

Reach Sets and the Hamilton-Jacobi Equation

Ian Mitchell

Department of Computer Science
The University of British Columbia

Joint work with

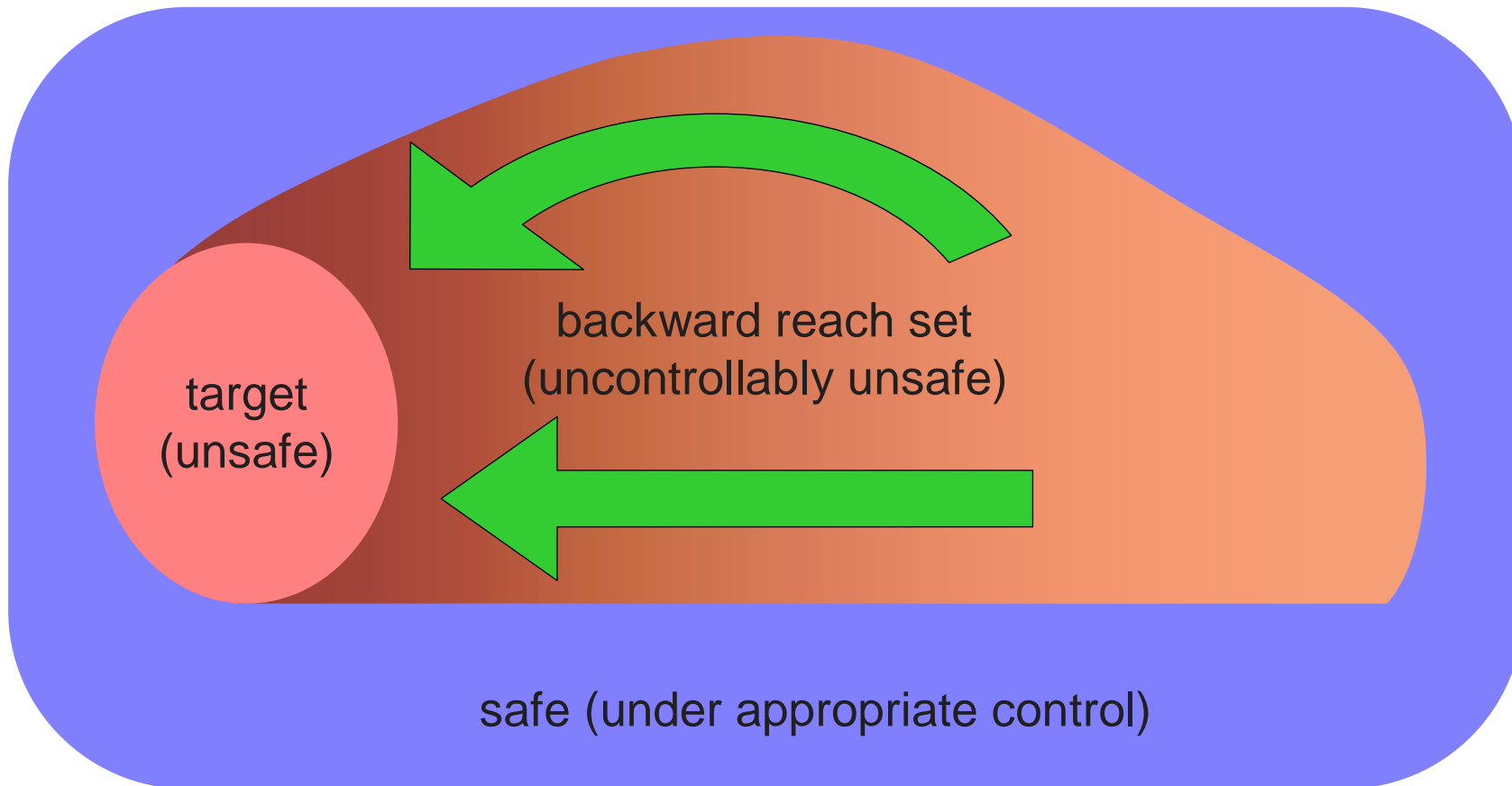
Alex Bayen, Meeko Oishi & Claire Tomlin (Stanford)

research supported by
National Science and Engineering Research Council of Canada
DARPA Software Enabled Control Project



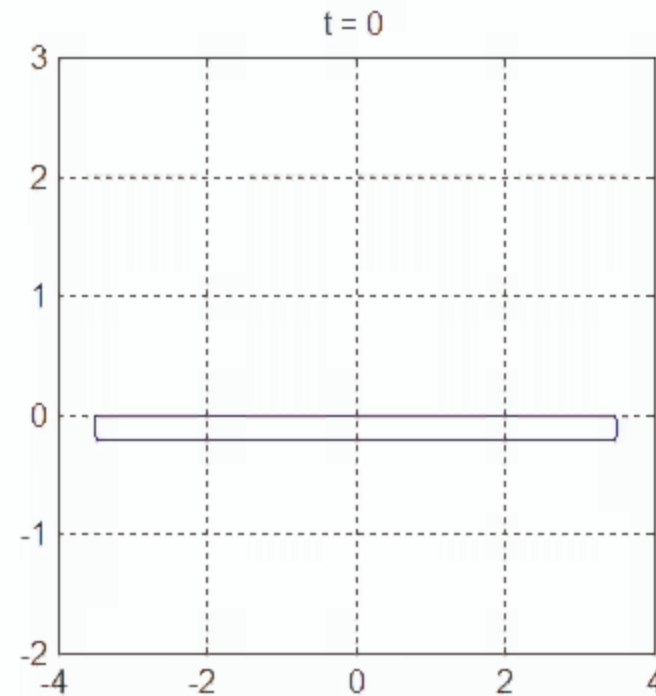
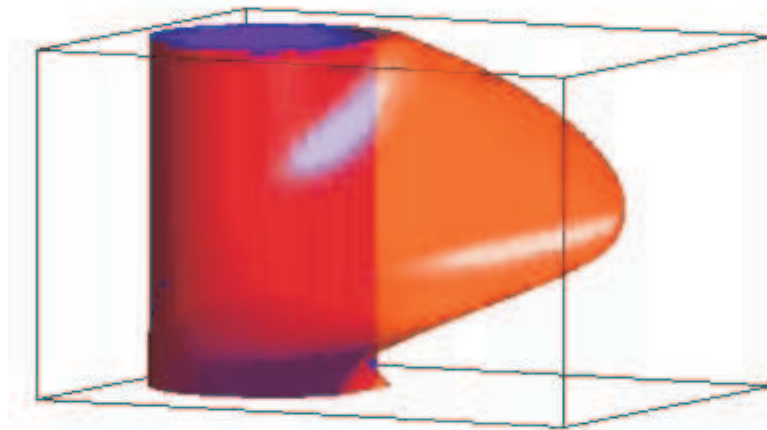
Reachable Sets: What and Why?

- One application: safety analysis
 - What states are doomed to become unsafe?
 - What states are safe given an appropriate control strategy?



Calculating Reach Sets

- Two primary challenges
 - How to represent set of reachable states
 - How to evolve set according to dynamics
- Discrete systems $x_{k+1} = \delta(x_k)$
 - Enumerate trajectories and states
 - Efficient representations: Binary Decision Diagrams
- Continuous systems $dx/dt = f(x)$?

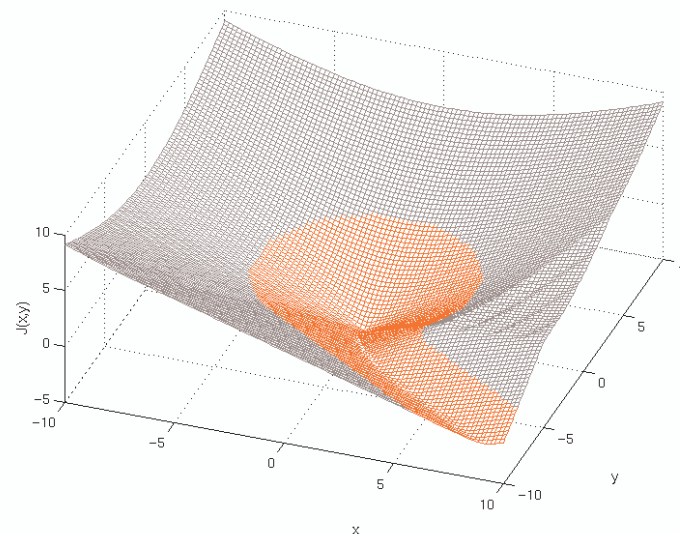
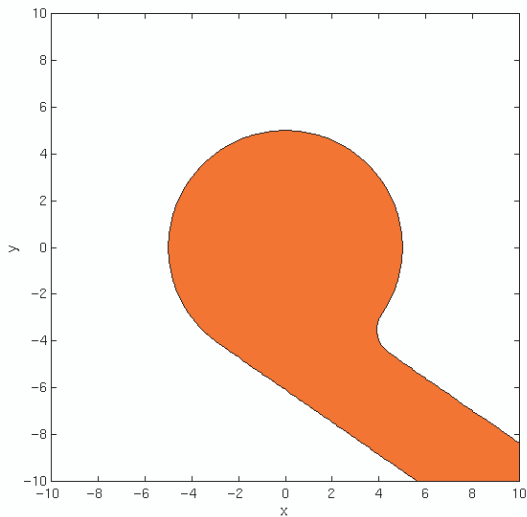


Implicit Surface Functions

- Set $G(t)$ is defined implicitly by an isosurface of a scalar function $\phi(x,t)$, with several benefits
 - State space dimension does not matter conceptually
 - Surfaces automatically merge and/or separate
 - Geometric quantities are easy to calculate

