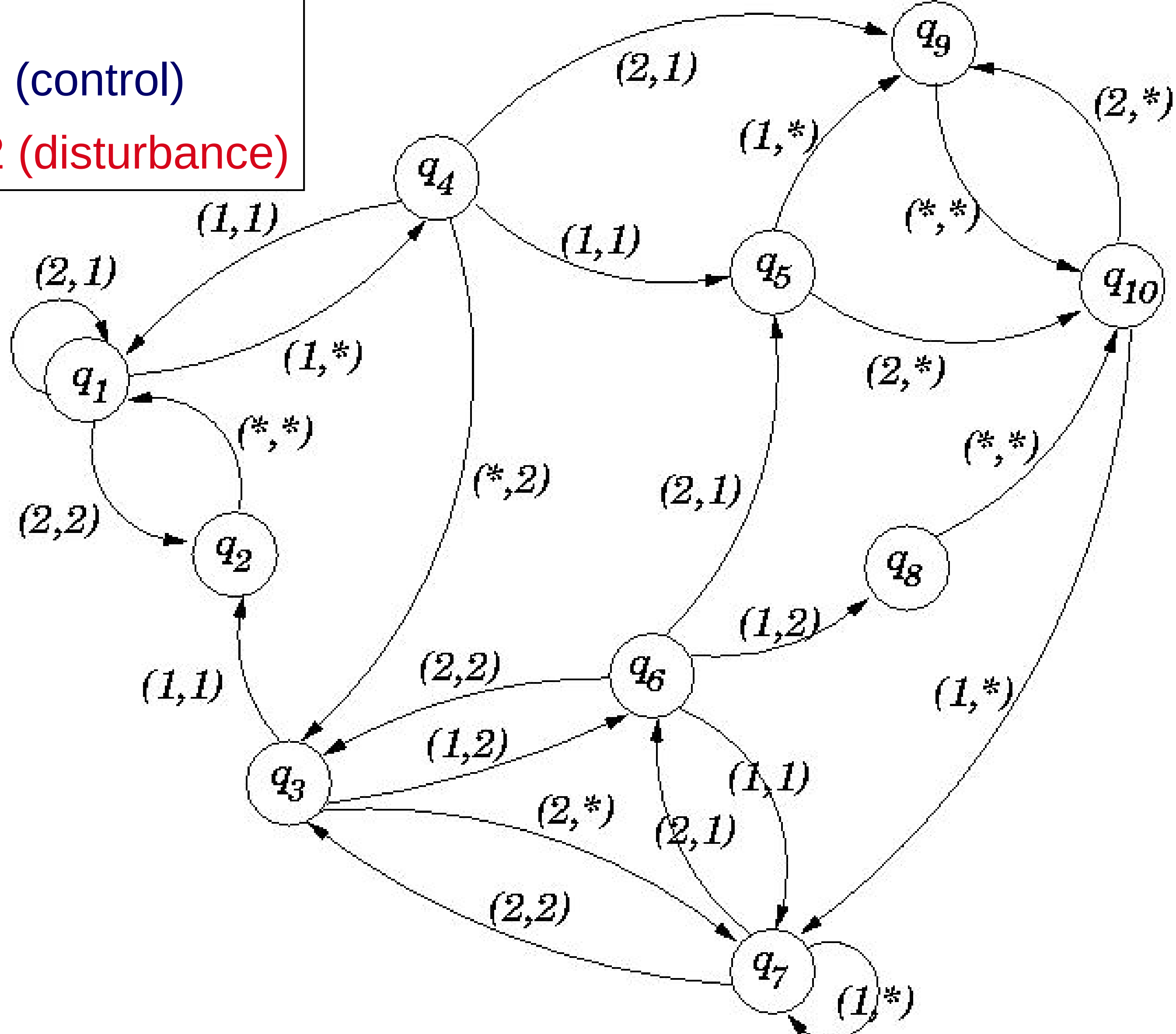
Reachability as game: disturbance attempts to force system into unsafe region, control attempts to stay safe

A discrete game

(i, j)

$i \in \{1, 2\}$ Player 1 (control)

$j \in \{1, 2\}$ Player 2 (disturbance)

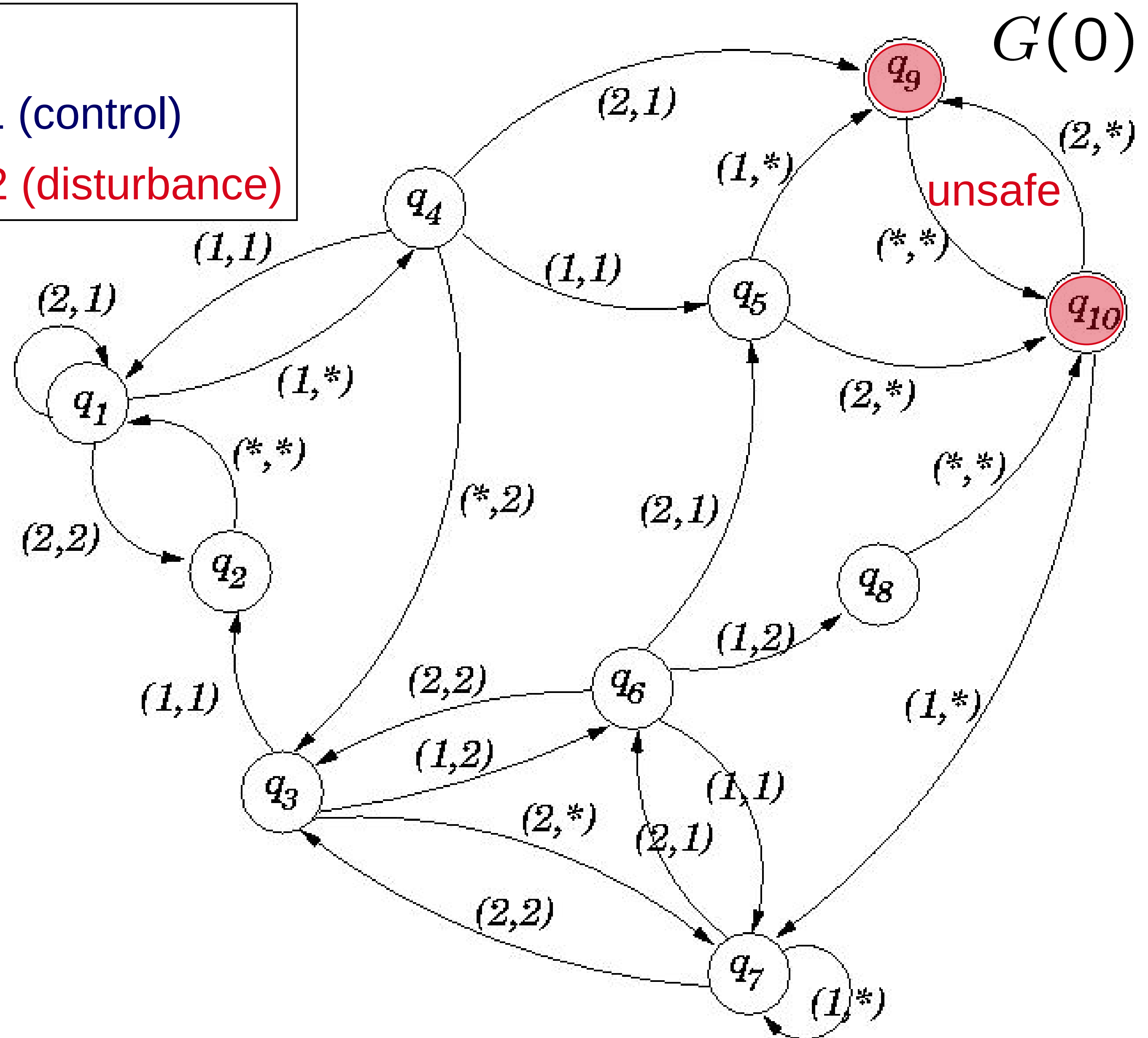


A discrete game

(i, j)

$i \in \{1, 2\}$ Player 1 (control)

$j \in \{1, 2\}$ Player 2 (disturbance)

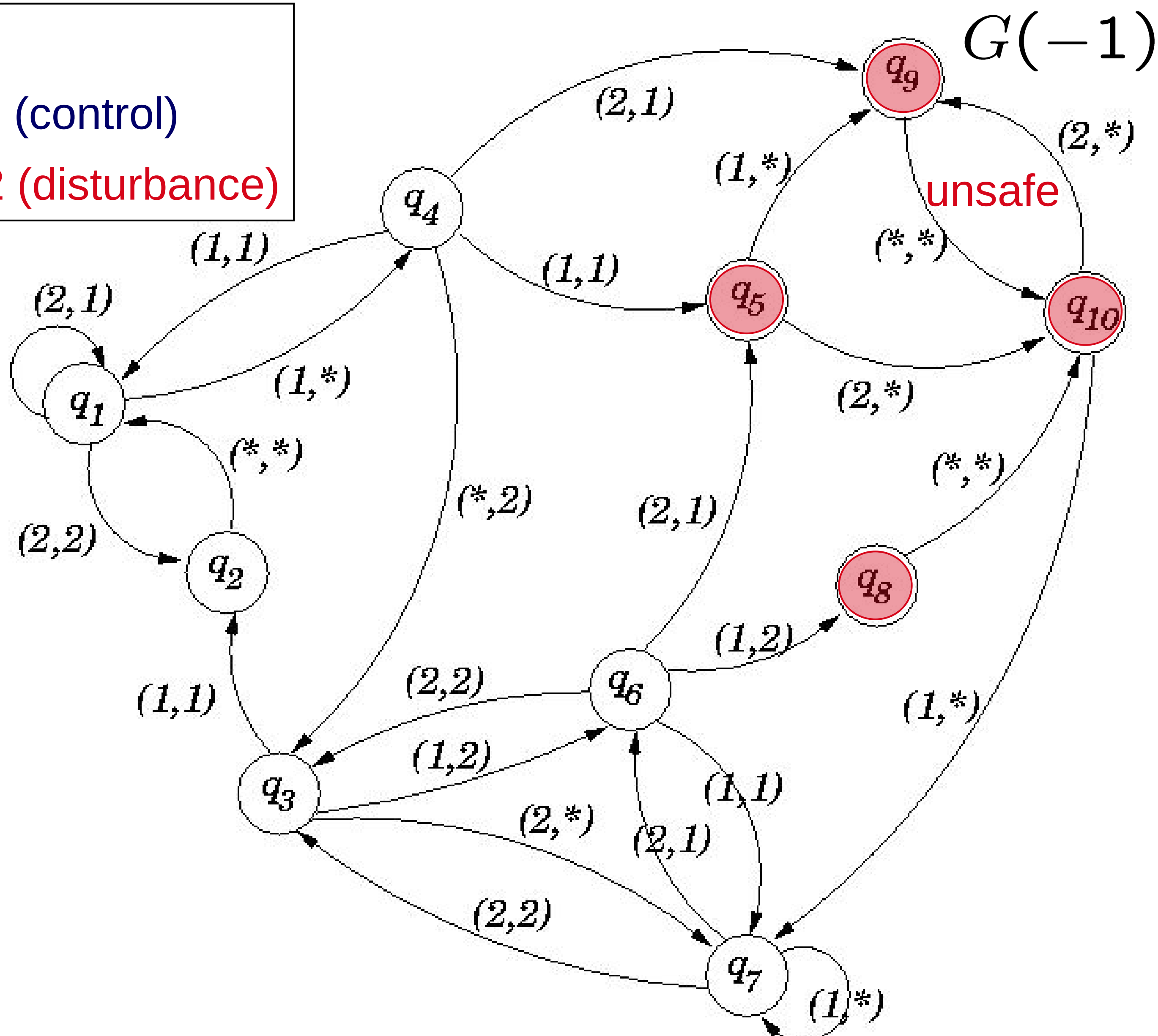


A discrete game

(i, j)

$i \in \{1, 2\}$ Player 1 (control)

$j \in \{1, 2\}$ Player 2 (disturbance)

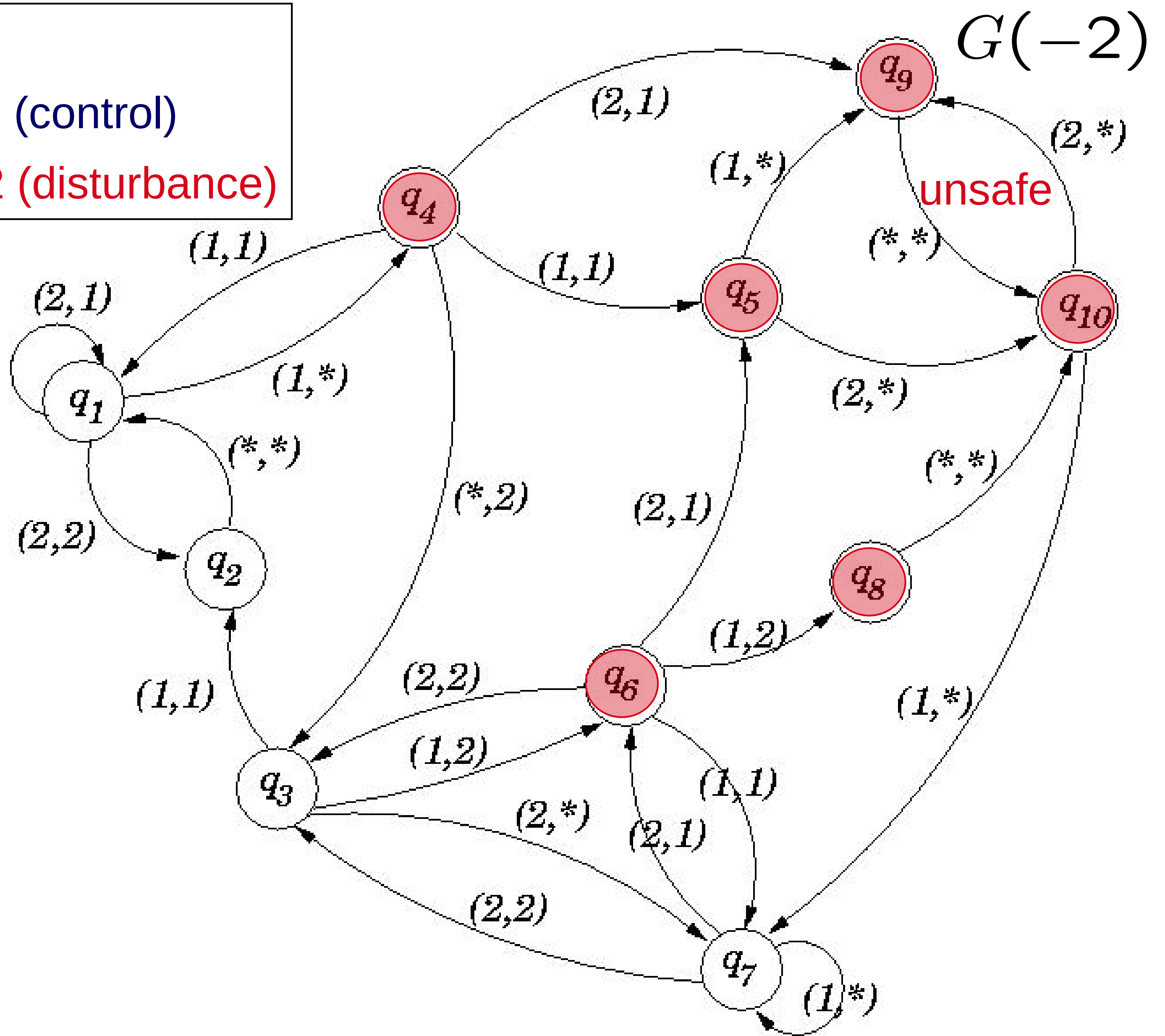


A discrete game

(i, j)

