EECE 571M/491M, Spring 2007
Lecture 18

Hybrid reachability

Meeko Oishi, Ph.D.

Electrical and Computer Engineering

University of British Columbia, BC

http://www.ece.ubc.ca/~elec571m.html

moishi@ece.ubc.ca

Tomlin LN 7-9; Tomlin, Mitchell, Bayen, Oishi (2003)

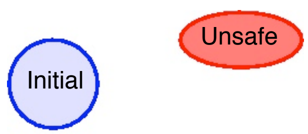
Today's lecture

- Background
  - Verification through reachability
  - Literature and tools survey
- Preliminaries
  - Discrete reachability
  - Continuous reachability
- Hybrid reachability algorithm
- Examples

Verification through Reachability

Verification

A mathematical proof that the system satisfies a property



1. **Reachable set**

   States for which the property does not hold

2. **Controller synthesis**

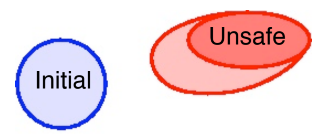
   Design of control laws to guarantee that the system satisfies the property

Methods give definite answers over all possible initial conditions

Verification through Reachability

Verification

A mathematical proof that the system satisfies a property



1. **Reachable set**

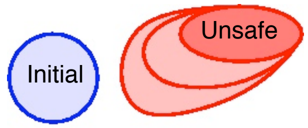
   States for which the property does not hold

2. **Controller synthesis**

   Design of control laws to guarantee that the system satisfies the property

Methods give definite answers over all possible initial conditions

Verification through Reachability



Verification

A mathematical proof that the system satisfies a property

1. **Reachable set**

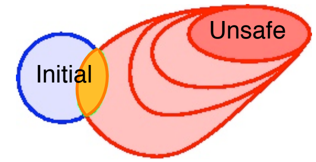
   States for which the property does not hold

2. **Controller synthesis**

   Design of control laws to guarantee that the system satisfies the property

Methods give definite answers over all possible initial conditions

Verification through Reachability



Verification

A mathematical proof that the system satisfies a property

1. **Reachable set**

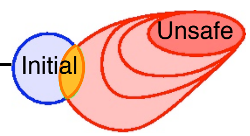
   States for which the property does not hold

2. **Controller synthesis**

   Design of control laws to guarantee that the system satisfies the property

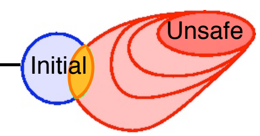
Methods give definite answers over all possible initial conditions

Computational tools



- Linear dynamics, piecewise affine systems
  - Multi-Parametric Toolbox for constrained linear dynamics (ETH Zurich -- Morari, Bemporad, Borelli, Grieder, others)
  - PHAVer for over-approximations of piecewise affine dynamics (Frehse)
  - MATISSE for large, constrained linear systems using approximate bisimulations (Girard, Pappas)
- Linear differential inclusions and timed automata
  - d/dt for linear differential inclusions (Dang, Maler)
  - HyTech for linear hybrid automata (Alur, Henzinger, Wong-Toi, Ho)
  - CheckMate (Chutinan, Krogh, et al)
  - KRONOS for timed automata (Hovine, Olivero, Daws, Tripakis)
  - UPPALL for timed automata (Larsen, Yi, Behrmann, et al)

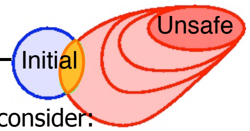
Computational tools



- Nonlinear dynamics
  - Level Set Toolbox for general nonlinear dynamics (Mitchell)
  - Viability tools for nonlinear differential inclusions (Aubin, Saint-Pierre, et al)
  - SOSTools for polynomial dynamics (Prajna, Papachristodoulou, Seiler, Parrilo)
- Discrete event systems
  - Murϕ (Dill et al.)
  - PVS (Rushby, Shankar, et al.)
  - others…

See the "Hybrid Systems Tools" wiki (G. Pappas): http://wiki.grasp.upenn.edu/~graspdoc/wiki/hst

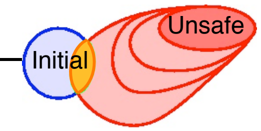
Computational tools



When selecting tools for a particular problem, consider:

- Type of dynamics (continuous, discrete, hybrid)
- Form of continuous dynamics (rectangular, linear, affine, polynomial, nonlinear)
- Type of sets (rectangular, linear, affine, polynomial, nonlinear)
- Type of inputs (if any; controlled vs. disturbance)
- Model-checking vs. synthesis
- Computational complexity
- Computation through abstraction vs. direct computation
- Accuracy (over-approximation, convergent-approximation)
- Other factors…

Computational tools



This lecture focuses on theory and tools for

- General, nonlinear dynamics
- General, nonlinear sets
- Bounded continuous controlled and disturbance inputs
- Controller synthesis / reachable set synthesis
- Continuous or hybrid dynamics
- Level Set Toolbox which provides a convergent-approximation through direct computation

General method from

- Tomlin, Lygeros, Sastry (TAC 2000)
- Mitchell, Bayen, Tomlin (TAC 2004)

Many other methods and tools are available

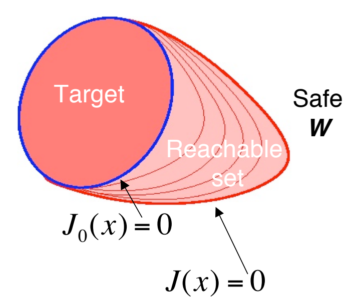
- UBC's verification group (CS)

Today's lecture

- Background
  - Verification through reachability
  - Literature and tools survey
- Preliminaries
  - **Continuous reachability**
  - Discrete reachability
- Hybrid reachability algorithm
- Examples

Continuous reachability

$$\dot{x} = f(x,u), u \in U$$



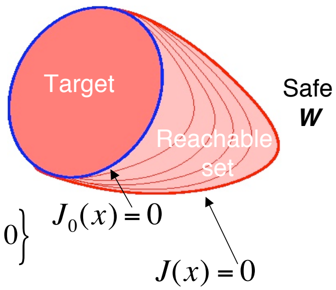
- Goal: Find those states for which there exists a control law that will keep the state away from the target
- With a controlled input and no disturbance input, this is an optimal control problem
- Solve Hamilton-Jacobi-Isaacs equation to find **backwards reachable set**
- Implicitly define target through sub-level sets of J(x)

Continuous reachability

- Modified time-dependent terminal value Hamilton-Jacobi equation

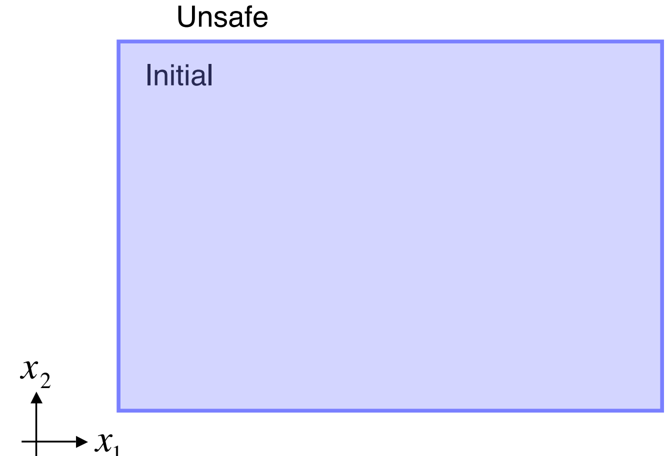
$$\dot{x} = f(x,u), u \in U$$

$$J(x,0) = J_0(x)$$
$$H(x,p) = \max_{u \in U} p^T f(x,u)$$
$$-\frac{\partial J(x,t)}{\partial t} = \min\left\{0, H\left(x, \frac{\partial J(x,t)}{\partial x}\right)\right\}$$
$$W(t) = \{x \in X : J(x,t) \geq 0\}$$
$$u^*(x) = \left\{u \in U : \left(\tfrac{\partial J(x)}{\partial x}\right)^T f(x,u) \geq 0\right\}$$
for $x \in \partial W$