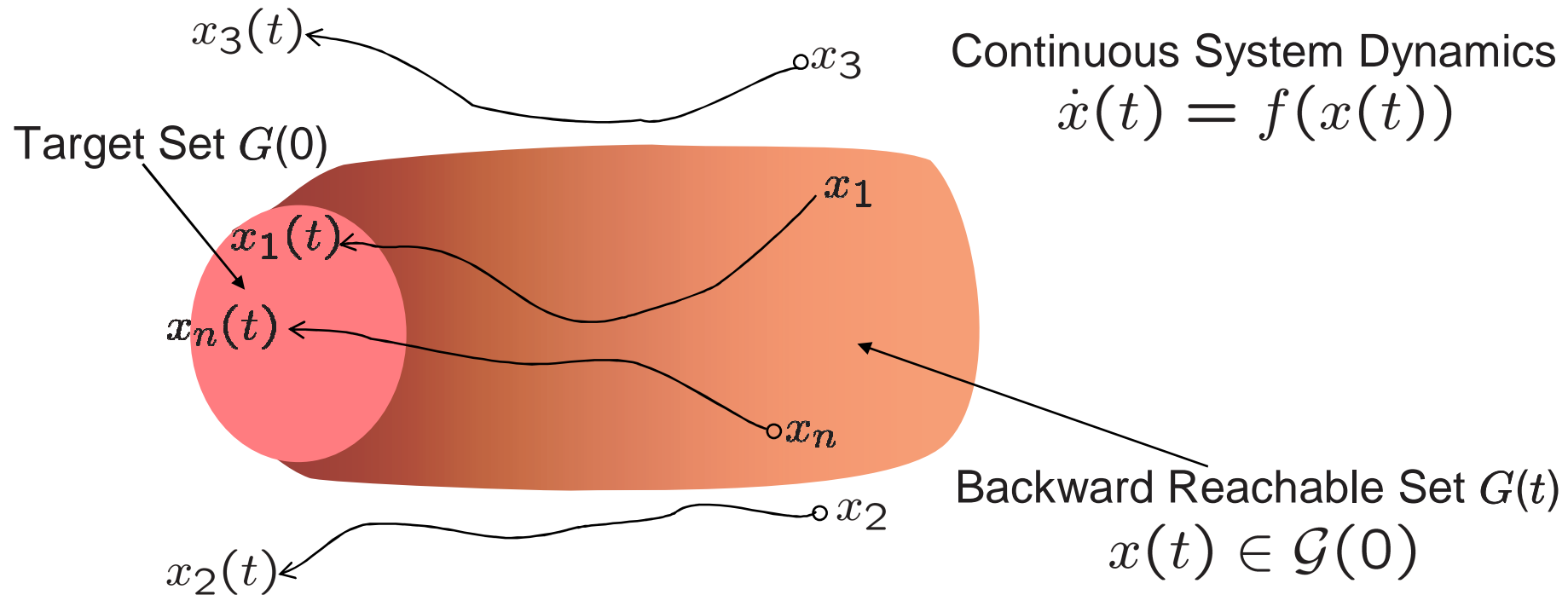
$$\phi : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$$

$$G(t) = \{x \in \mathbb{R}^n \mid \phi(x, t) \leq 0\}$$



Continuous Backward Reachable Sets

- Set of all states from which trajectories can reach some given target state
 - For example, what states can reach $G(t)$?

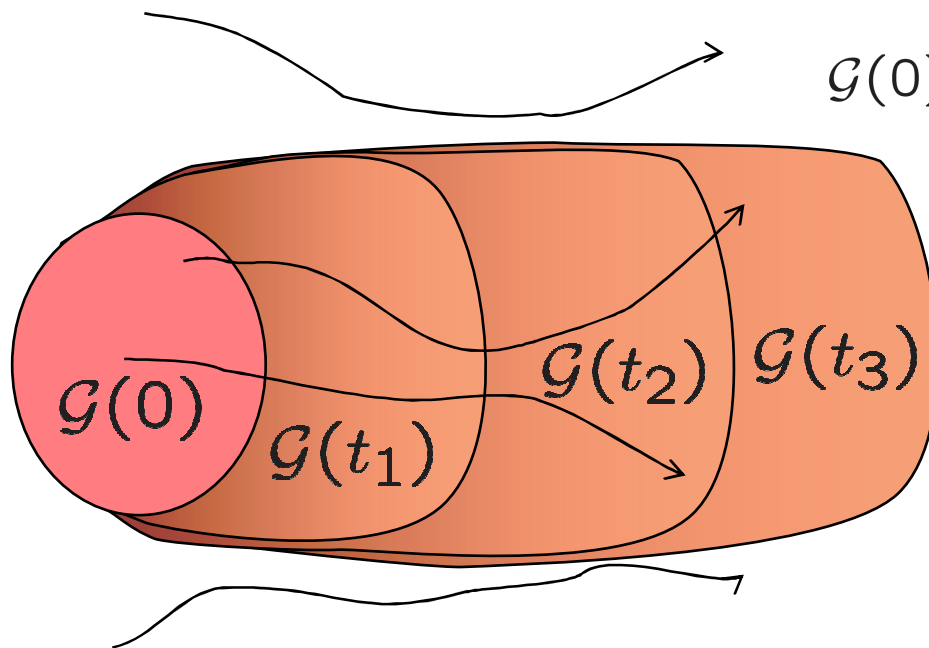


Why “Backward” Reachable Sets?

- To distinguish from forward reachable set
- To compute, run dynamics backwards in time from target set

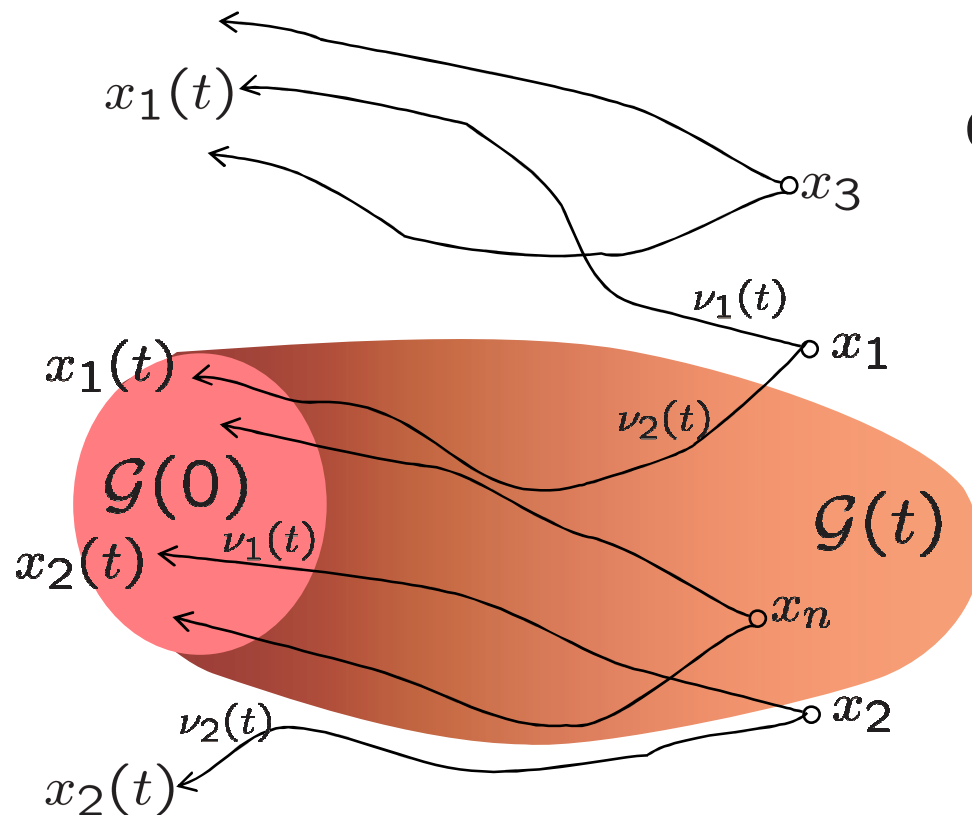
$$\dot{x}(t) = -f(x(t))$$

$$0 < t_1 < t_2 < t_3$$
$$\mathcal{G}(0) \subseteq \mathcal{G}(t_1) \subseteq \mathcal{G}(t_2) \subseteq \mathcal{G}(t_3)$$



Reachable Sets (controlled input)

- For most of our examples, target set is unsafe
- If we can control the input, choose it to avoid the target set
- Backward reachable set is unsafe no matter what we do



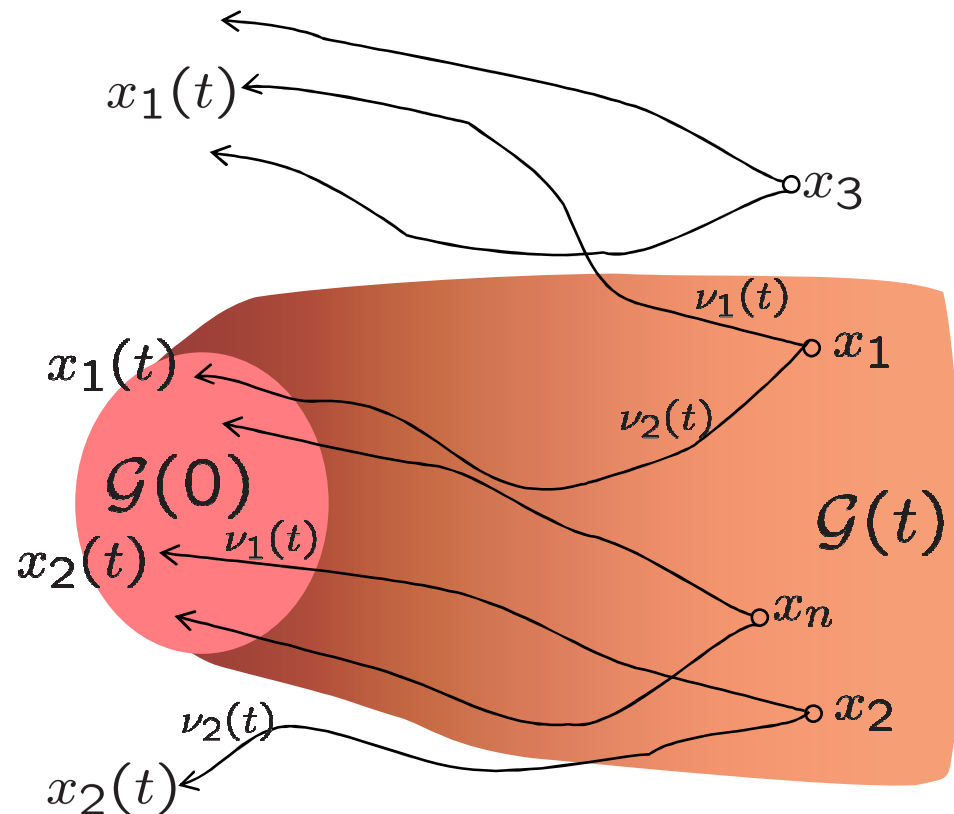
Continuous System Dynamics

$$\dot{x}(t) = f(x(t), \nu(t))$$

$$\forall \nu(\cdot), x(t) \in \mathcal{G}(0)$$

Reachable Sets (uncontrolled input)