$i \in \{1, 2\}$ Player 1 (control)

$j \in \{1, 2\}$ Player 2 (disturbance)



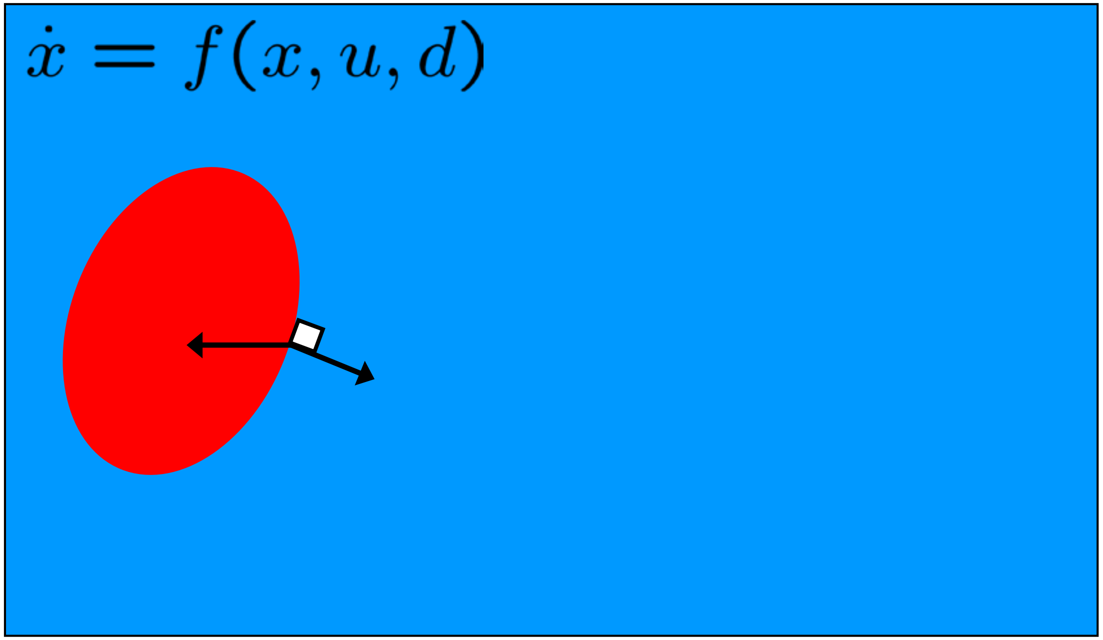
Reachable Set Propagation

Theorem [Computing $G(t)$]:

$$G(t) = \{x : J(x,t) < 0\}$$

where $J(x,t)$ is the unique Crandall-Evans-Lions viscosity solution to:

$$-\frac{\partial J(x,t)}{\partial t} = \min\{0, \max_u \min_d \frac{\partial J(x,t)}{\partial x} f(x,u,d)\}$$



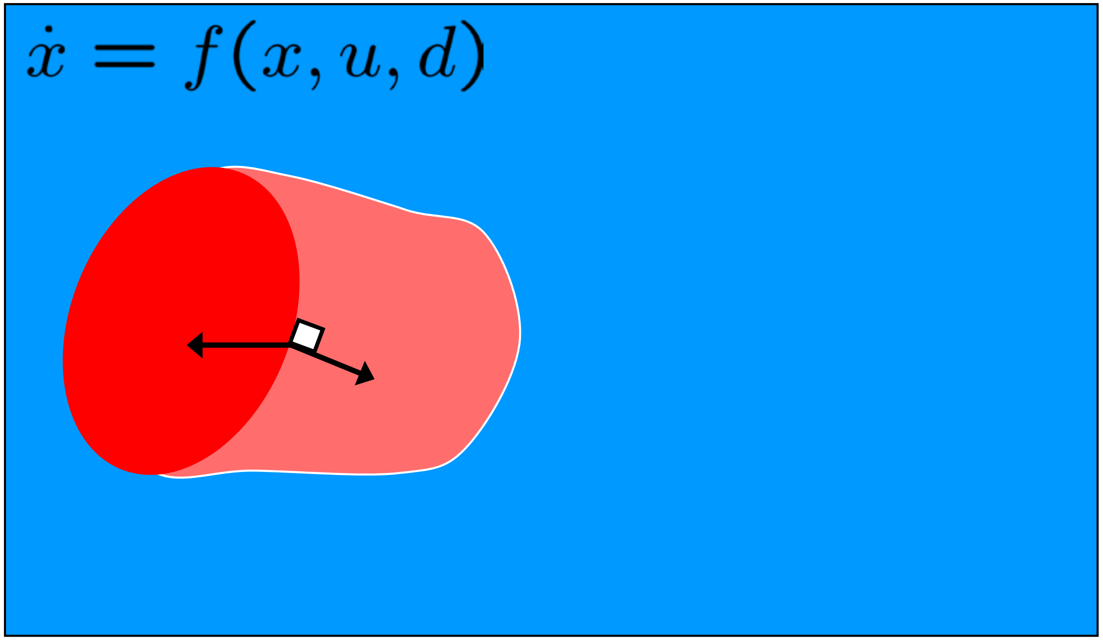
Reachable Set Propagation

Theorem [Computing $G(t)$]:

$$G(t) = \{x : J(x,t) < 0\}$$

where $J(x,t)$ is the unique Crandall-Evans-Lions viscosity solution to:

$$-\frac{\partial J(x,t)}{\partial t} = \min\{0, \max_u \min_d \frac{\partial J(x,t)}{\partial x} f(x,u,d)\}$$



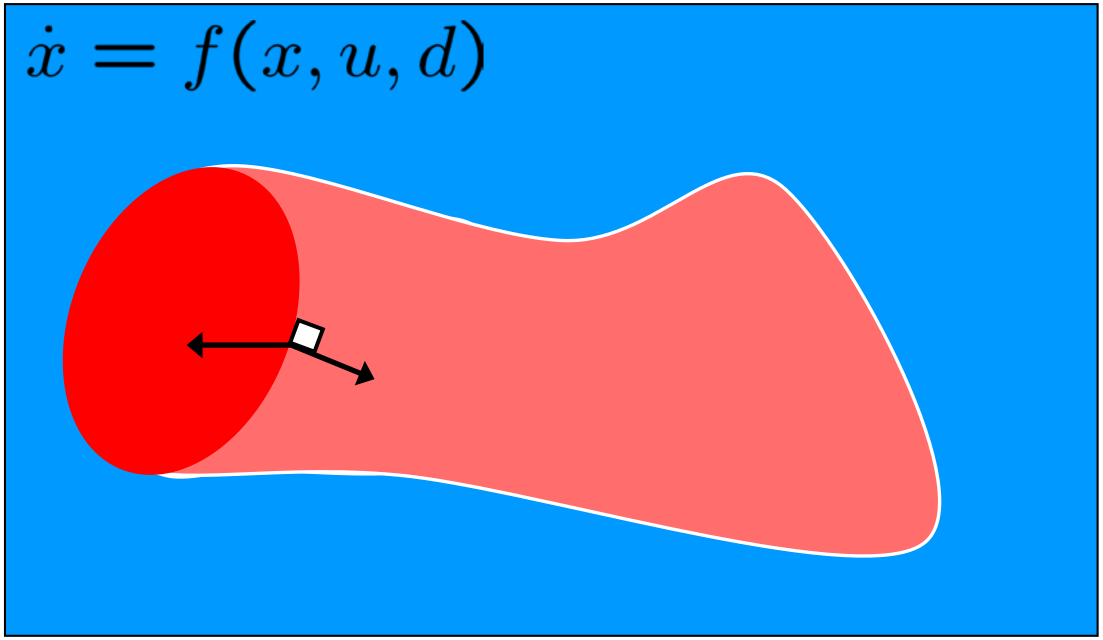
Reachable Set Propagation

Theorem [Computing $G(t)$]:

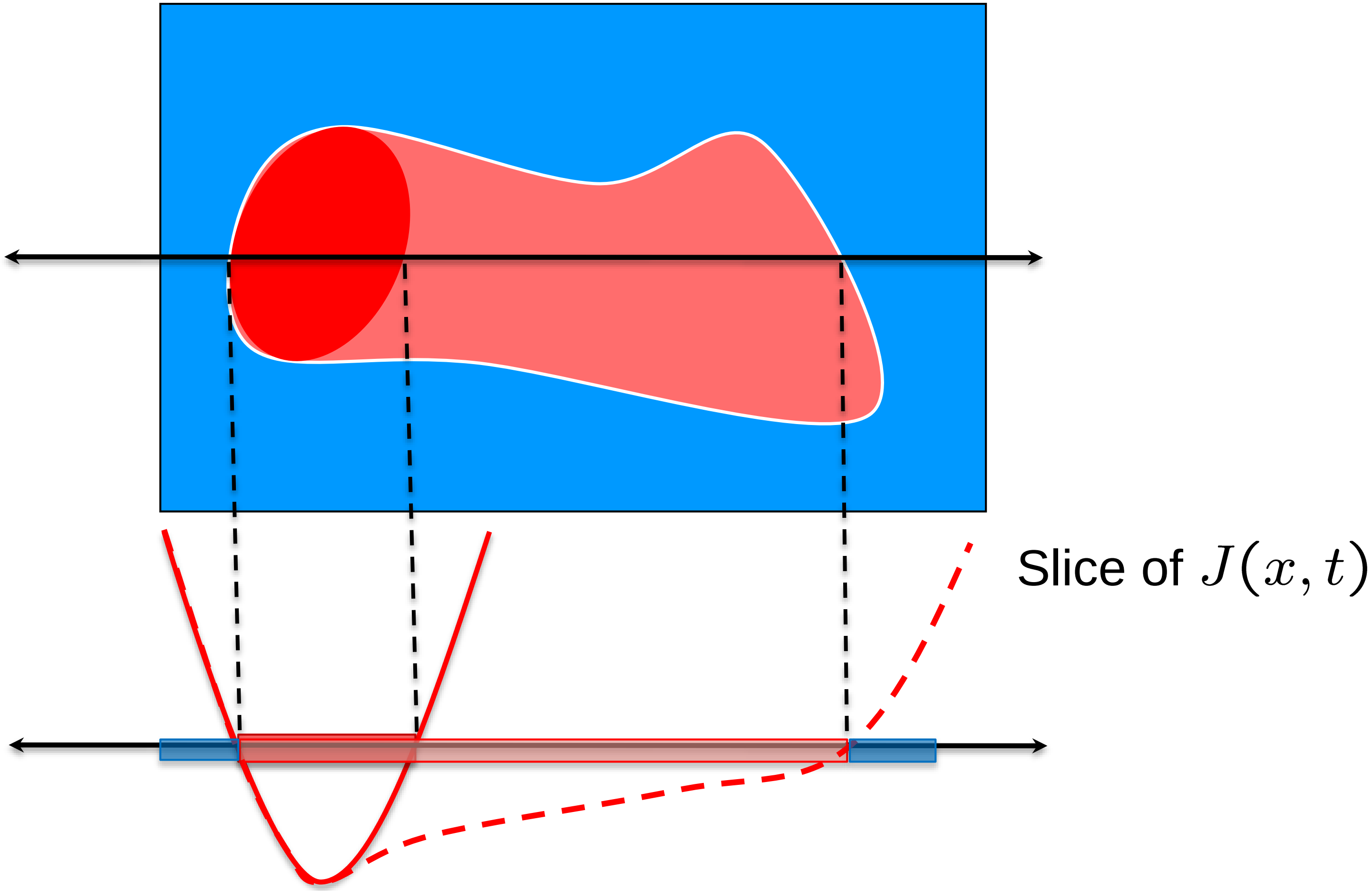
$$G(t) = \{x : J(x,t) < 0\}$$

where $J(x,t)$ is the unique Crandall-Evans-Lions viscosity solution to:

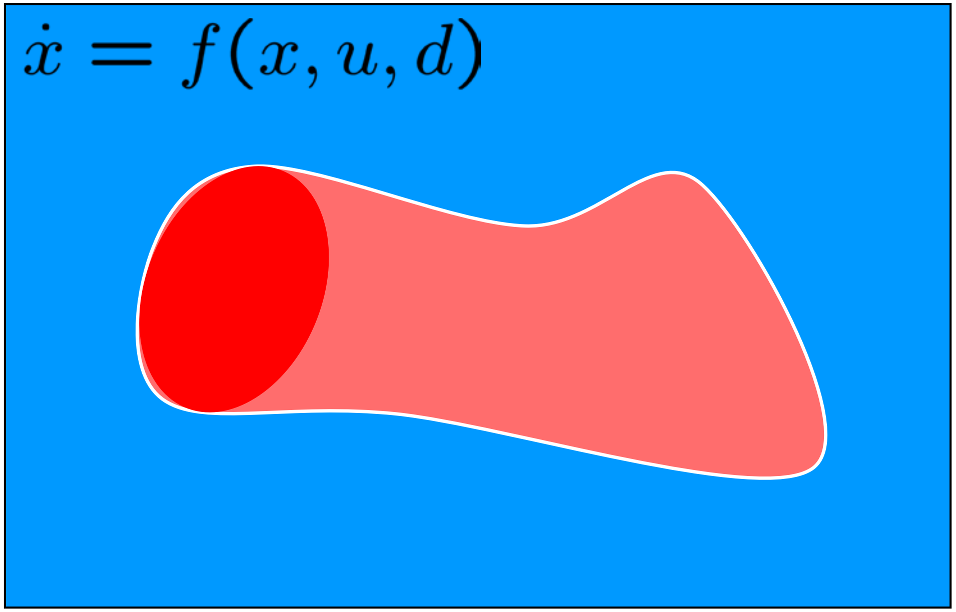
$$-\frac{\partial J(x,t)}{\partial t} = \min\{0, \max_u \min_d \frac{\partial J(x,t)}{\partial x} f(x,u,d)\}$$