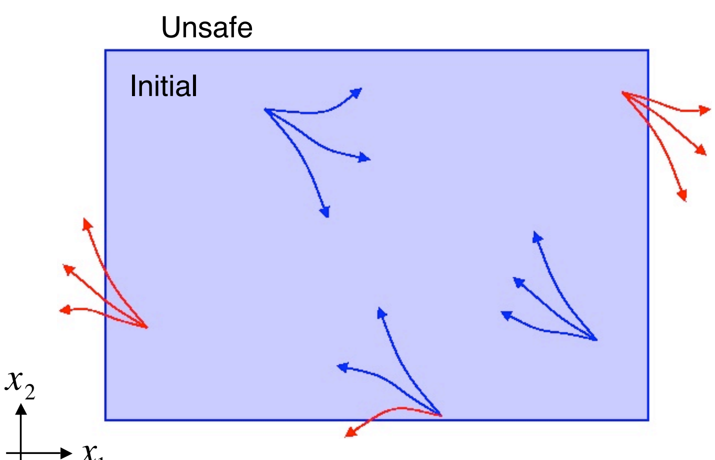


- Chooses control input closest to tangent of boundary of zero-level set

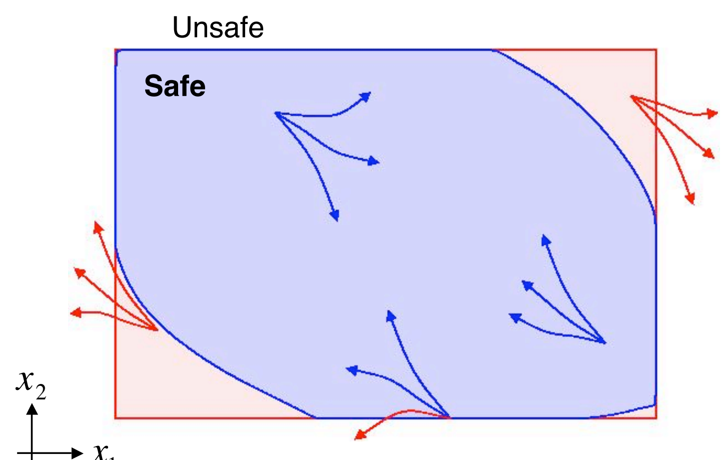
Example: Double integrator



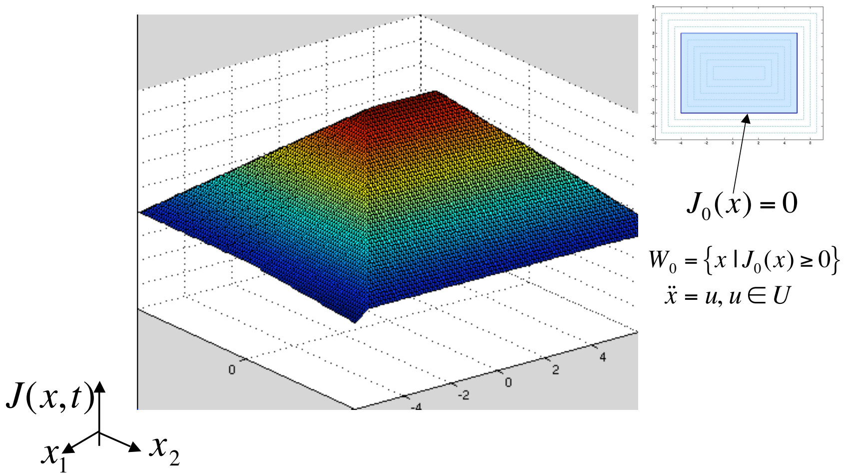
Example: Double integrator



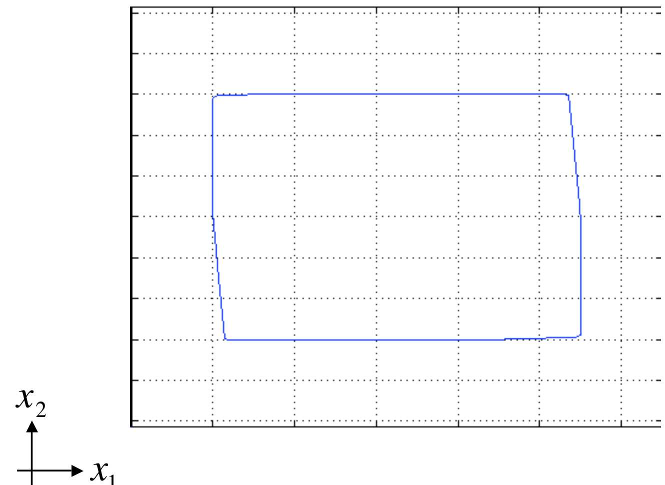
Example: Double integrator



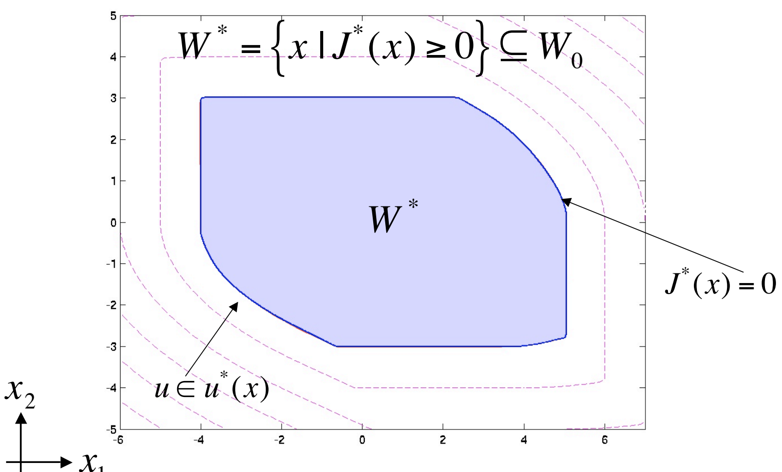
Example: Double integrator



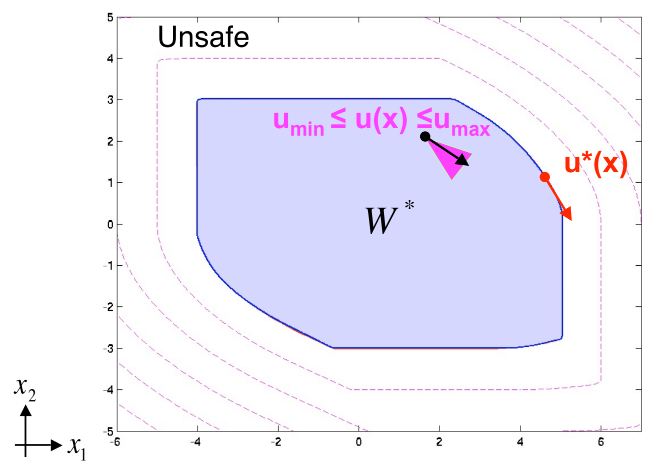
Example: Double integrator



Example: Double integrator



Example: Double integrator



Continuous reachability

- Extension to systems with both controlled and disturbance inputs

$$J(x,0) = J_0(x)$$

$$H(x,p) = \max_{u\in U}\min_{d\in D} p^T f(x,u,d)$$

$$-\frac{\partial J(x,t)}{\partial t} = \min\left\{0, H\left(x, \frac{\partial J(x,t)}{\partial x}\right)\right\}$$

$$W(t) = \{x \in X : J(x,t) \geq 0\}$$