- Sometimes we have no control over input signal
 - noise, actions of other agents, unknown system parameters
- It is safest to assume the worst case



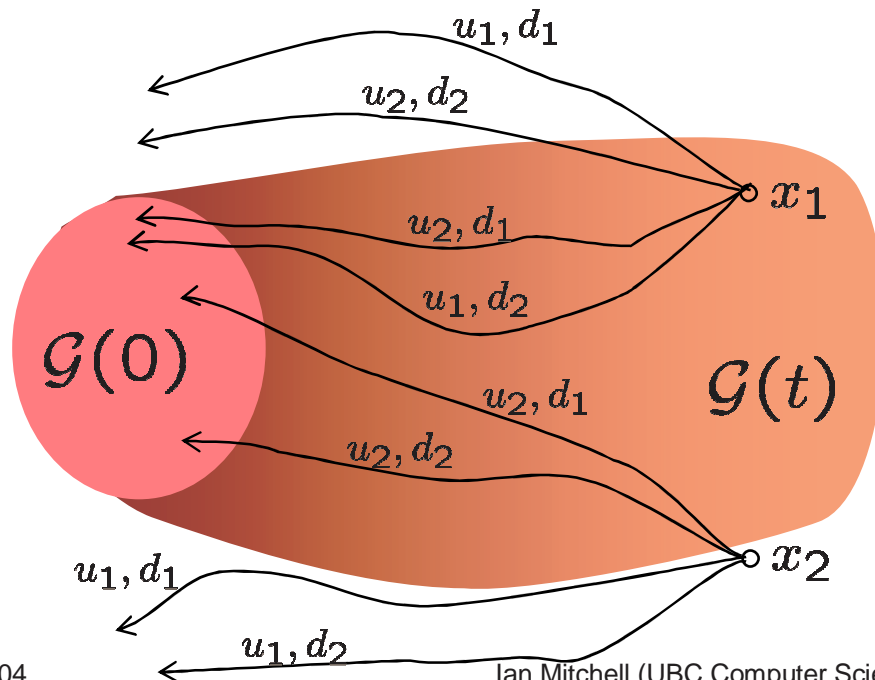
Continuous System Dynamics

$$\dot{x}(t) = f(x(t), \nu(t))$$

$$\exists \nu(\cdot), x(t) \in \mathcal{G}(0)$$

Two Competing Inputs

- For some systems there are two classes of inputs $v = (u, d)$
 - Controllable inputs $u \in U$
 - Uncontrollable (disturbance) inputs $d \in D$
- Equivalent to a zero sum differential game formulation
 - If there is an advantage to input ordering, give it to disturbances



Continuous System Dynamics
 $\dot{x}(t) = f(x(t), u(t), d(t))$

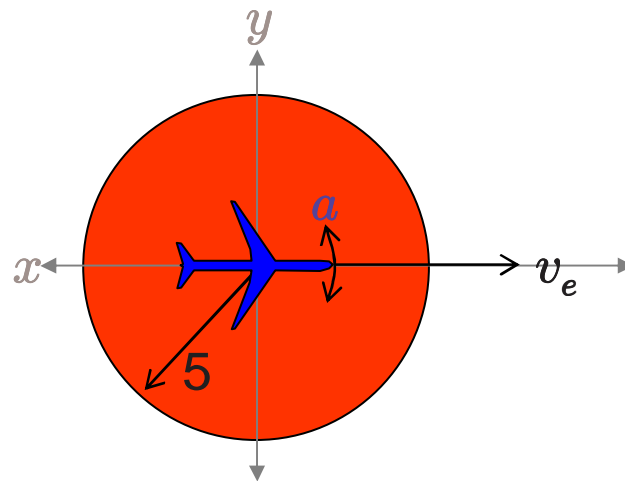
$\forall u(\cdot), \exists d(\cdot), x(t) \in \mathcal{G}(0)$

Game of Two Identical Vehicles

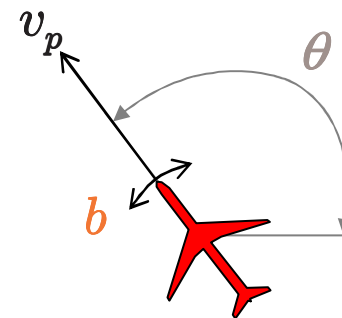
- Classical collision avoidance example
 - Collision occurs if vehicles get within five units of one another
 - Evader chooses turn rate $|a| \leq 1$ to avoid collision
 - Pursuer chooses turn rate $|b| \leq 1$ to cause collision
 - Fixed equal velocity $v_e = v_p = 5$

dynamics (pursuer)

$$\frac{d}{dt} \begin{bmatrix} x_p \\ y_p \\ \theta_p \end{bmatrix} = \begin{bmatrix} v_p \cos \theta_p \\ v_p \sin \theta_p \\ b \end{bmatrix}$$



evader aircraft (control)

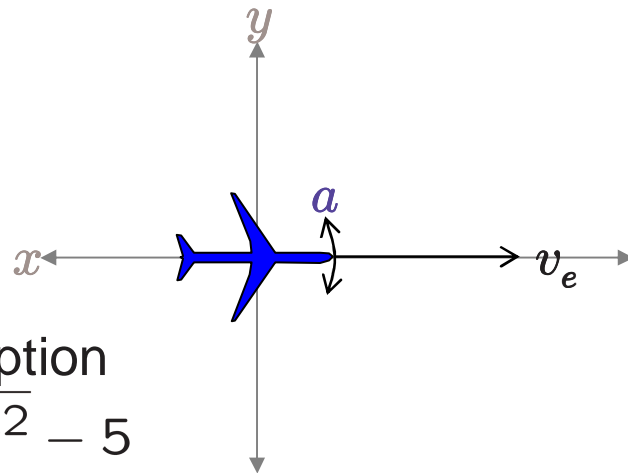


pursuer aircraft (disturbance)

Collision Avoidance Computation

- Work in relative coordinates with evader fixed at origin
 - State variables are now relative planar location (x,y) and relative heading ψ

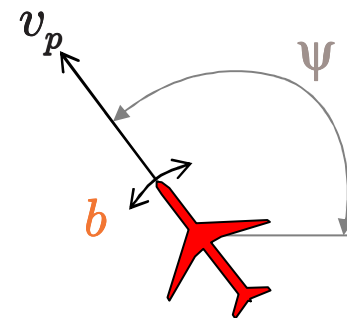
$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} = \begin{bmatrix} -v_e + v_p \cos \psi - ay \\ v_p \sin \psi - ax \\ b - a \end{bmatrix}$$



evader aircraft (control)

target set description

$$h(x) = \sqrt{x^2 + y^2} - 5$$



pursuer aircraft (disturbance)

Evolving Reachable Sets

- Modified Hamilton-Jacobi partial differential equation

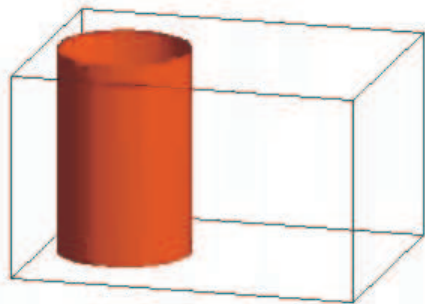
$$D_t \phi(x, t) + \min [0, H(x, D_x \phi(x, t))] = 0$$

$$\text{with Hamiltonian : } H(x, p) = \max_{a \in \mathcal{A}} \min_{b \in \mathcal{B}} f(x, a, b) \cdot p$$

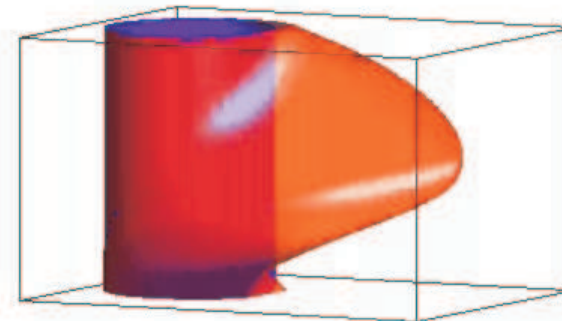
$$\text{and terminal conditions : } \phi(x, 0) = h(x)$$

$$\text{where } G(0) = \{x \in \mathbb{R}^n \mid h(x) \leq 0\}$$

$$\text{and } \dot{x} = f(x, a, b)$$



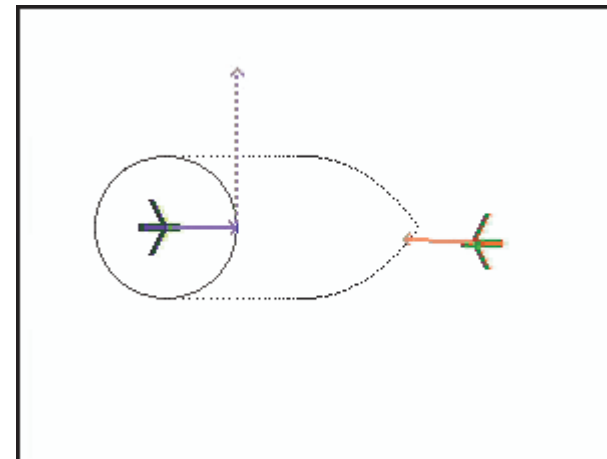
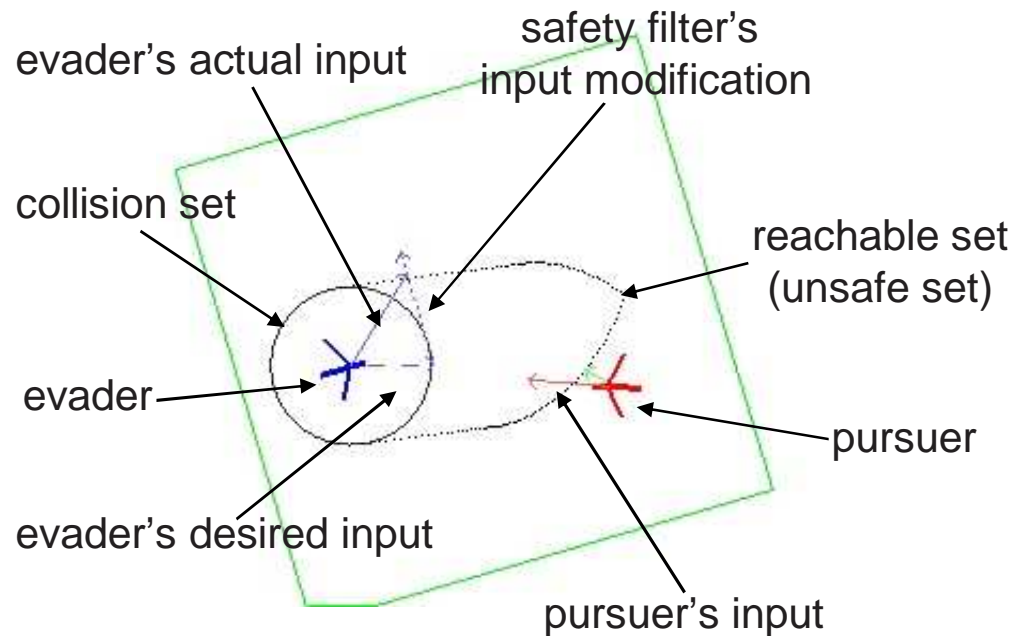
growth of reachable set