Level set interpretation



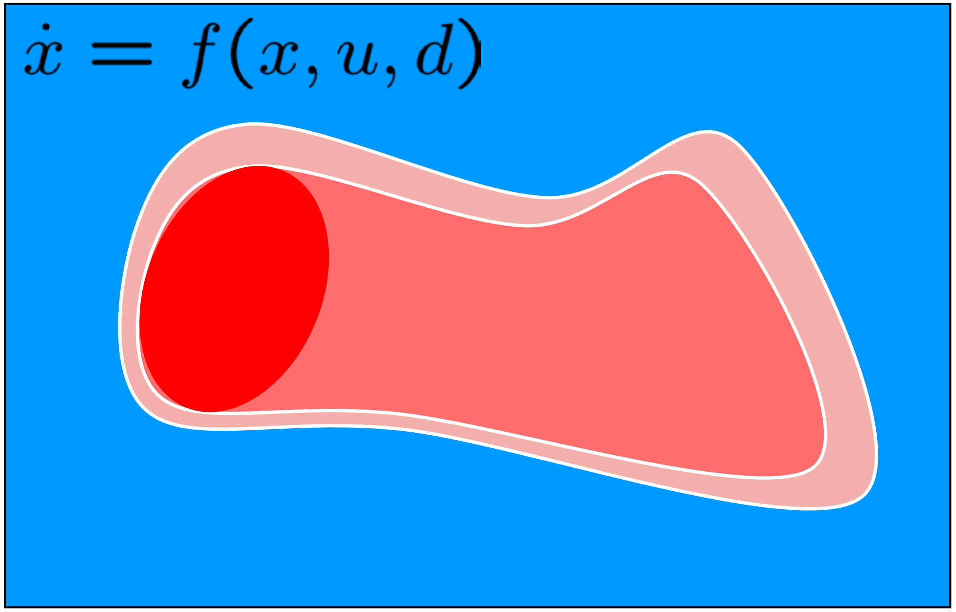
Level set interpretation



Invariance:

- If the set is controlled invariant for all t, any super zero level set is also invariant, and may be used for safety

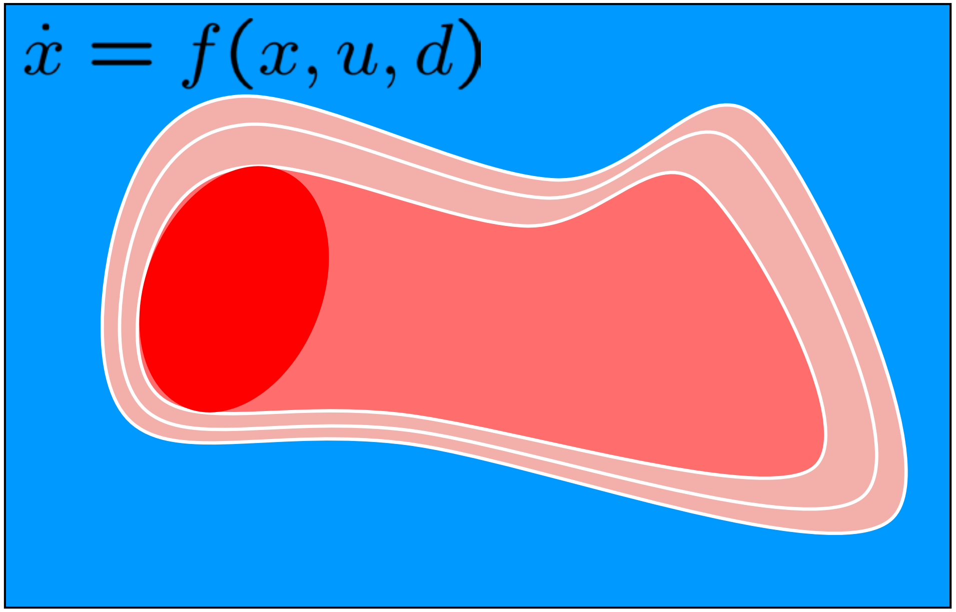
Level set interpretation



Invariance:

- If the set is controlled invariant for all t, any super zero level set is also invariant, and may be used for safety

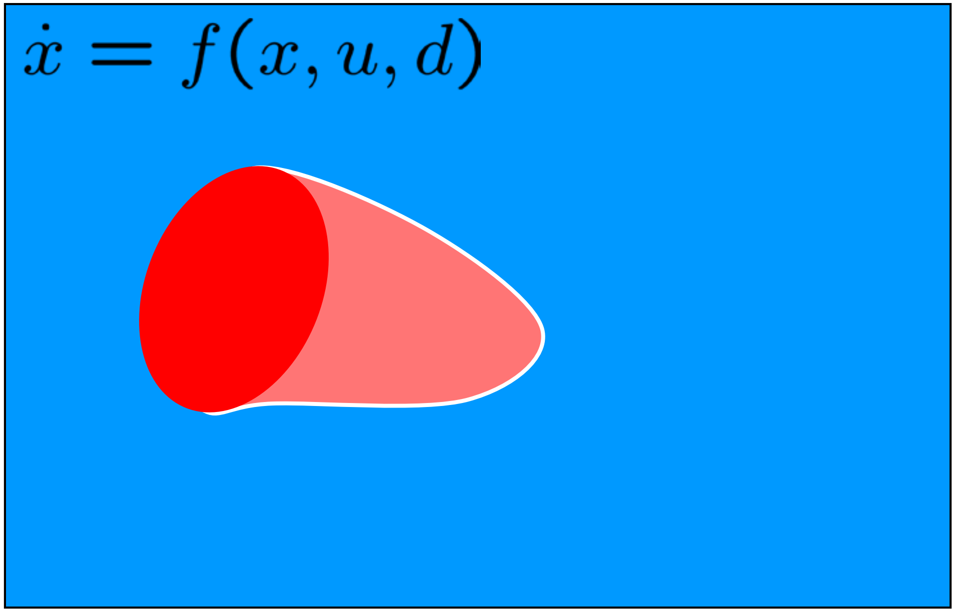
Level set interpretation



Invariance:

- If the set is controlled invariant for all t, any super zero level set is also invariant, and may be used for safety

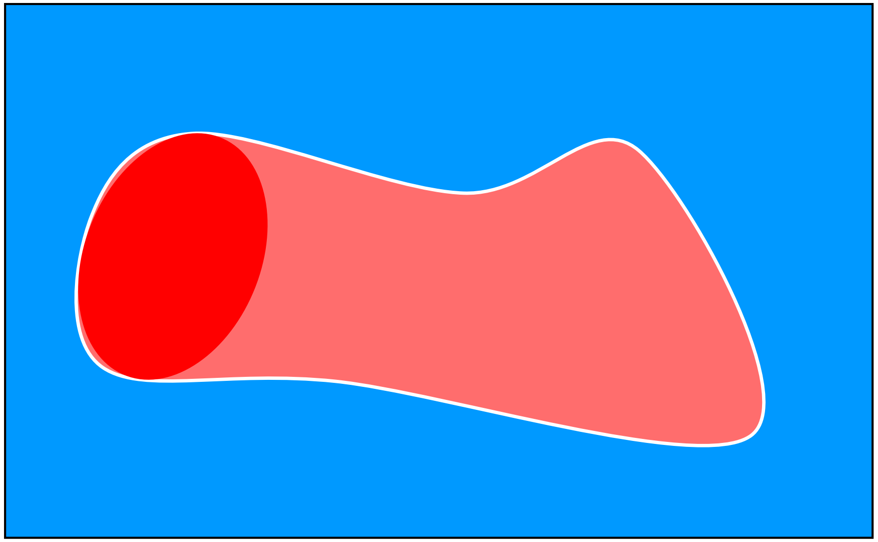
Level set interpretation



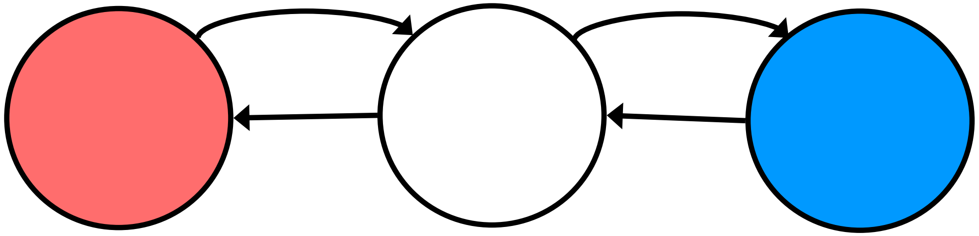
Disturbance:

- And, if the disturbance d were known, the *optimal* reachable set could be computed

Partitions the state space



In red, system may become unsafe

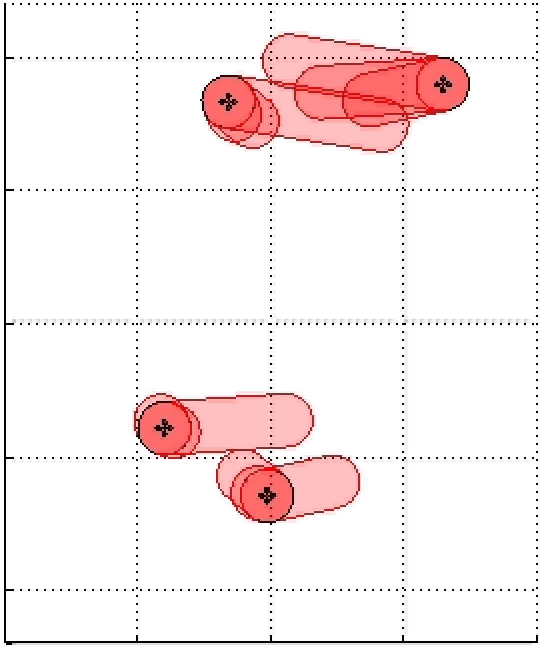


In blue, system will stay safe

On boundary, apply control to stay out of red

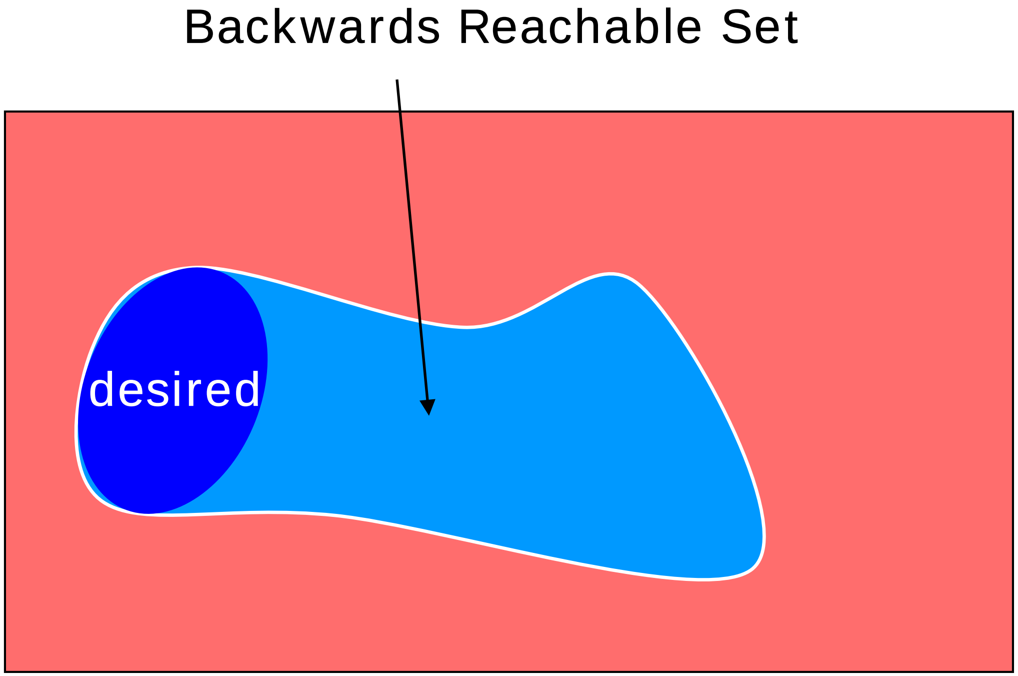
Example 1: Collision Avoidance

Pilots instructed to attempt to collide vehicles



[STARMAC: Stanford Testbed of Autonomous Rotorcraft for MultiAgent Control]

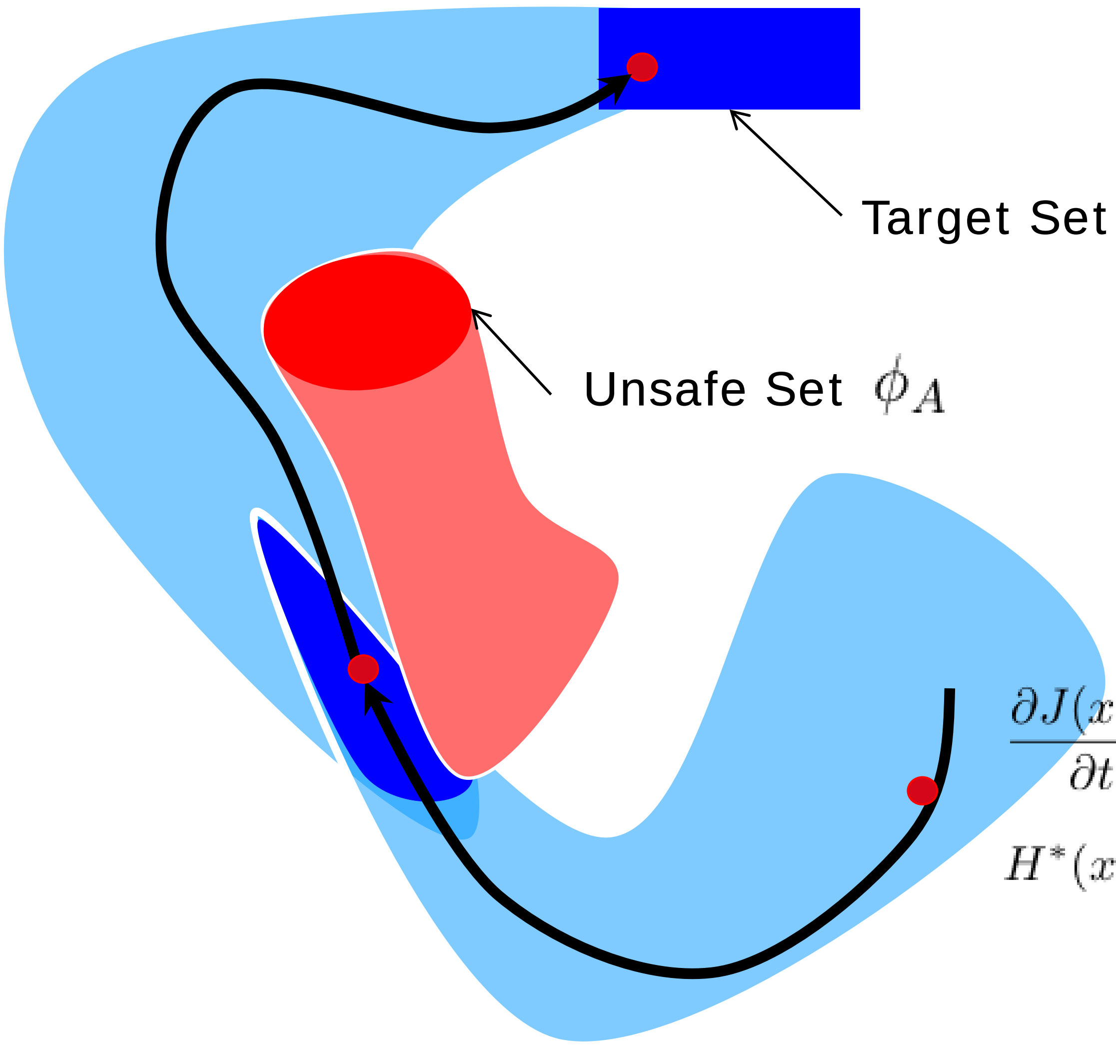
Backwards Reachable Set: Capture



Capture property can also be encoded as a condition on the system's reachable set of states

$$-\frac{\partial J(x,t)}{\partial t} = \min\{0, \min_u \max_d \frac{\partial J(x,t)}{\partial x} f(x,u,d)\}$$