$$u^*(x) = \left\{u \in U : \left(\tfrac{\partial J(x)}{\partial x}\right)^T f(x,u,d) \geq 0\right\}$$

for $x \in \partial W$, and any $d \in D$

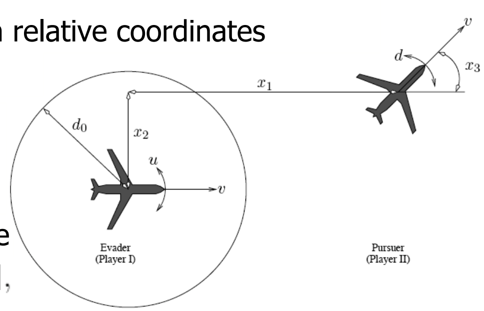
- This is a differential game.

Continuous reachability

- The solution to the HJI is the "viscosity solution"
- This is equivalent to the solution to the minimal-time-to-reach problem:
  - Find the bounded disturbance input *d* in *D* which drives the state of the system to the target in minimal time
- (See Mitchell, Bayen, Tomlin TAC 2004)
- Viability tools have been developed to compute this solution (Aubin, St. Pierre, Cruck, and others)

Example: Collision avoidance

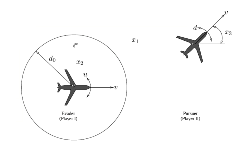
- Evader-pursuer in relative coordinates
- Identical vehicles
- 5nmi separation constraint
  $\sqrt{x_1^2 + x_2^2} \leq d_0$
- Bounded turn rate
  $u \in \mathcal{U} = [-1, +1],$
  $d \in \mathcal{D} = [-1, +1]$