final reachable set

Application: Softwalls for Aircraft Safety

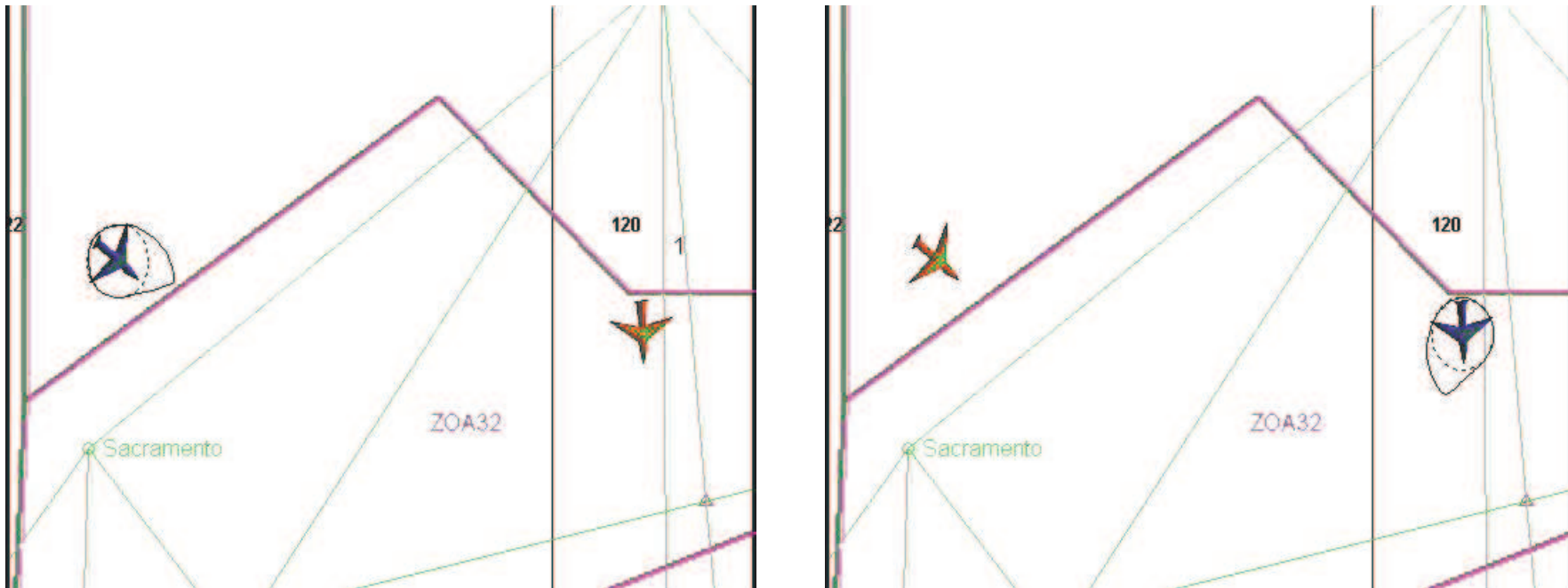
- Use reachable sets to guarantee safety
- Basic Rules
 - Pursuer: turn to head toward evader
 - Evader: turn to head east
- Evader's input is filtered to guarantee that pursuer does not enter the reachable set



joint work with Edward Lee & Adam Cataldo

Application: Collision Alert for ATC

- Use reachable set to detect potential collisions and warn Air Traffic Control (ATC)
 - Find aircraft pairs in ETMS database whose flight plans intersect
 - Check whether either aircraft is in the other's collision region
 - If so, examine ETMS data to see if aircraft path is deviated
 - One hour sample in Oakland center's airspace—
 - 1590 pairs, 1555 no conflict, 25 detected conflicts, 2 false alerts



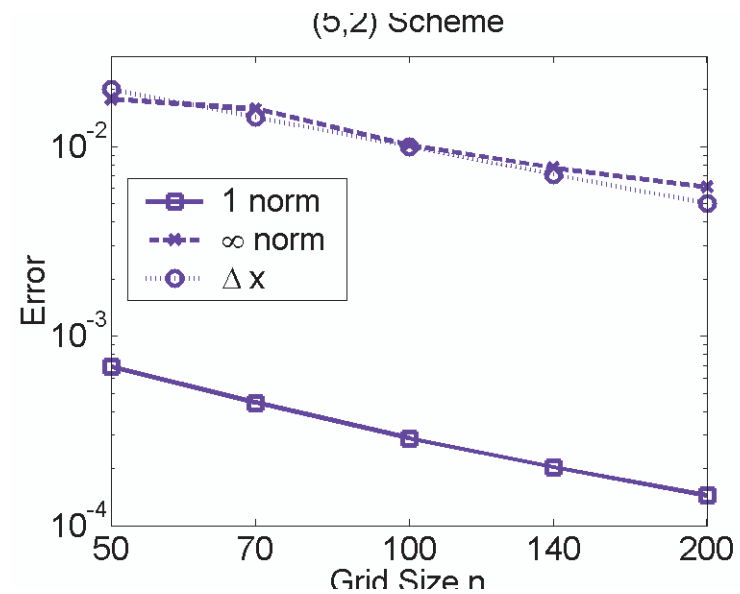
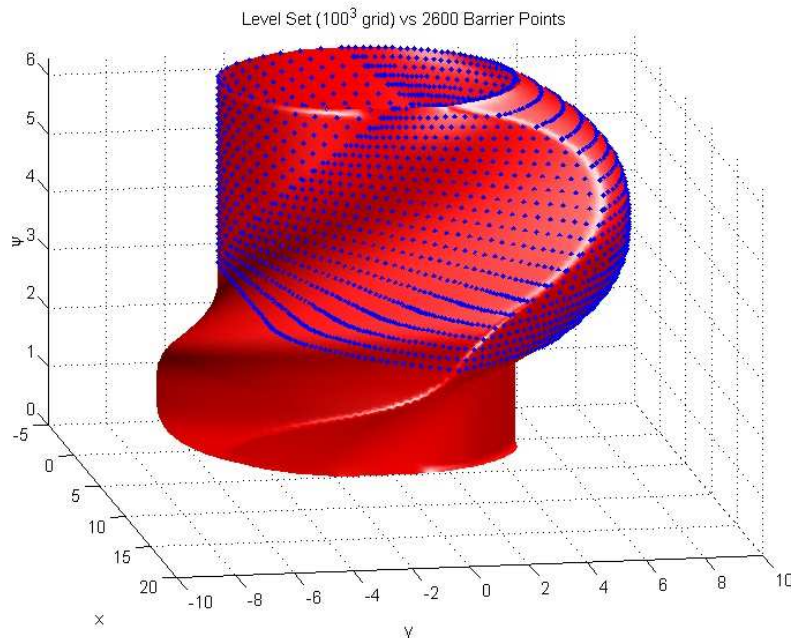
24 Oct 04

Ian Mitchell (UBC Computer Science)

14

Validating the Numerical Algorithm

- Analytic solution for reachable set can be found [Merz, 1972]
 - Applies only to identical pursuer and evader dynamics
 - Merz's solution placed pursuer at the origin, game is not symmetric
 - Analytic solution can be used to validate numerical solution
 - [Mitchell, 2001]



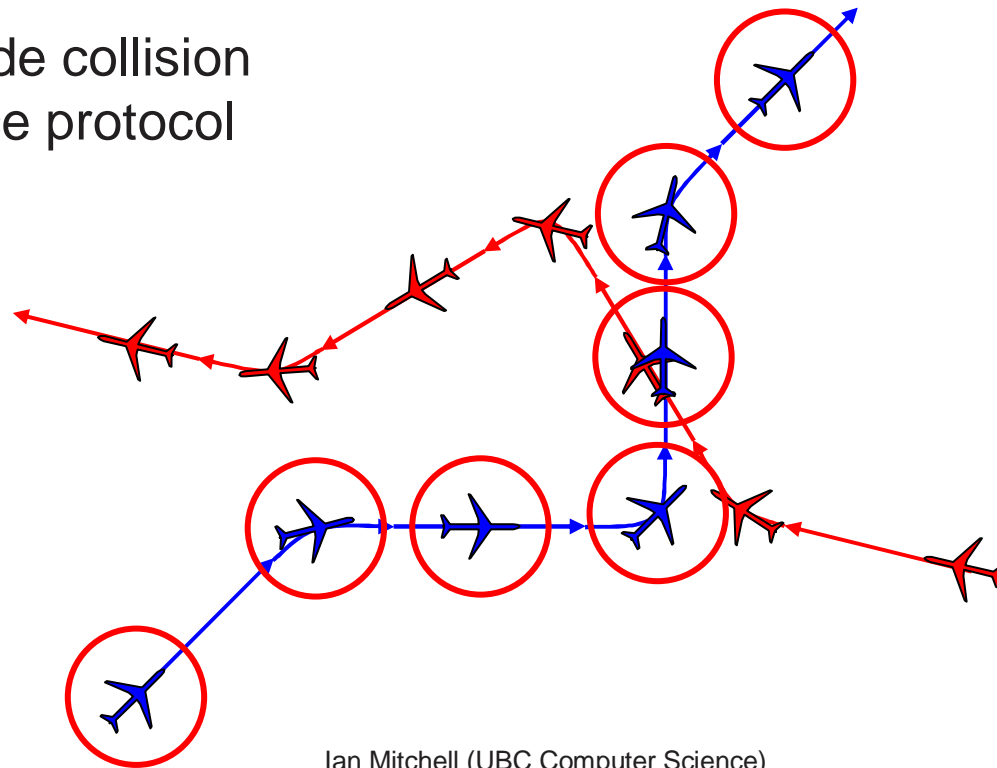
Hybrid System Reach Sets

Combining Continuous and Discrete
Evolution

Why Hybrid Systems?