Mode sequencing and reach-avoid



String together capture sets, starting from the target set and working backwards

Avoid sets can be combined with capture sets to guarantee safety

$$\frac{\partial J(x,t)}{\partial t} + \min[0, H^*(x, \frac{\partial J(x,t)}{\partial x})] = 0$$

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

$$\min_u \max_d \frac{\partial J(x,t)^T}{\partial x} \cdot f(x,u,d,t)$$

Subject to $J(x,t) \geq -\phi_A$

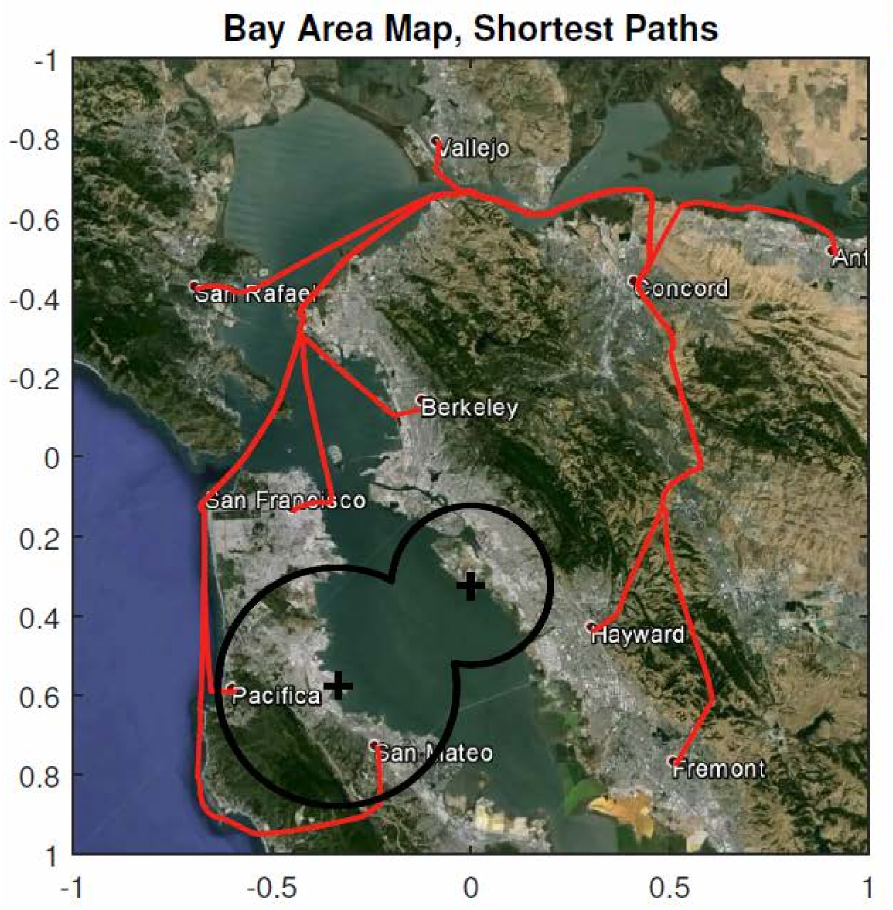
Dealing with the curse of dimensionality

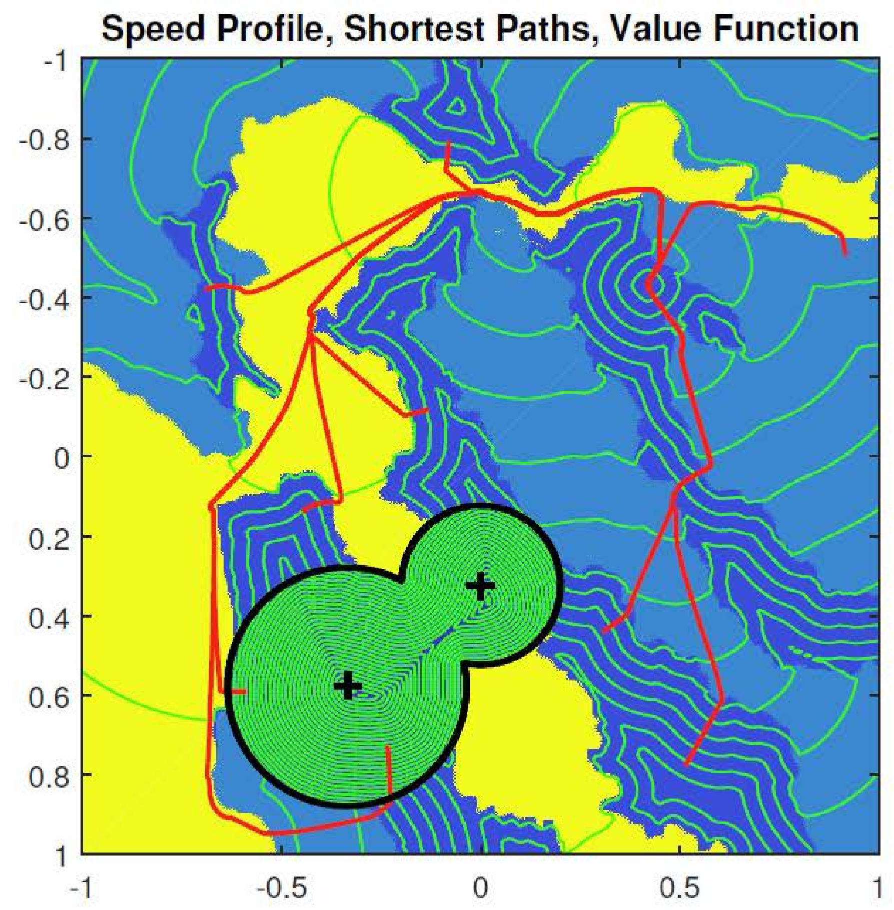
- **Convergent approximations**
  - Aubin, Saint-Pierre…
- **Decompositions**
  - Mitchell, Del Vecchio, Chen, Grizzle, Ames, Tabuada…
- **Approximate bisimulations**
  - Girard, Pappas, Tabuada…
- **Piecewise and multi-affine systems**
  - Morari, Bemporad, Borrelli, Krogh, Johansson, Rantzer, Belta, Kaynama, Oishi…
- **Ellipsoidal and polyhedral sets**
  - Kurzhanski, Kurzhanski, Varaiya, Girard, Frehse, Sankaranarayanan, Stipanovic…
- **Barrier certificates**
  - Papachistodoulou, Julius, Parrilo, Lall, Topcu…
- **Monotone systems**
  - Sontag, Del Vecchio, Arcak, Coogan
- **LTL specifications**
  - Kress-Gazit, Raman, Murray, Wongpiromsarn, Belta…

Outline

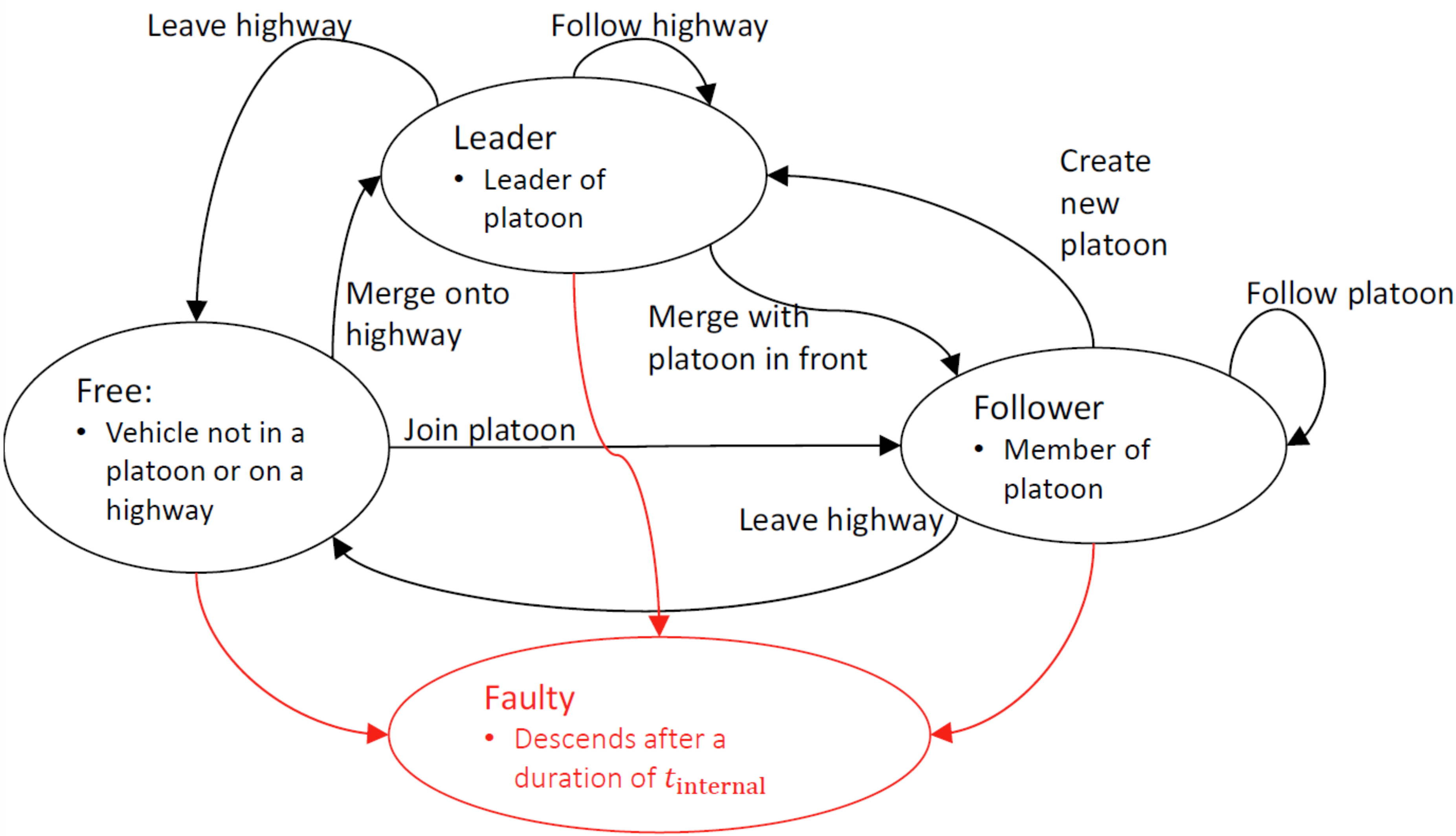
- Reachable sets for hybrid systems
  - Overview
  - Examples:
    - Collision avoidance
    - Mode sequencing
- Learning dynamic behavior safely
  - Overview
  - Examples:
    - Learning to fly