$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -v + v\cos x_3 + ux_2 \\ v \sin x_3 - ux_1 \\ d - u \end{bmatrix} = f(x,u,d)$$

Example: Collision avoidance

- Define the initial "target" set



$$\phi_0(x) = \sqrt{x_1^2 + x_2^2} - d_0,$$
$$\mathcal{G}_0 = \left\{x \in \mathbb{R}^3 | \phi_0(x) \leq 0\right\}$$

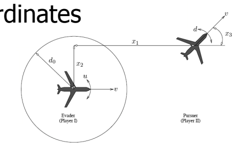
- Evolve this set backwards in time according to the relative coordinate-frame dynamics to find the backwards reachable set

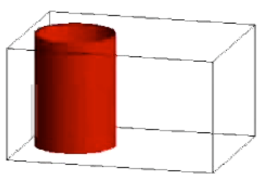
$$\mathcal{G}(\tau) = \left\{x \in \mathbb{R}^3 | \phi(x, -\tau) \leq 0\right\}$$

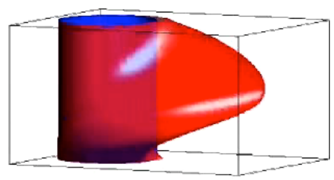
- This is the set of states for which NO control input exists (the evader's turn rate) that will keep a distance of at least d_0 between the two aircraft for τ seconds.
- The complement is the 'maximal controlled invariant set'

Example: Collision avoidance

- The initial set is a cylinder in relative coordinates
- The backwards reachable set has the largest cross-section in (x_1,x_2) for $x_3 = \pi$
- Animations courtesy Prof. Ian Mitchell, www.cs.ubc.ca/~mitchell





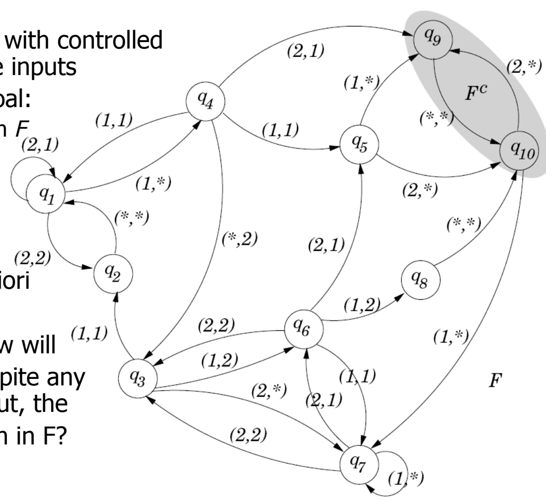


Today's lecture

- Background
  - Verification through reachability
  - Literature and tools survey
- Preliminaries
  - **Discrete reachability**
  - Continuous reachability
- Hybrid reachability algorithm
- Examples

Discrete reachability

- Consider a DES with controlled and disturbance inputs
- Control input goal:
  - Always stay in F
- Disturbance input goal:
  - Drive state out of F
- F is known a priori
- F^c is "target"
- What control law will ensure that despite any disturbance input, the state will remain in F?



Discrete reachability

- Discrete reachability algorithm

$$
\begin{aligned}
&\textbf{initialization: } W^0 = F,\ W^{-1} = \emptyset,\ i = 0. \\
&\textbf{while } W^i \neq W^{i-1} \textbf{ do} \\
&\quad W^{i-1} = W^i \cap \{q \in Q \mid \exists \sigma_1 \in \Sigma_1\ \forall \sigma_2 \in \Sigma_2 \\
&\qquad\qquad\qquad R(q,\sigma_1,\sigma_2) \subseteq W^i\ \} \\
&\quad i = i - 1 \\
&\textbf{end while}
\end{aligned}
$$

- Where Σ_1 is the set of controlled inputs and Σ_2 is the set of disturbance inputs

Discrete reachability

- Can be formulated as a discrete game
- Create the cost function at iteration i

$$J(q,i) = \begin{cases} 1 & q \in W^i \\ 0 & q \in (W^i)^c \end{cases}$$

- And evolving backwards in time according to the discrete transition function q' = R(q, σ_1, σ_2)

$$\max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} \min_{q' \in R(q,\sigma_1,\sigma_2)} J(q',i)$$

$$= \begin{cases} 1 & \text{if } \exists \sigma_1 \in \Sigma_1 \, \forall \sigma_2 \in \Sigma_2, R(q,\sigma_1,\sigma_2) \subseteq W^i \\ 0 & \text{otherwise} \end{cases}$$

$$J(q,i-1) - J(q,i) = \min\{0, \max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} [\min_{q' \in R(q,\sigma_1,\sigma_2)} J(q',i) - J(q,i)]\}$$