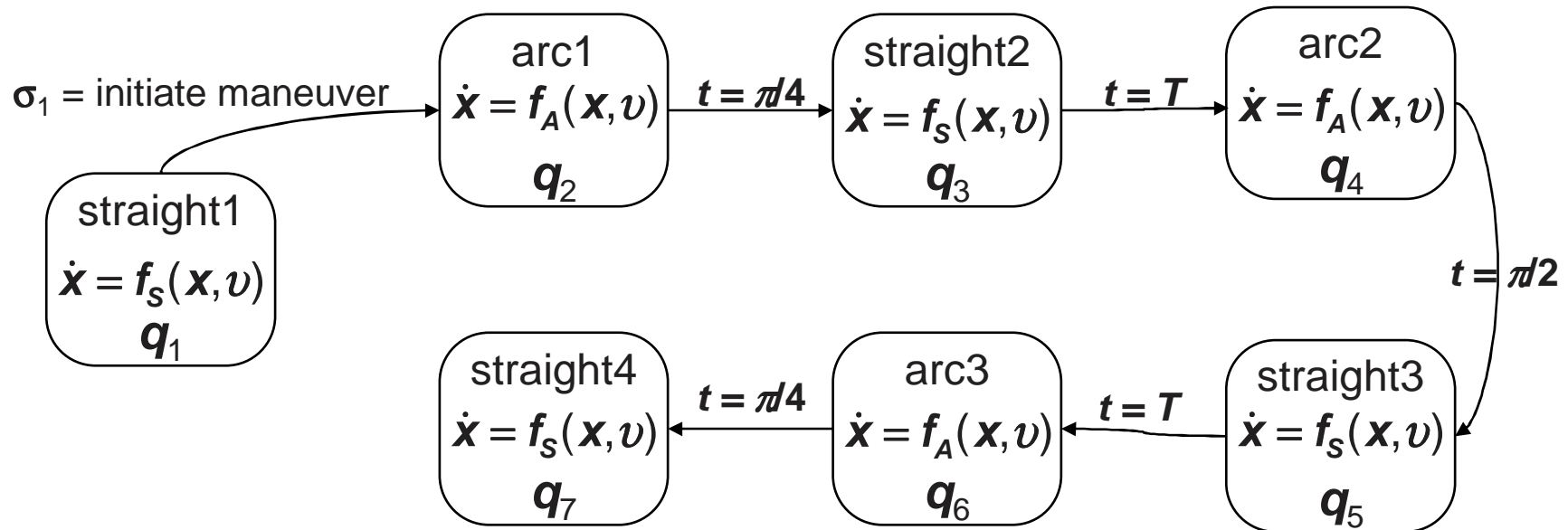
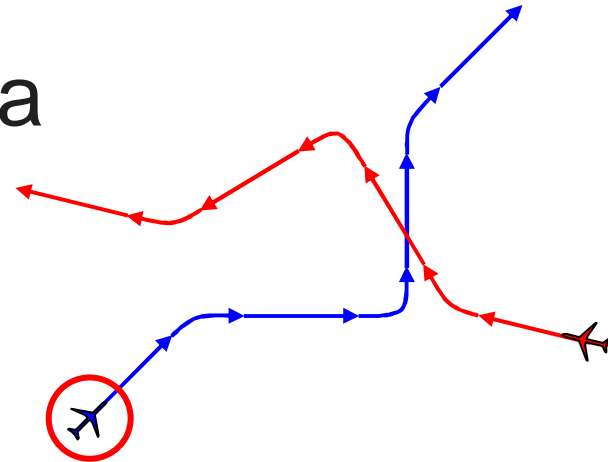
- Computers are increasingly interacting with external world
 - Flexibility of such combinations yields huge design space
 - Design methods and tools targeted (mostly) at either continuous or discrete systems
- Example: aircraft flight control systems

seven mode collision avoidance protocol



Hybrid Automata

- Discrete modes and transitions
- Continuous evolution within each mode



$$\mathbf{f}_S \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} -\nu + \nu \cos \psi \\ \nu \sin \psi \end{pmatrix}$$

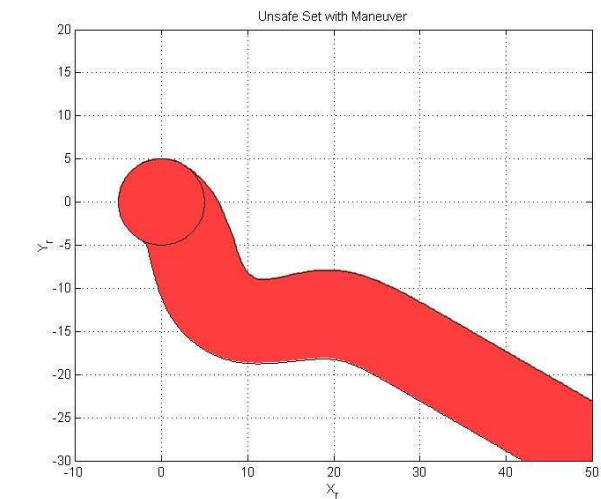
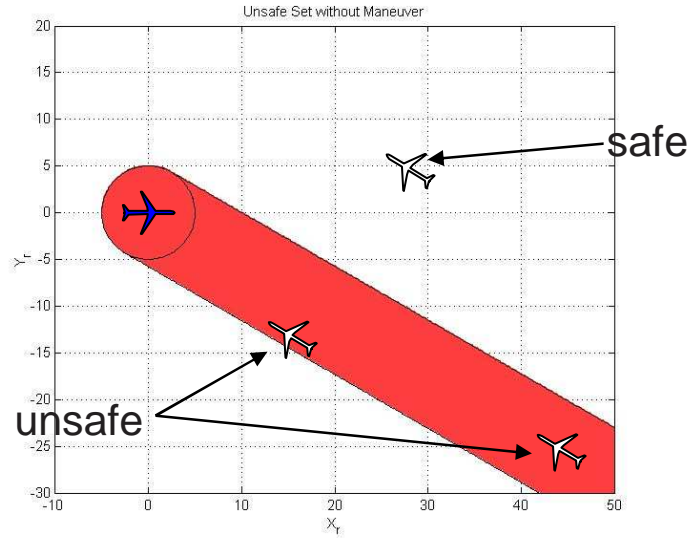
dynamics in straight modes

$$\mathbf{f}_A \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} -\nu + \nu \cos \psi - \mathbf{x}_2 \\ \nu \sin \psi + \mathbf{x}_1 \end{pmatrix}$$

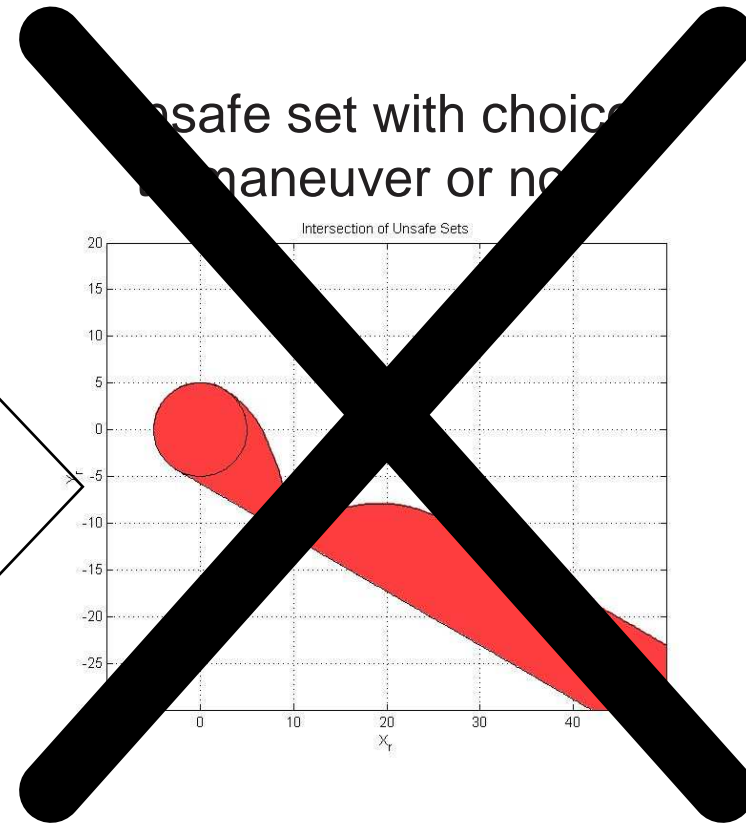
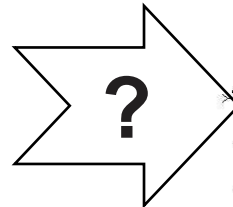
dynamics in arc modes