Example 2: Platooning UAVs

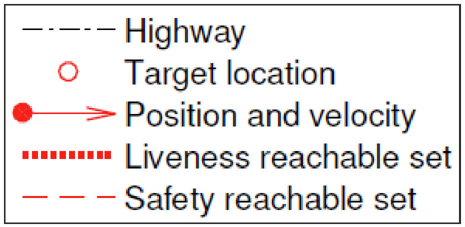


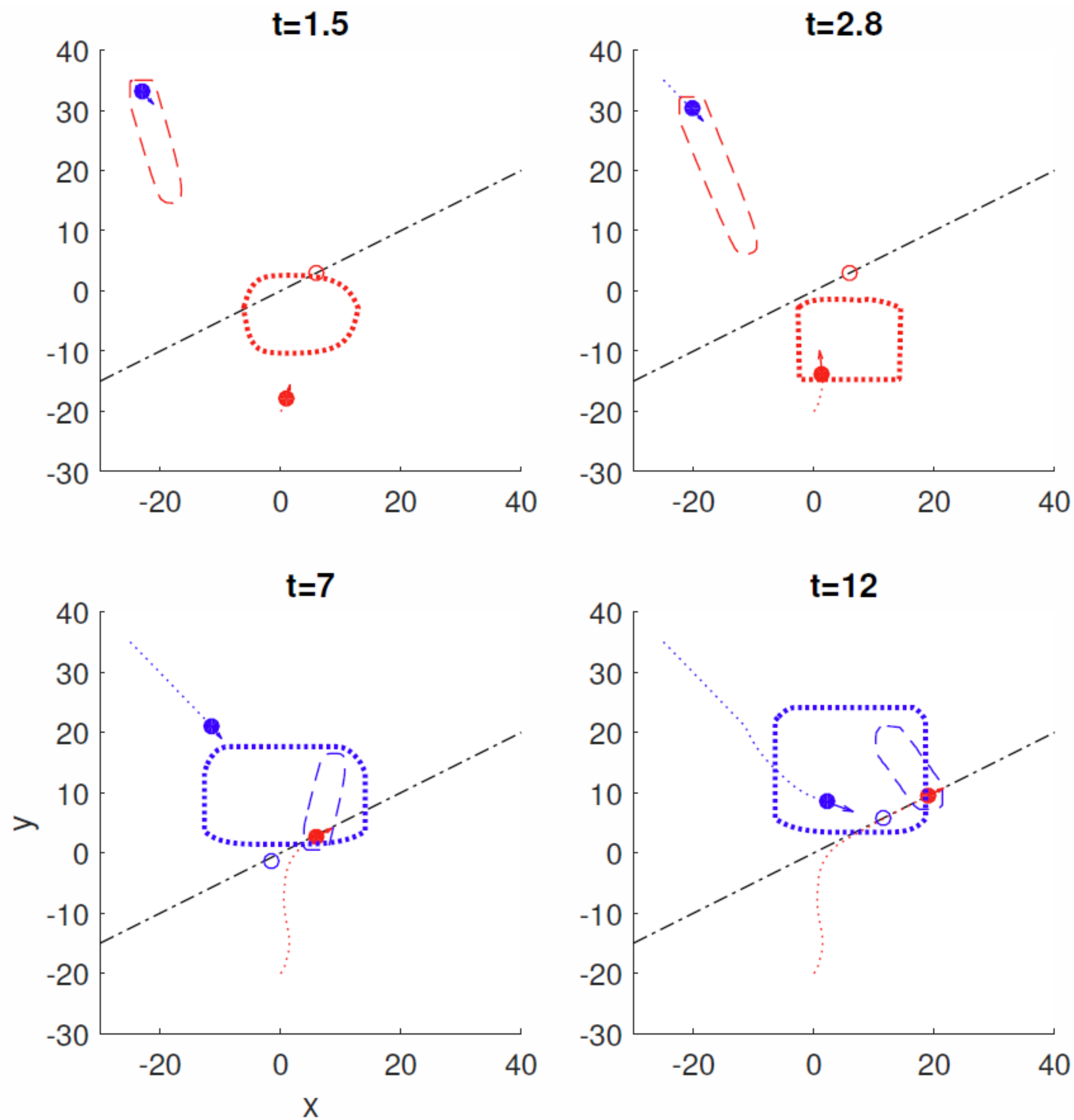


Example 2: Platooning UAVs



Merging onto highway and joining platoon

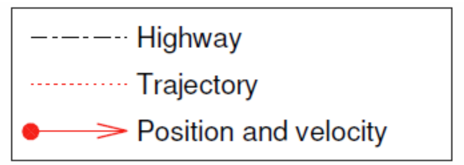




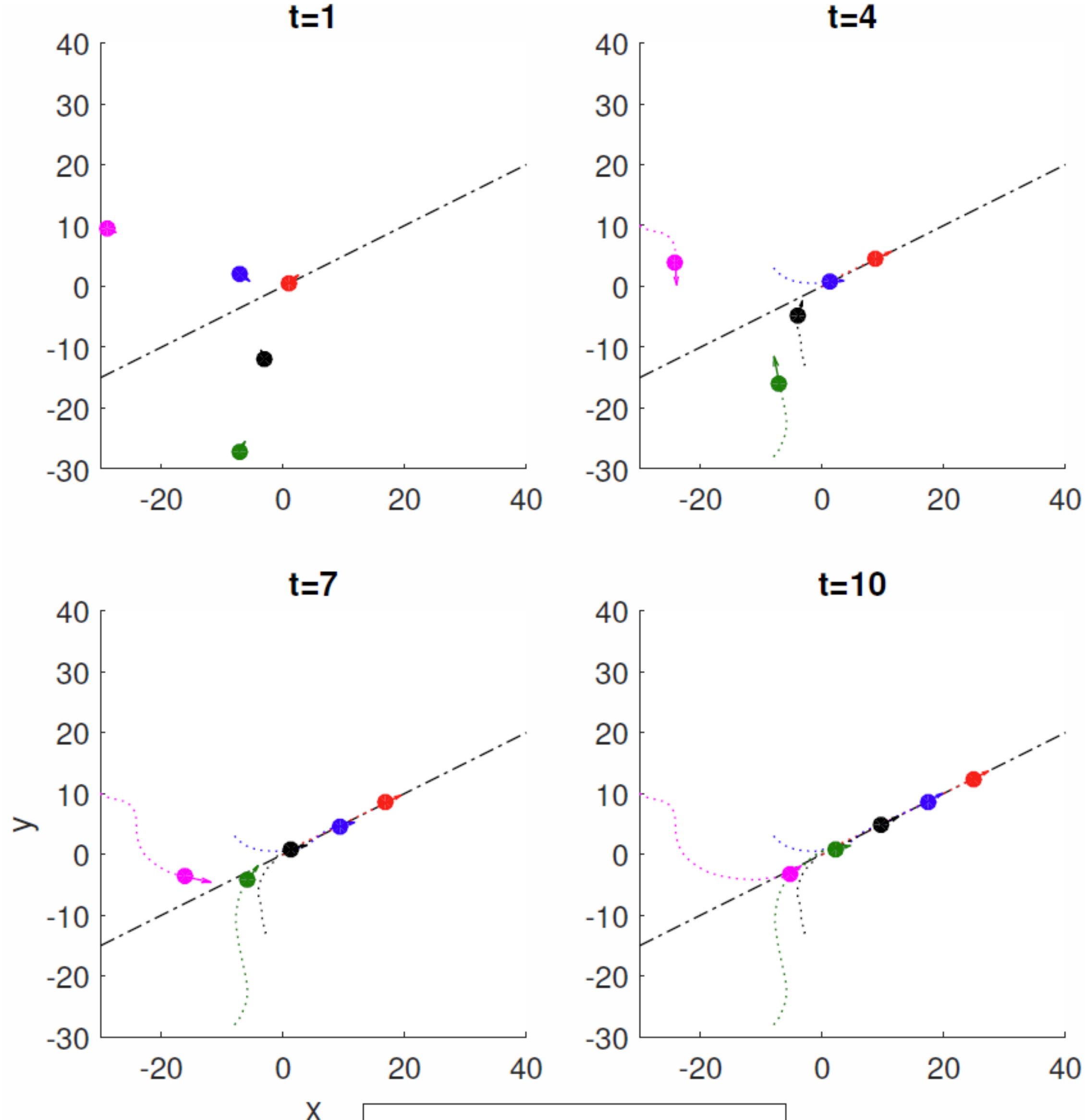
Red vehicle merges onto highway

Blue vehicle joins red vehicle’s platoon

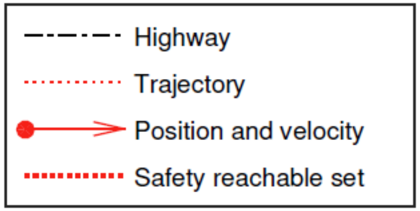
Merging onto highway and joining platoon

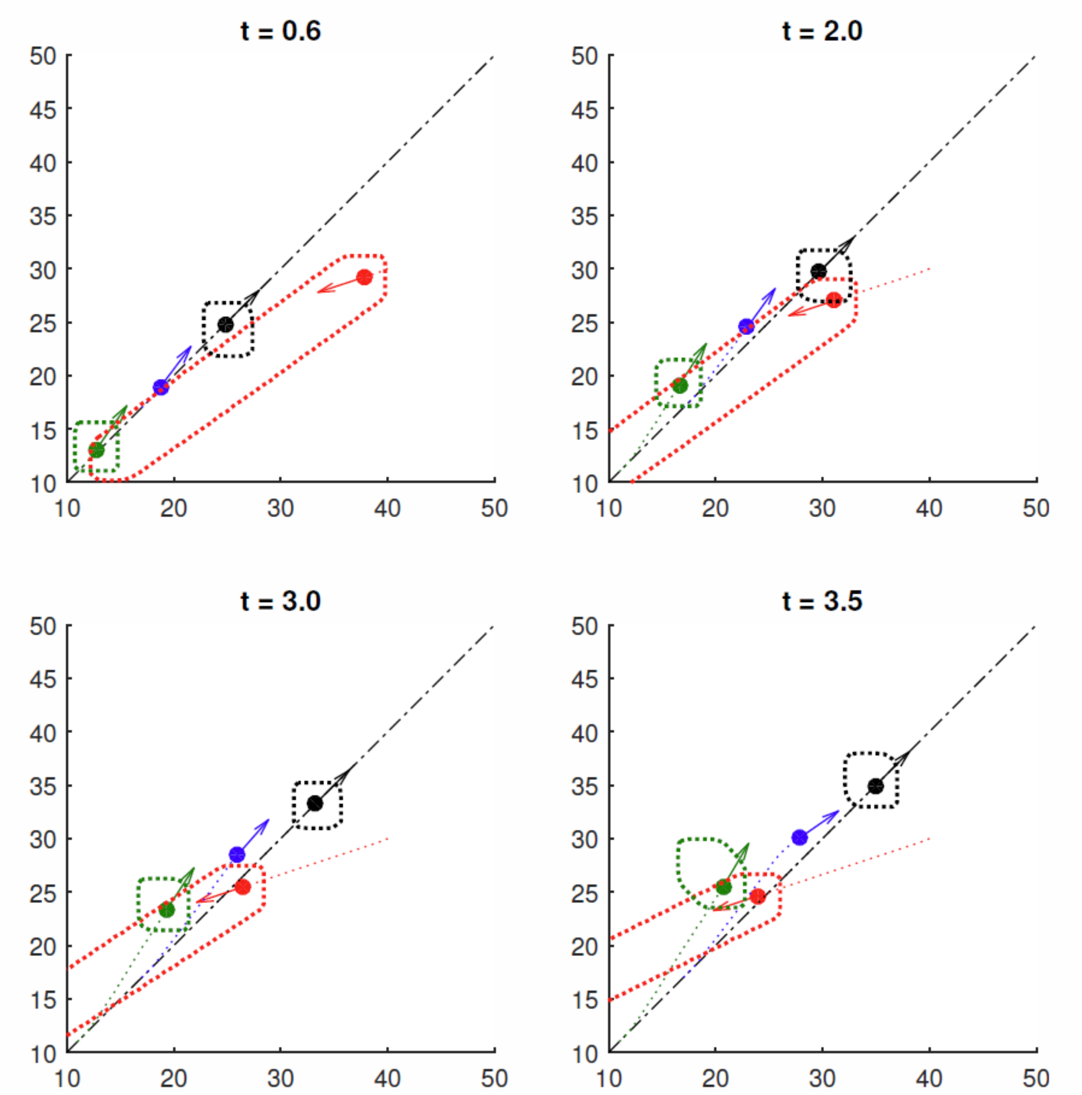


4 vehicles join platoon following red vehicle



Intruder vehicle





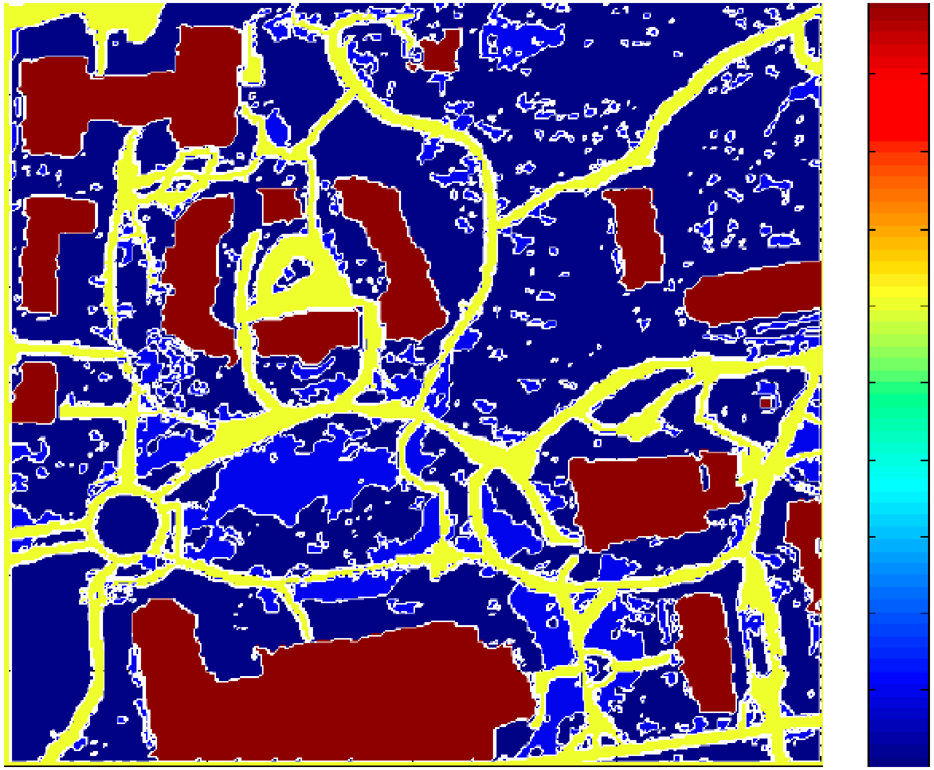
Platoon responding to intruder (red vehicle)

Reachable sets for blue vehicle are shown

Blue vehicle must stay outside of all dotted boundaries

Example 3: Forced Landing System

Courtesy of Google Maps - ©2014 Google



Outline

- Reachable sets for hybrid systems
  - Overview
  - Examples:
    - Collision avoidance
    - Mode sequencing
- Learning dynamic behavior safely
  - Overview
  - Examples:
    - Learning to fly

Safety
Simplicity

Ability to adapt to new information

Learn models from data…

… but stay safe while learning

- **Safety:**
  - A nominal model with error bounds
  - Reachable sets computed to ensure safety in worst case
- **Performance:**
  - Use online learning to update model
  - Cost function used to generate control action within the safe set

[Chisci,Tedrake, Abbeel, Aswani, Zeilinger]

Example 4: Safe - Policy Gradient Reinforcement Learning

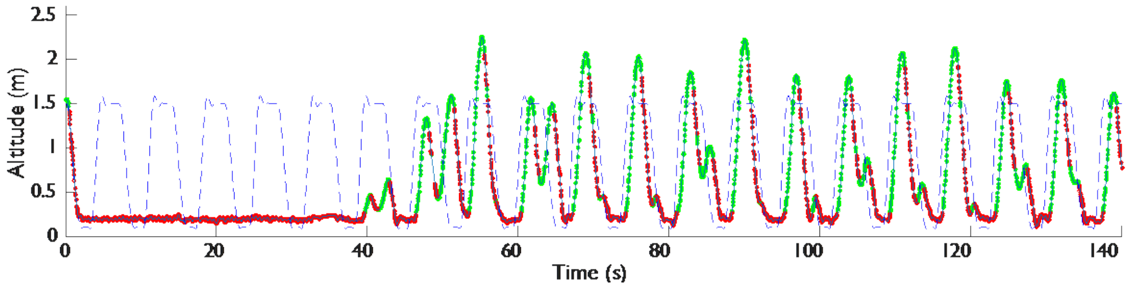
Learn to fly from scratch?

[PGSD: Kolter and Ng, 2009]

Example 4: Safe - Policy Gradient Reinforcement Learning

The quadrotor first:

Learn to fly from scratch?



After about 1 minute,
it can roughly track the trajectory

Soon, it starts experimenting

drops

...but the safe controller steps in

[PGSD: Kolter and Ng, 2009]

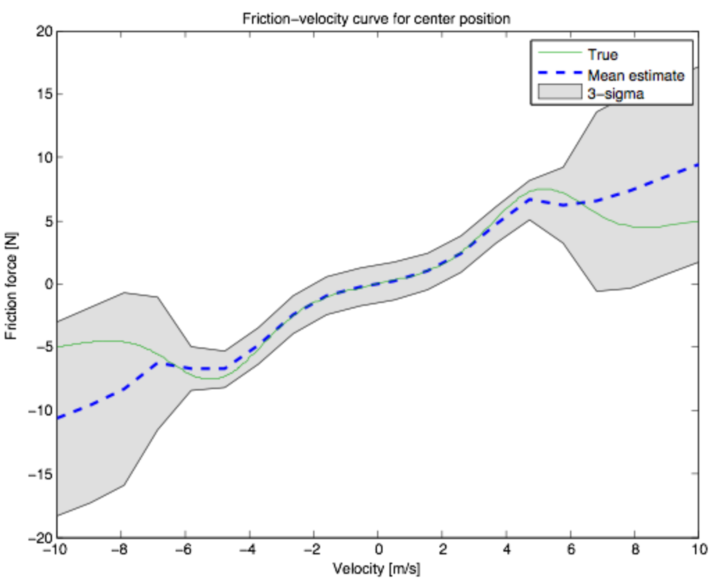
Gaussian Processes (GP)

(a Gaussian distribution over functions)

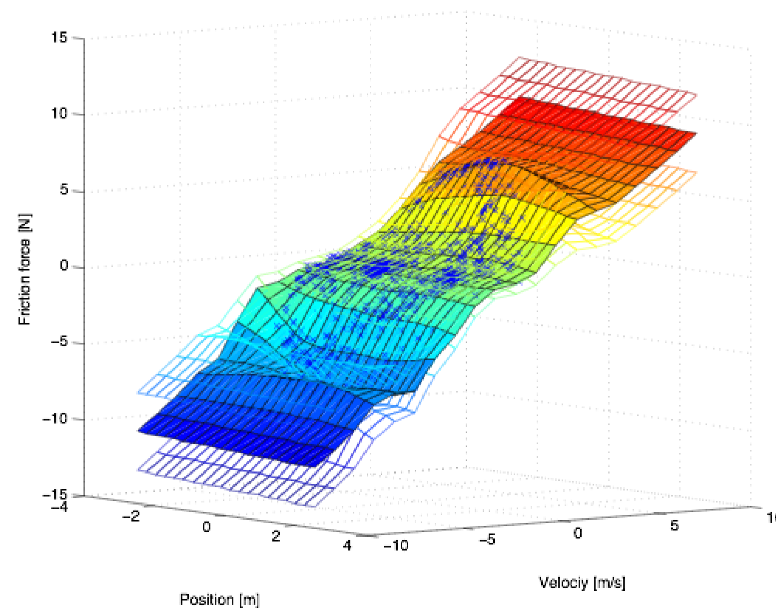
visited states $\{x_i\}_{i=1}^N$ → **GP Regression** → mean function $\bar{d}(x)$

measured disturbance $\{\hat{d}_i\}_{i=1}^N$ → **GP Regression** → variance $\sigma^2(x)$

$$\mathcal{D}(x) = [\bar{d}(x) - m\sigma(x), \bar{d}(x) + m\sigma(x)]$$

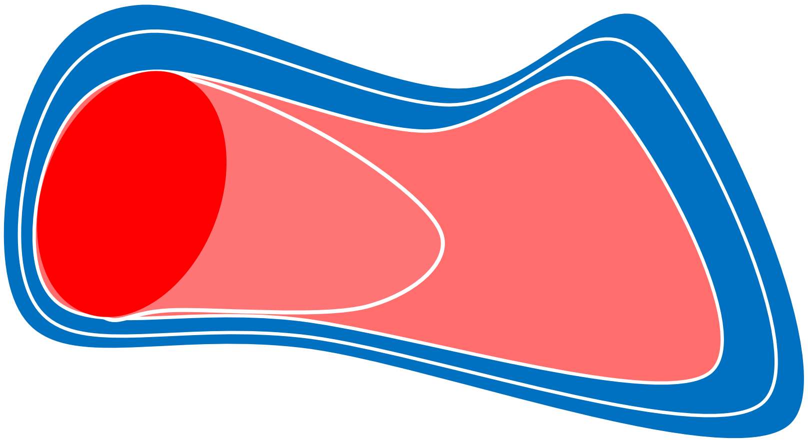


Data-driven bounds



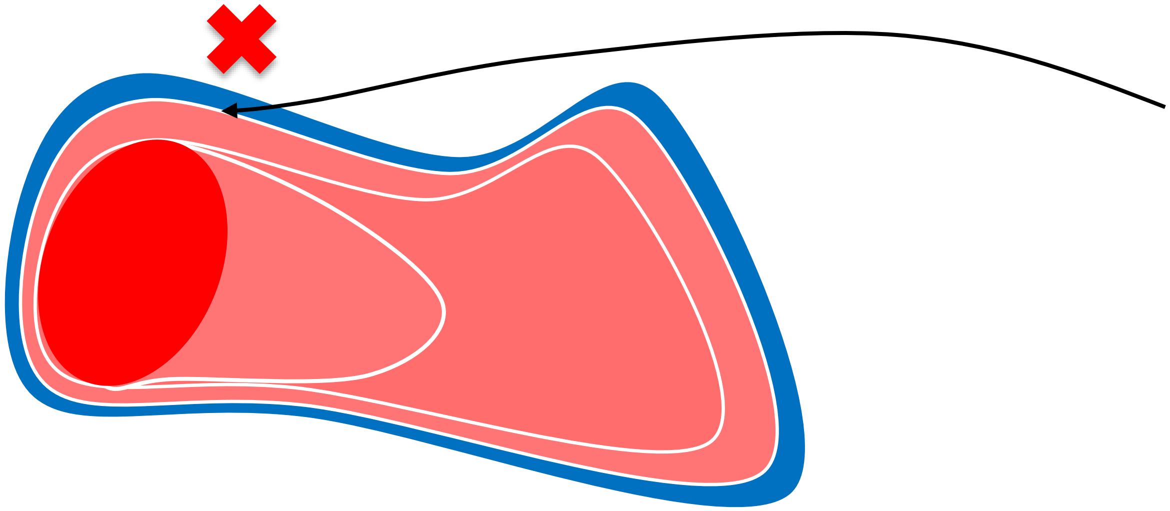
[GP: Rasmussen and Williams, 2006]

Online Disturbance Model Validation



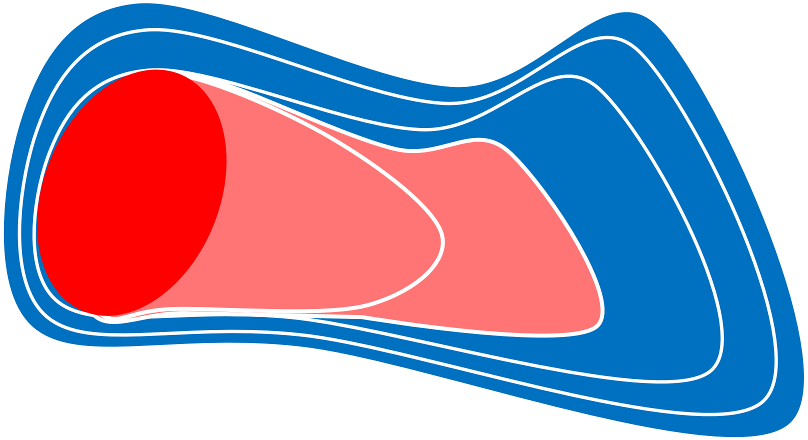
- Initialize active unsafe set = smallest candidate set

Online Disturbance Model Validation



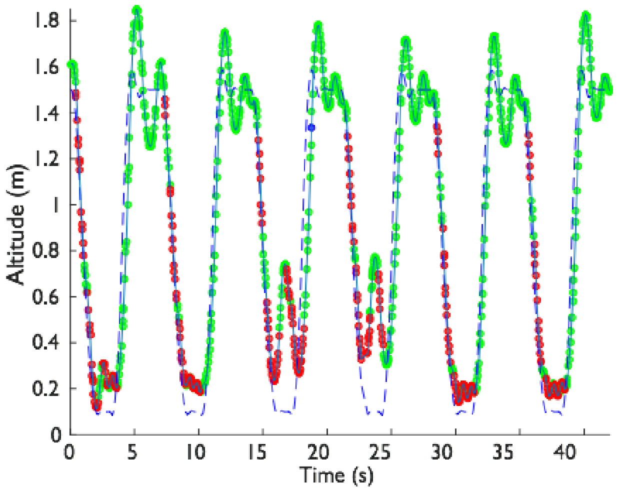
- Initialize active unsafe set = smallest candidate set
- Repeat:
  - Measure disturbance
  - Validate measured disturbance at visited states against model
  - If model inaccuracy is detected, expand unsafe set
  - Update disturbance model

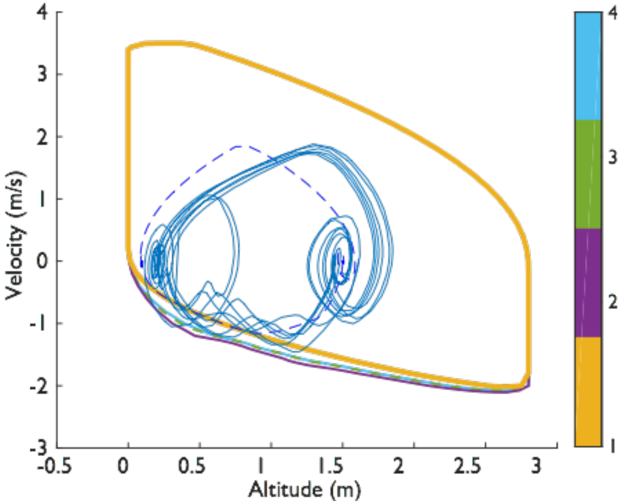
Online Disturbance Model Validation



- Initialize active unsafe set = smallest candidate set
- Repeat:
  - Measure disturbance
  - Validate measured disturbance at visited states against model
  - If model inaccuracy is detected, expand unsafe set
  - Update disturbance model

Example 5: Safe Learning

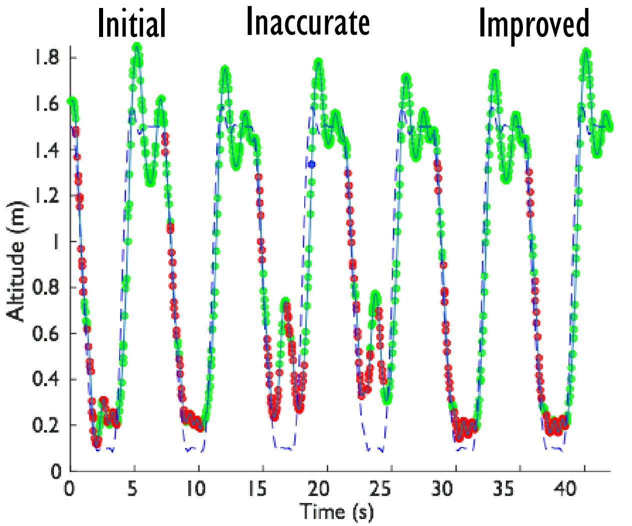




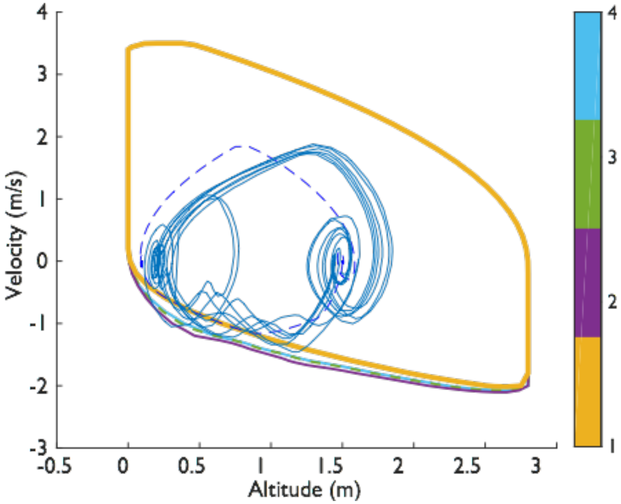
[Akametalu, Fisac, Zeilinger]

Example 5: Safe Learning

First computed model is locally inaccurate



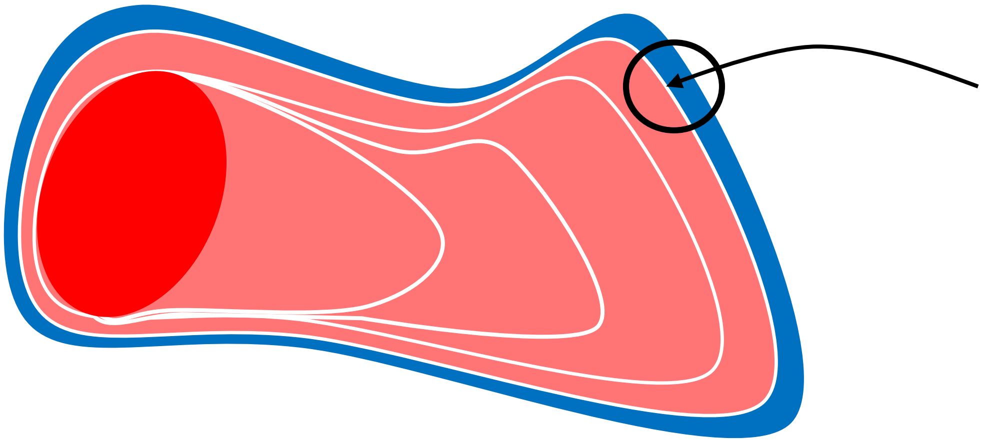
System detects inconsistency,
slightly contracts safe set