Seven Mode Safety Analysis

unsafe set without maneuver

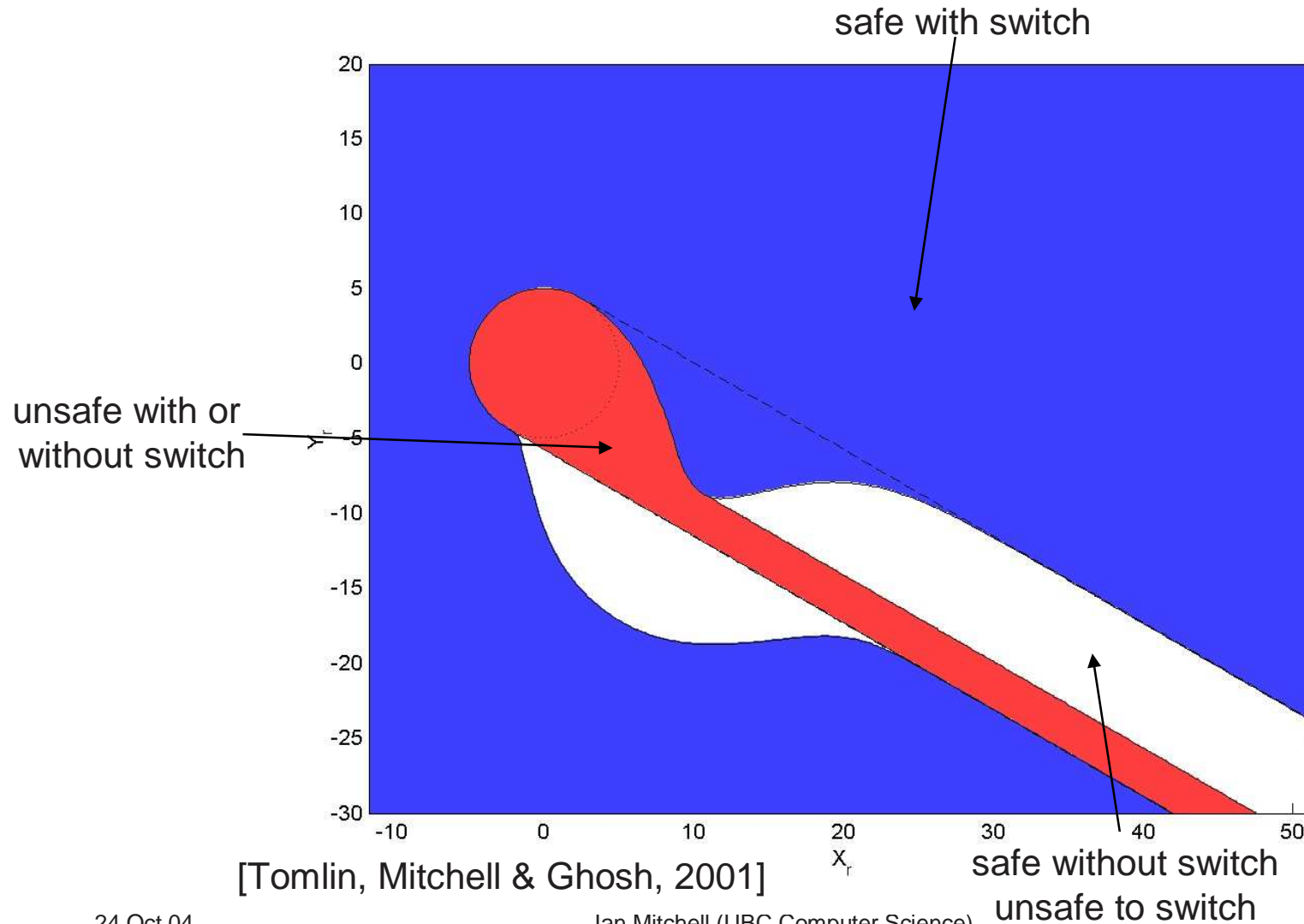


unsafe set with maneuver



Seven Mode Safety Analysis

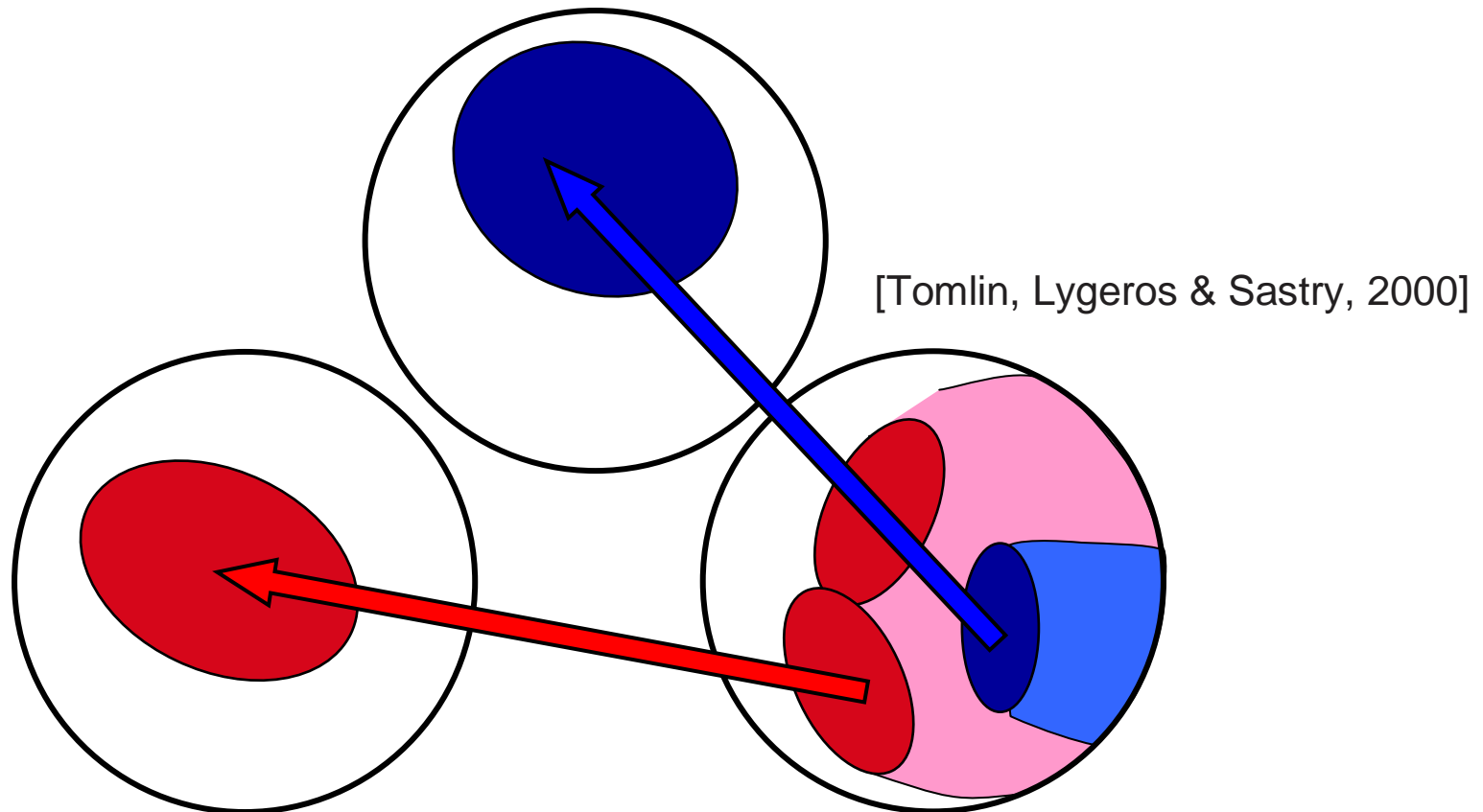
- Ability to choose maneuver start time further reduces unsafe set



[Tomlin, Mitchell & Ghosh, 2001]

Computing Hybrid Reachable Sets

- Compute continuous reachable set in each mode separately
 - Uncontrollable switches may introduce unsafe sets
 - Controllable switches may introduce safe sets
 - Forced switches introduce boundary conditions

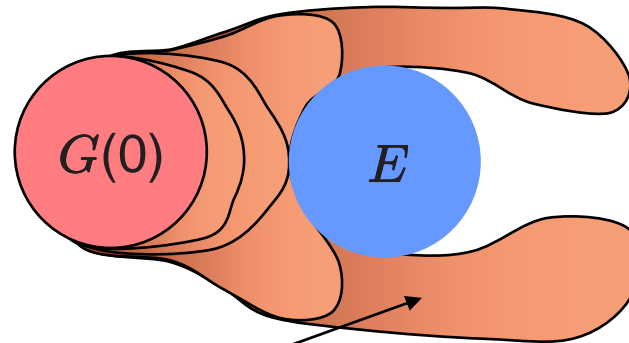


Reach-Avoid Operator

- Compute set of states which reaches $G(0)$ without entering E

$$G(t) = \{x \in \mathbb{R}^n \mid \phi_G(x, t) \leq 0\}$$

$$E = \{x \in \mathbb{R}^n \mid \phi_E(x) \leq 0\}$$



Reach-Avoid Set $G(t)$

- Formulated as a constrained Hamilton-Jacobi equation or variational inequality
 - [Mitchell & Tomlin, 2000]

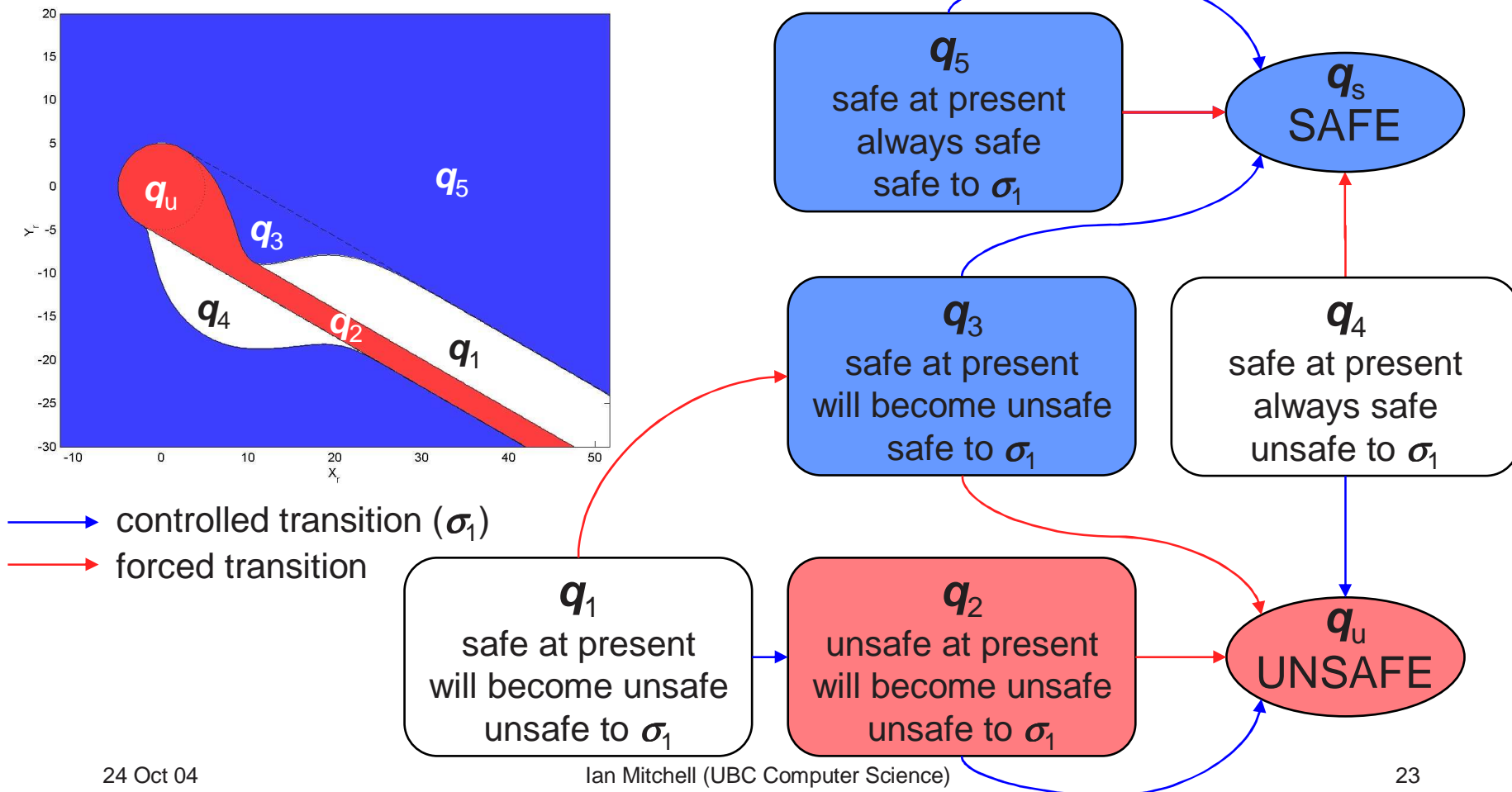
$$D_t \phi_G(x, t) + \min [0, H(x, D_x \phi_G(x, t))] = 0$$

$$\text{subject to: } \phi_G(x, t) \geq \phi_E(x)$$

- Level set can represent often odd shape of reach-avoid sets

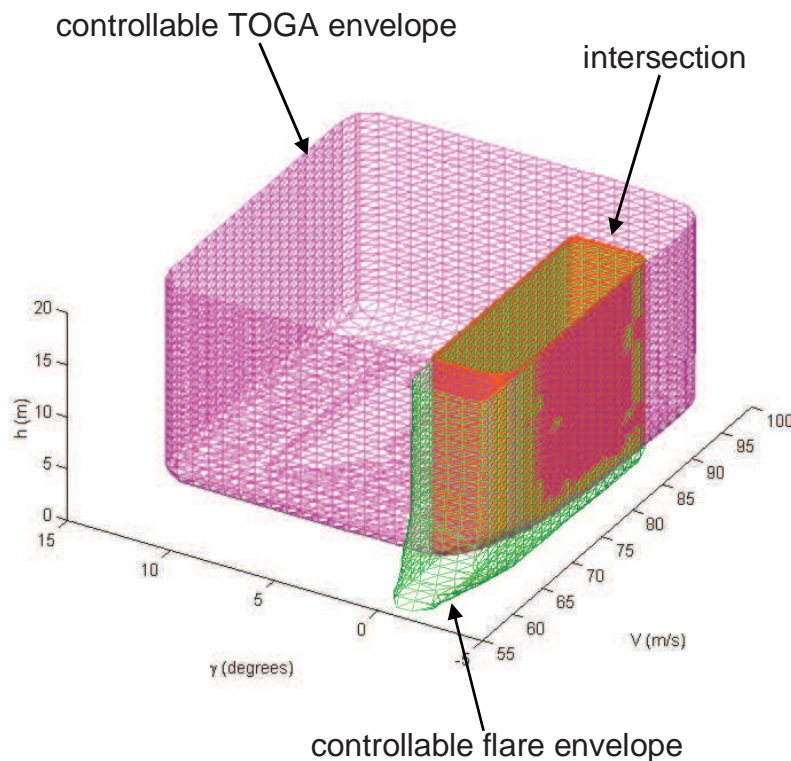
Application: Discrete Abstractions

- It can be easier to analyze discrete automata than hybrid automata or continuous systems
 - Use reachable set information to abstract away continuous details

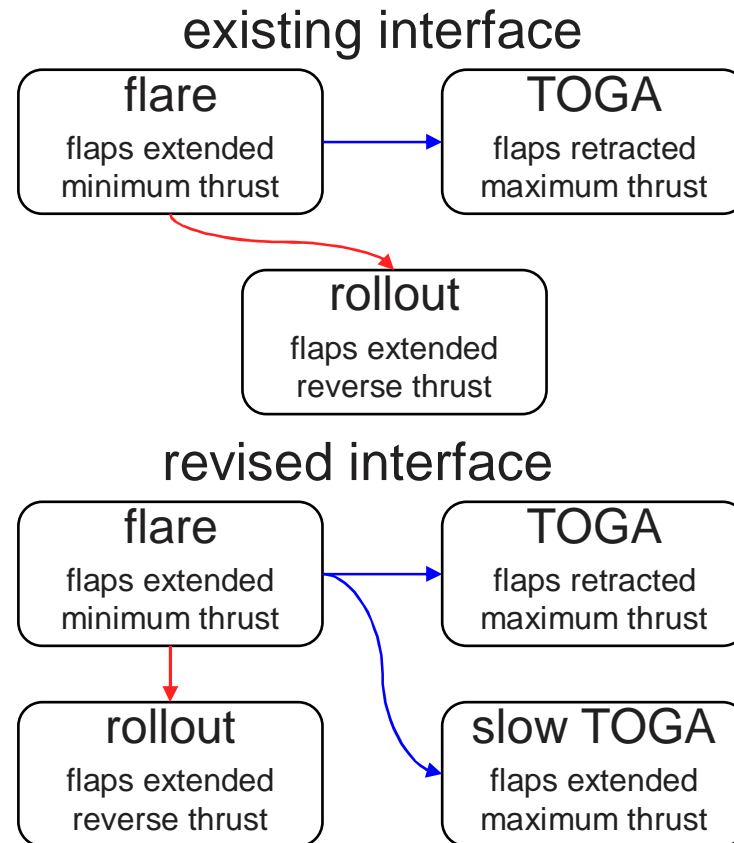


Application: Cockpit Display Analysis

- Controllable flight envelopes for landing and Take Off / Go Around (TOGA) maneuvers may not be the same
- Pilot's cockpit display may not contain sufficient information to distinguish whether TOGA can be initiated



24 Oct 04

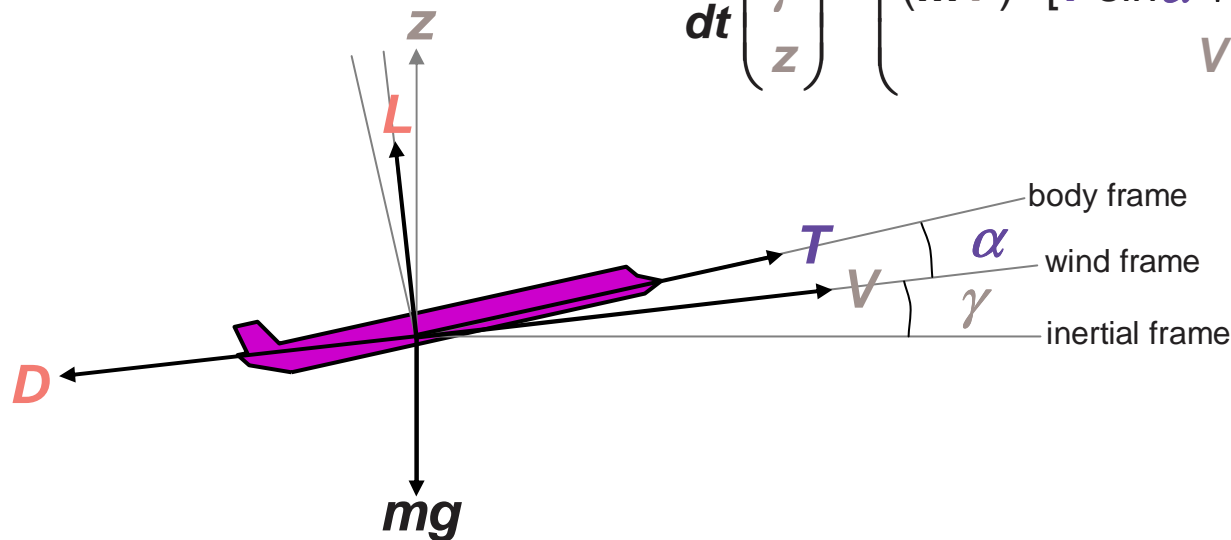


24

Application: Aircraft Autolander

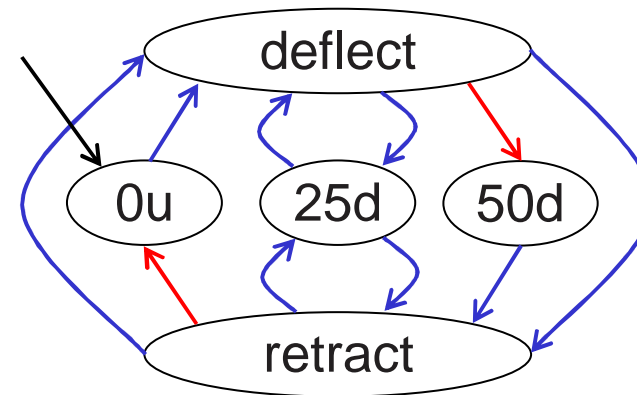
- Airplane must stay within safe flight envelope during landing
 - Bounds on velocity (V), flight path angle (γ), height (z)
 - Control over engine thrust (T), angle of attack (α), flap settings
 - Model flap settings as discrete modes of hybrid automata
 - Terms in continuous dynamics may depend on flap setting
 - [Mitchell, Bayen & Tomlin, 2001]

$$\frac{d}{dt} \begin{pmatrix} V \\ \gamma \\ z \end{pmatrix} = \begin{pmatrix} m^{-1}[T \cos \alpha - D(\alpha, V) - mg \sin \gamma] \\ (mV)^{-1}[T \sin \alpha + L(\alpha, V) - mg \cos \gamma] \\ V \sin \gamma \end{pmatrix}$$