Tracking resumes after a better model is computed

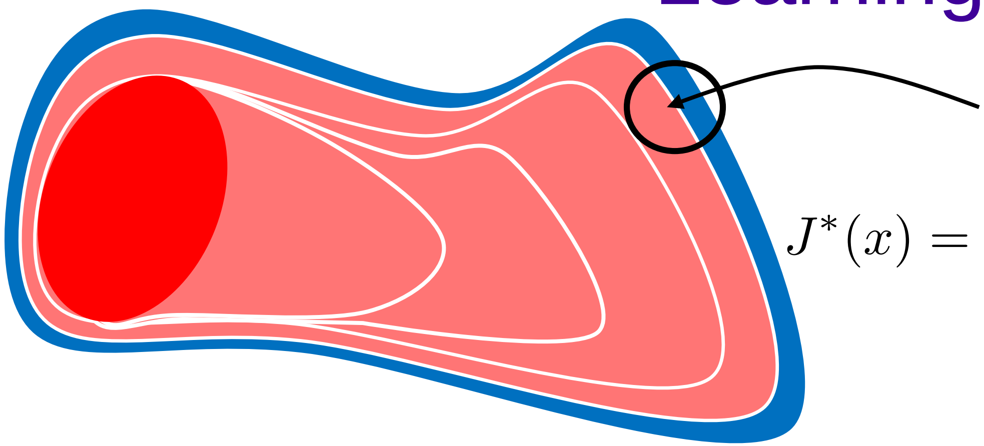
[Akametalu, Fisac, Zeilinger]

Local Updates



- Instead of learning disturbance function globally:
  - Measure value function locally
  - Update value function locally

Local Updates using Temporal Difference Learning



$$J^*(x) = \min\{J^*(f_{\Delta t}(x, u^*(x), d(x)), l(x)\}$$

Conservative Initialization

$$\frac{\partial J^0(x,t)}{\partial t} = -\min\left\{0, \max_{u\in\mathcal{U}} \min_{d\in\mathcal{D}} \frac{\partial J^0(x,t)}{\partial x}^T f(x,u,d)\right\}$$

Current Least Restrictive Control Law

$$u \in \begin{cases} \mathcal{U}, & \text{if } J^k(x_k) > 0 \\ u^*(x_k), & \text{otherwise} \end{cases}$$

Find Error

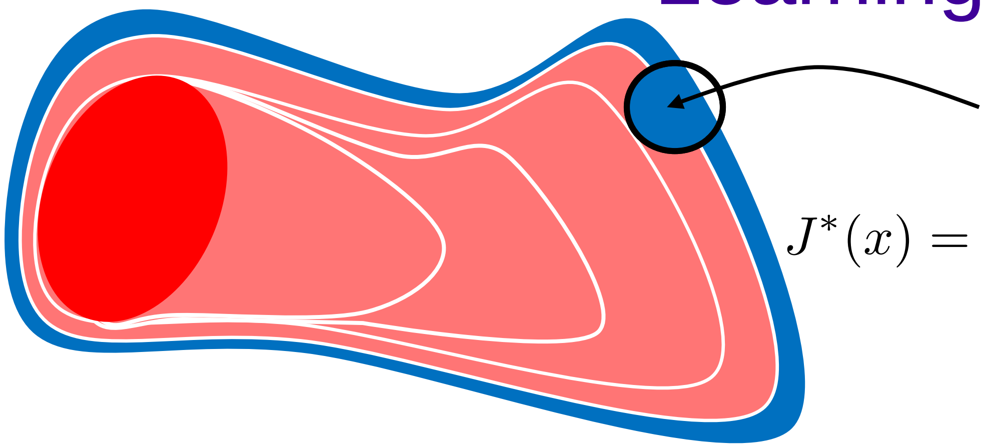
$$e = J^k(x_{k+1}) - J^k(x_k)$$

Update Online

$$J^{k+1}(x_k) \leftarrow \min\{\alpha e + J^k(x_k), l(x_k)\}$$

[Akametalu 2015; TD Learning: Sutton 1988]

Local Updates using Temporal Difference Learning



$$J^*(x) = \min\{J^*(f_{\Delta t}(x, u^*(x), d(x)), l(x)\}$$

Conservative Initialization

$$\frac{\partial J^0(x,t)}{\partial t} = -\min\left\{0, \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \frac{\partial J^0(x,t)}{\partial x}^T f(x,u,d)\right\}$$

Current Least Restrictive Control Law

$$u \in \begin{cases} \mathcal{U}, & \text{if } J^k(x_k) > 0 \\ u^*(x_k), & \text{otherwise} \end{cases}$$

Find Error

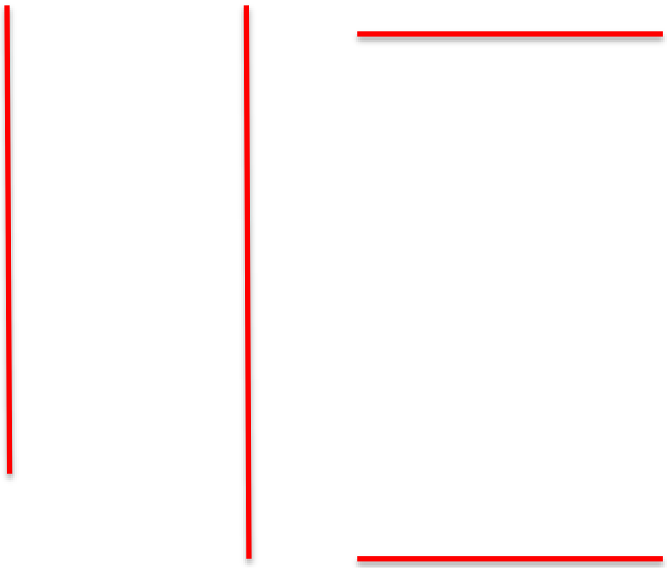
$$e = J^k(x_{k+1}) - J^k(x_k)$$

Update Online

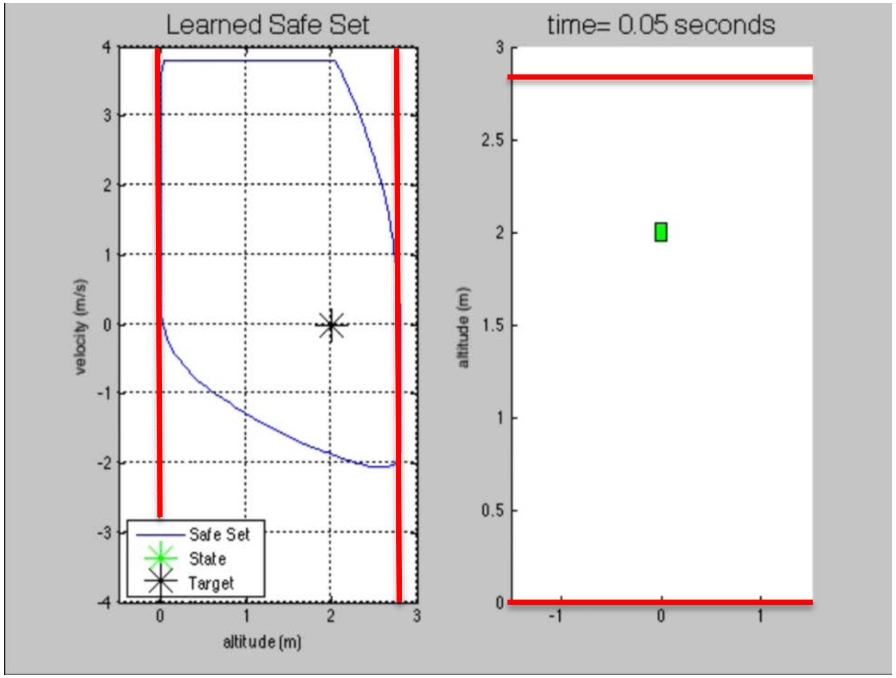
$$J^{k+1}(x_k) \leftarrow \min\{\alpha e + J^k(x_k), l(x_k)\}$$

[Akametalu 2015; TD Learning: Sutton 1988]

Example 6: Learning to Fly
(in a confined space with unknown payload)



Example 6: Learning to Fly
(in a confined space with unknown payload)



Conclusions and current work

- Analysis and control of hybrid systems
  - Safety, from reachability analysis
  - Simplicity, from hybrid system representation
  - UAV safety from reach-avoid games
  - Contrails: ATC game for Android

- Ability to learn from new information
  - Safe learning, Local updates
  - Forced Landing System

Conclusions and current work

- Analysis and control of hybrid systems
  - Safety, from reachability analysis
  - Simplicity, from hybrid system representation
  - UAV safety from reach-avoid games
  - Contrails: ATC game for Android

- Ability to learn from new information
  - Safe learning, Local updates
  - Forced Landing System

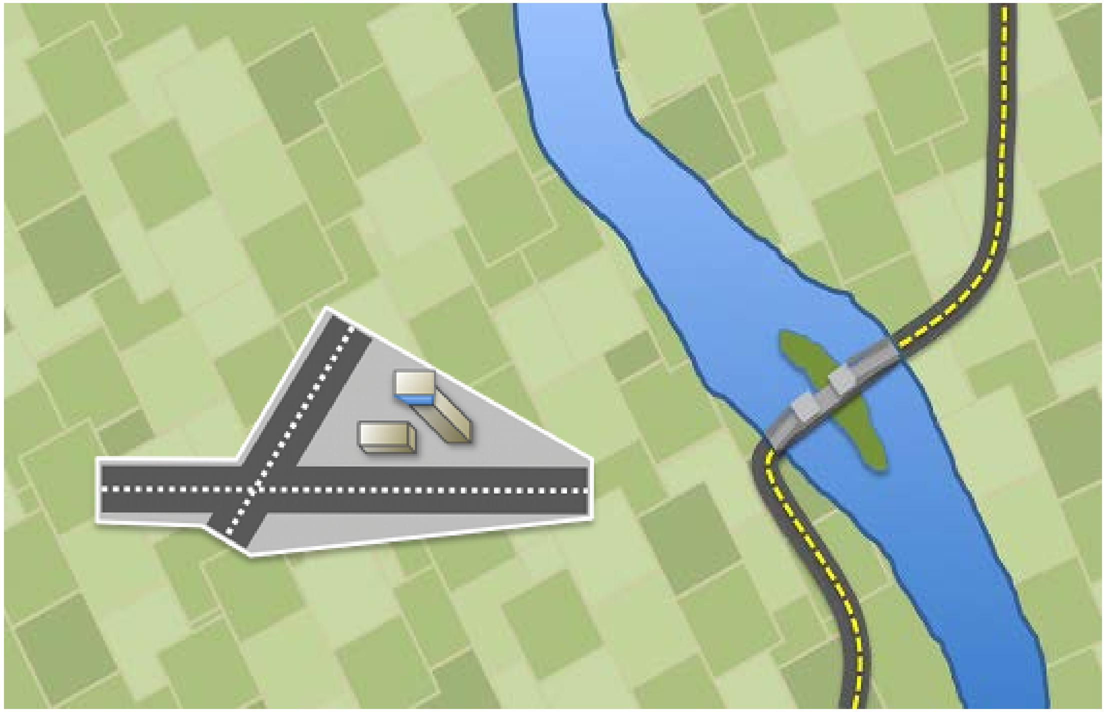
Thanks

- **Kene Akametalu**
- Anil Aswani (now at IEOR, UC Berkeley)
- **Max Balandat**
- Patrick Bouffard (now at Airware)
- **Young Hwan Chang**
- **Mo Chen**
- Jerry Ding (now at UTRC)
- **Roel Dobbe**
- **Jaime Fisac**
- Jeremy Gillula (now at EFF)
- Gabe Hoffmann (now at Zee.Aero)
- **Qie Hu**
- Haomiao Huang (now at Kuna Systems)
- Soulaiman Itani (now at Atheer Labs)
- Maryam Kamgarpour (now at ETHZ)
- Shahab Kaynama (now at ClearPath)
- **Casey Mackin**
- **Frauke Oldewurtel**
- Michael Vitus (now at hiDOF)
- Steve Waslander (now at ME, University of Waterloo)
- **Insoon Yang**
- **Melanie Zeilinger**
- Wei Zhang (now at ECE, Ohio State University)

NSF
ONR
NIH
NASA
AFOSR

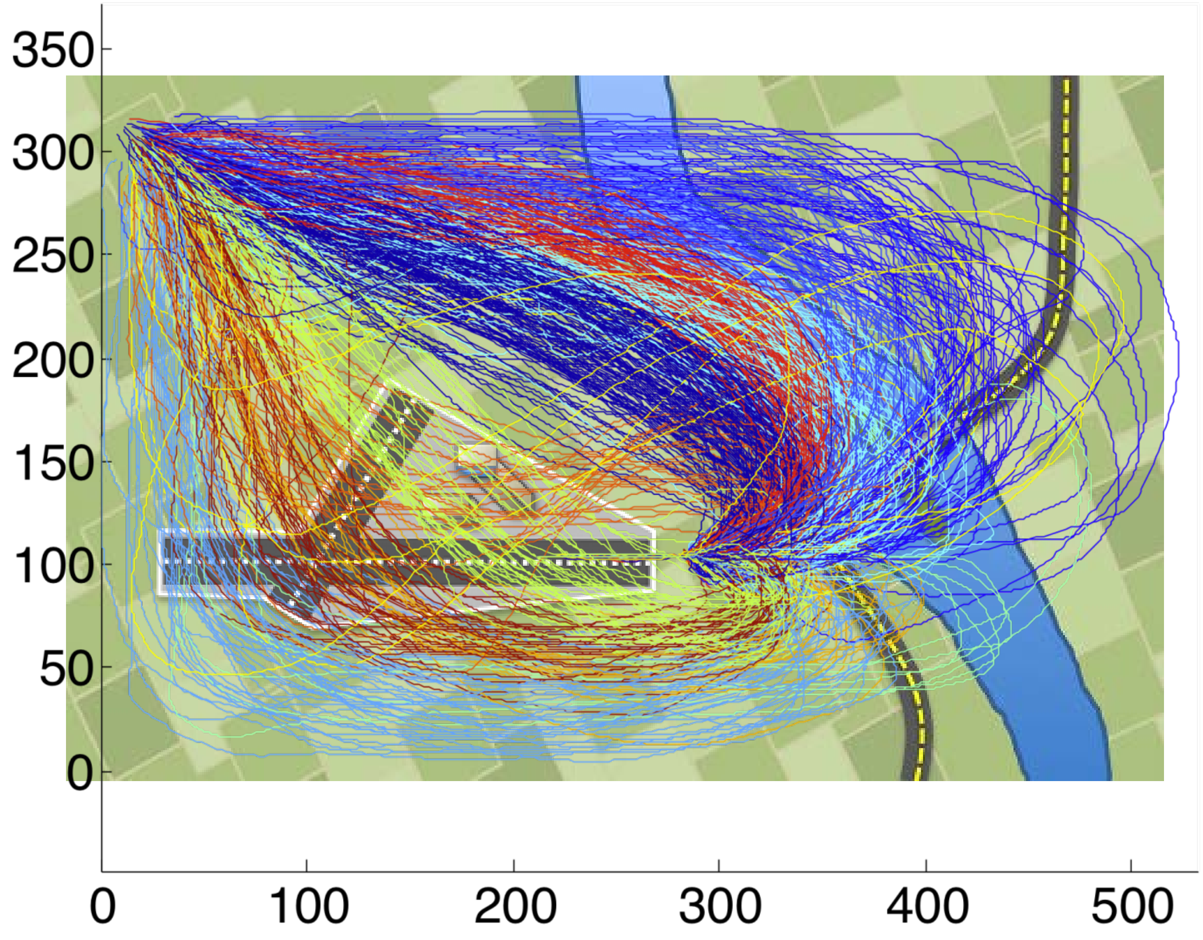
Modeling using gathered data

- Hypothesized hybrid model for controlled aircraft
- Data is supportive; clustering suggests discrete set of maneuvers used