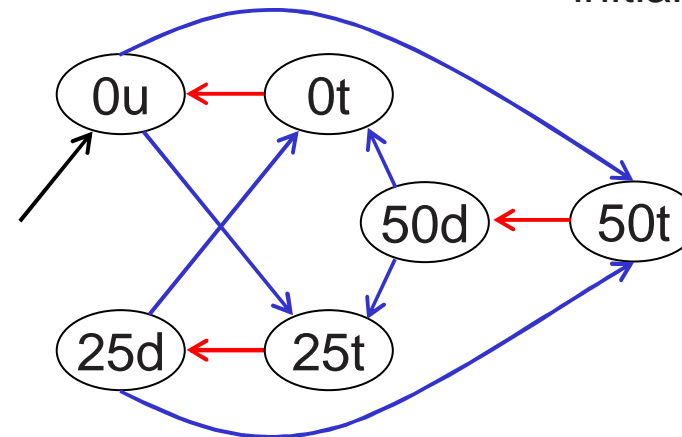
Landing Example: Discrete Model

- Flap dynamics version
 - Pilot can choose one of three flap deflections
 - Thirty seconds for zero to full deflection



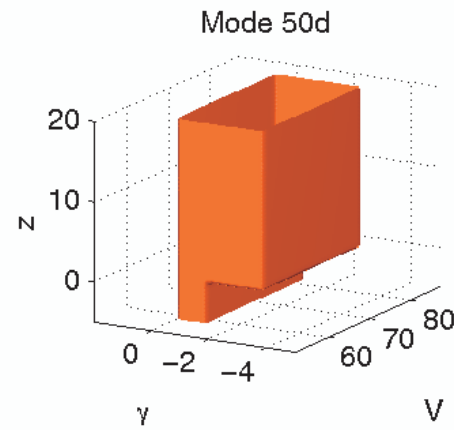
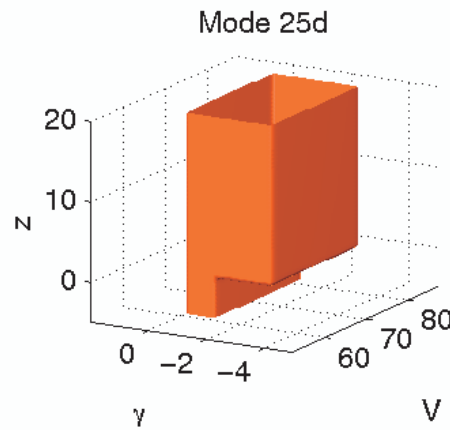
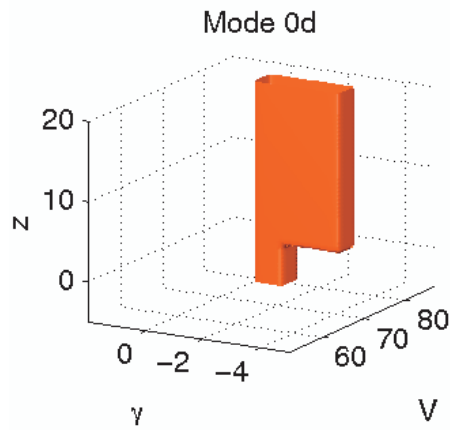
← controlled
← forced
← initial

- Implemented version
 - Instant switches between fixed deflections
 - Additional timed modes to remove Zeno behavior

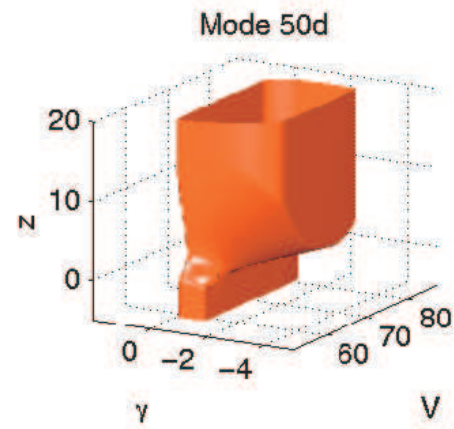
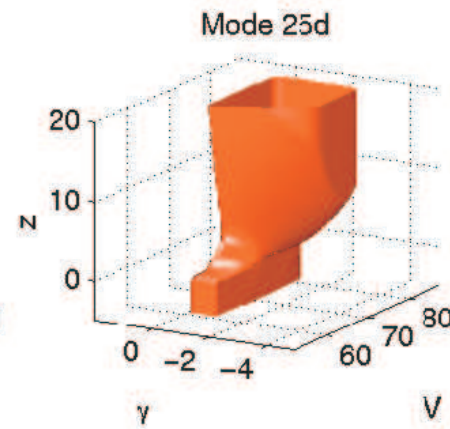
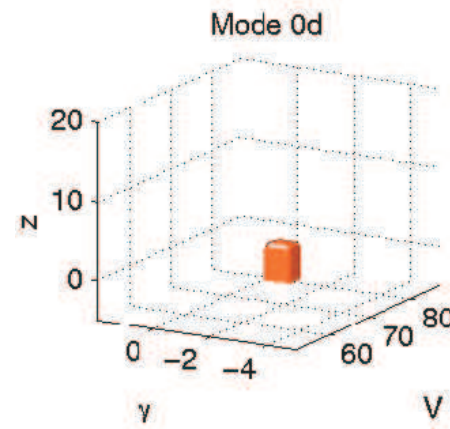


Landing Example: No Mode Switches

Envelopes

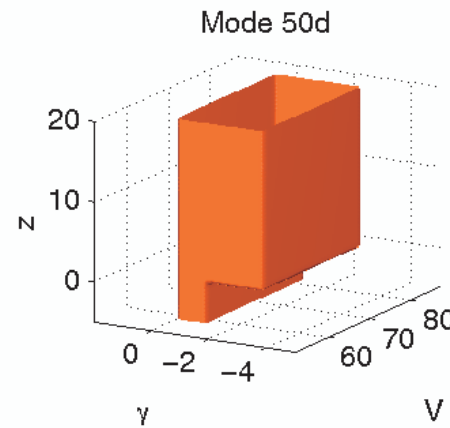
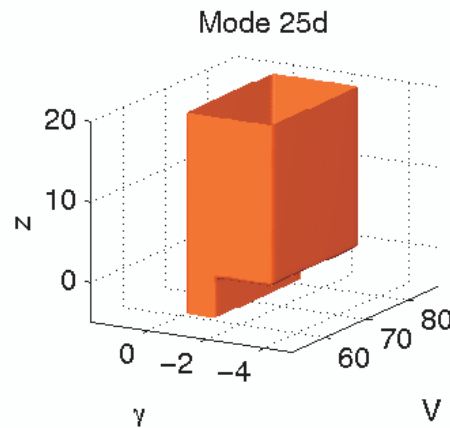
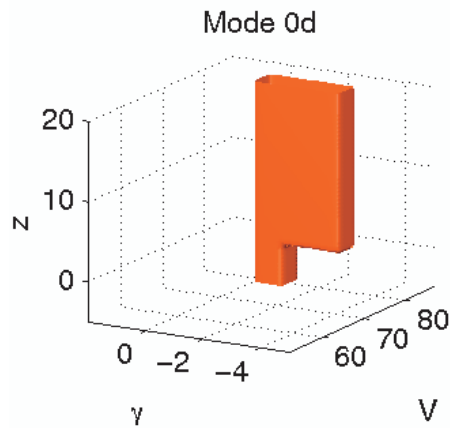


Safe sets

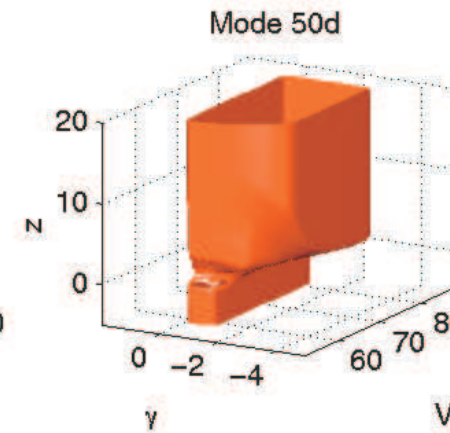
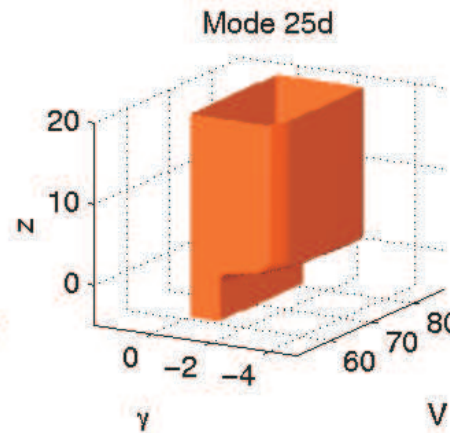
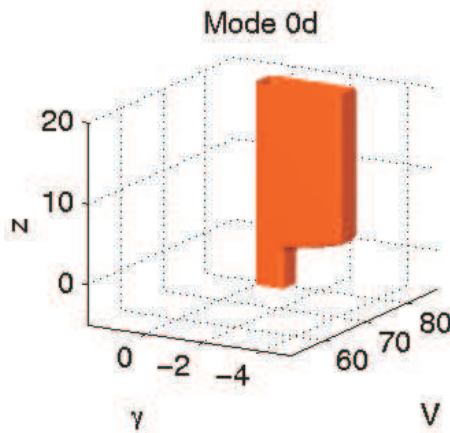


Landing Example: Mode Switches

Envelopes



Safe sets



Landing Example: Synthesizing Control

- For states at the boundary of the safe set, results of reach-avoid computation determine
 - What continuous inputs (if any) maintain safety
 - What discrete jumps (if any) are safe to perform
 - Level set values & gradients provide all relevant data