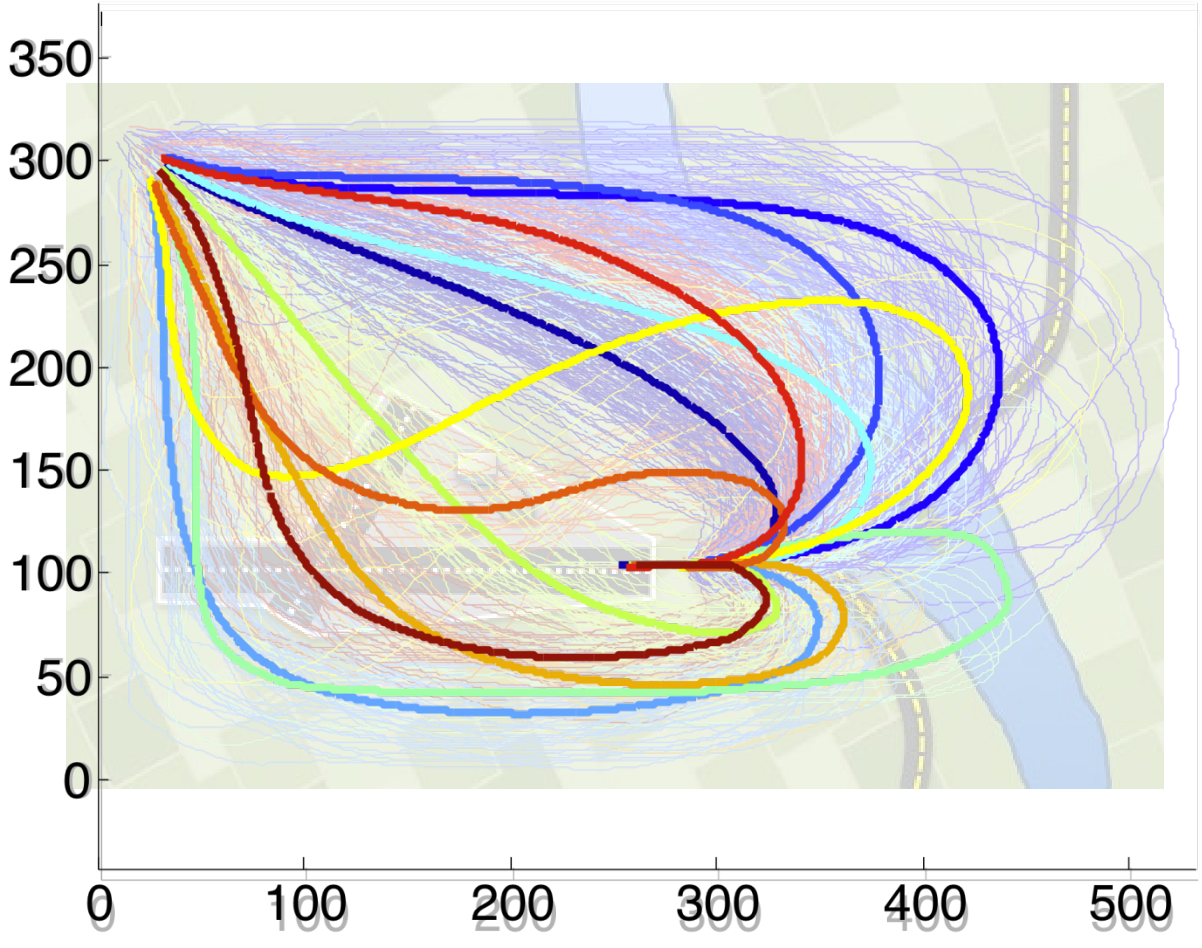
Modeling using gathered data

- Hypothesized hybrid model for controlled aircraft
- Data is supportive; clustering suggests discrete set of maneuvers used



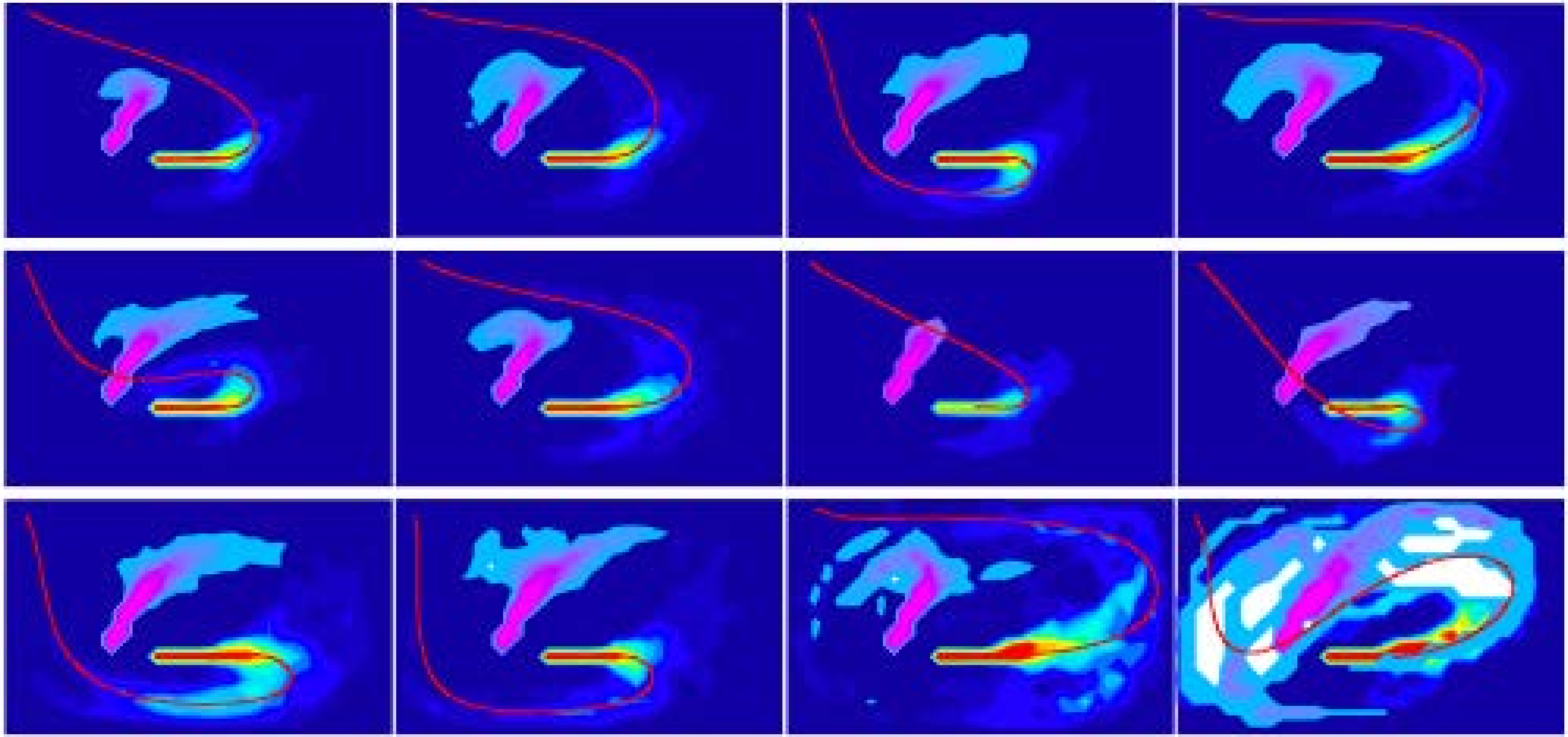
Modeling using gathered data

- Hypothesized hybrid model for controlled aircraft
- Data is supportive; clustering suggests discrete set of maneuvers used



Modeling using gathered data

- Predict the maneuver given the airspace
  - Avoidance maneuvers plotted on learned conditional airspace distributions
  - How people sequence moving objects

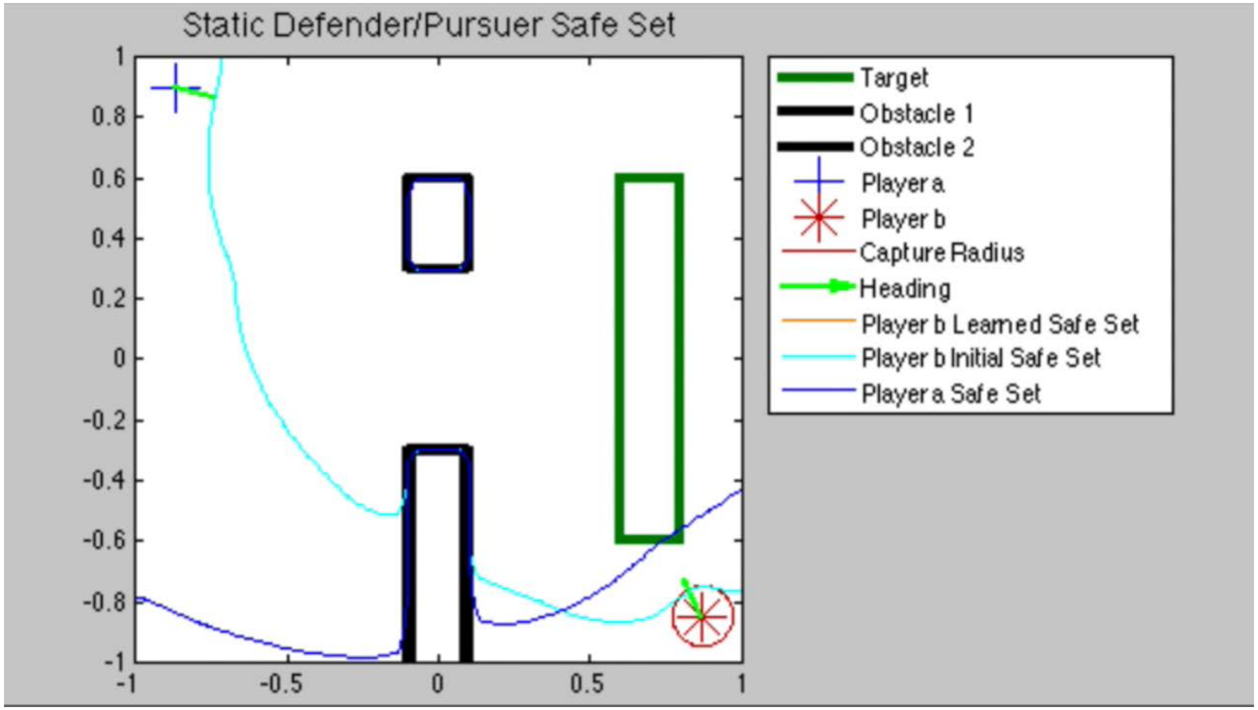


Example 7: Catch me if you can

For reachability $\mathcal{U} = \mathcal{D} = [0, 2] \times [0, 2\pi]$

Player a: (1) evade; (2) attack
$\mathcal{D}_{sim} = [0, 1] \times [0, 2\pi]$

Player b: (1) defend; (2) pursue
$\mathcal{U}_{sim} = [0, 2] \times [0, 2\pi]$



Example 7: Catch me if you can

For reachability $\mathcal{U} = \mathcal{D} = [0, 2] \times [0, 2\pi]$

Player a: (1) evade; (2) attack
$\mathcal{D}_{sim} = [0, 1] \times [0, 2\pi]$

Player b: (1) defend; (2) pursue
$\mathcal{U}_{sim} = [0, 2] \times [0, 2\pi]$

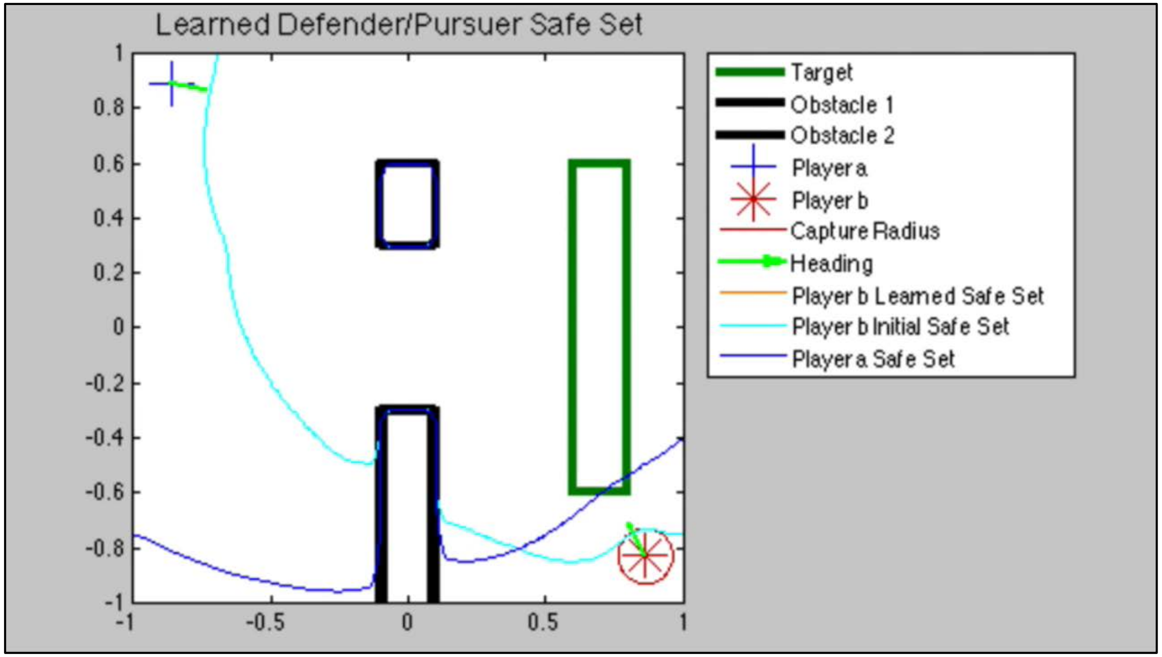
(with local updates)

Example 7: Catch me if you can

For reachability $\mathcal{U} = \mathcal{D} = [0, 2] \times [0, 2\pi]$

Player a: (1) evade; (2) attack
$\mathcal{D}_{sim} = [0, 1] \times [0, 2\pi]$

Player b: (1) defend; (2) pursue
$\mathcal{U}_{sim} = [0, 2] \times [0, 2\pi]$



(with local updates)