Discrete reachability

- The solution to this game is the set of "winning states" W*
- This is the largest set of states for which there exists a control input which, if enforced, will keep the state in F.

$$W^* = \{q \in Q \mid J^*(q) = 1\}.$$

Today's lecture

- Background
  - Verification through reachability
  - Literature and tools survey
- Preliminaries
  - Continuous reachability
  - Discrete reachability
- **Hybrid reachability algorithm**
- Examples

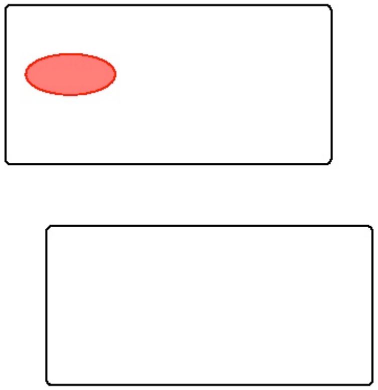
Hybrid reachability

- Consider the hybrid system with hybrid target set

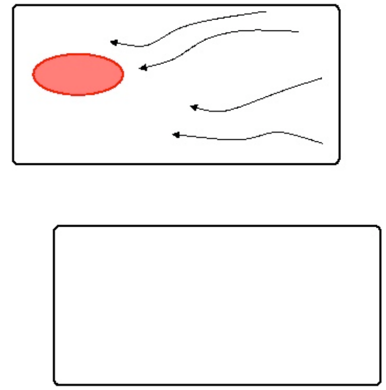
$$\mathcal{G}_0 = \{(q,x) \in Q \times \mathbb{R}^n | g(q,x) \leq 0\}$$

- Controlled discrete and continuous inputs
  (try to keep the state *away from* the target)
- Disturbance discrete and continuous inputs
  (try to steer the state *into* the target)
- Goal: Find the largest set of states for which there exists controlled inputs that can keep the state away from the target, despite disturbance inputs

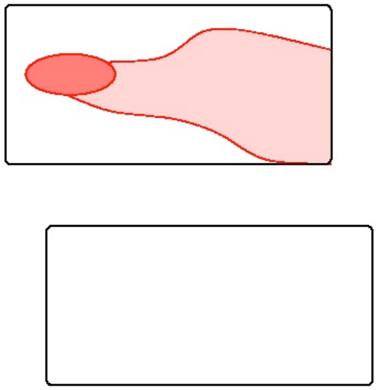
Hybrid reachability



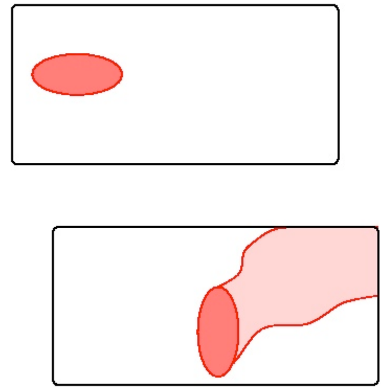
Hybrid reachability



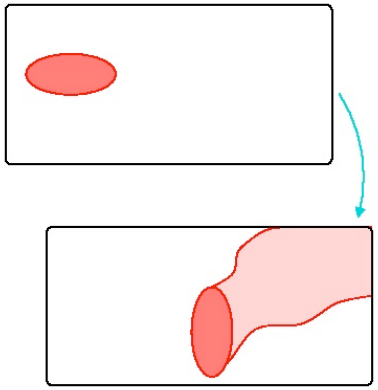
Hybrid reachability



Hybrid reachability

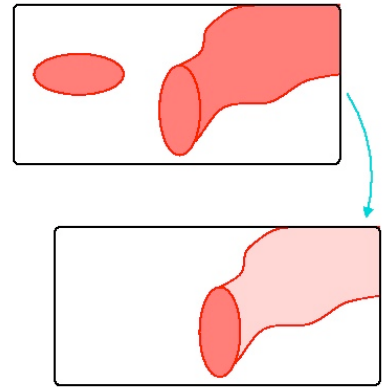


Hybrid reachability



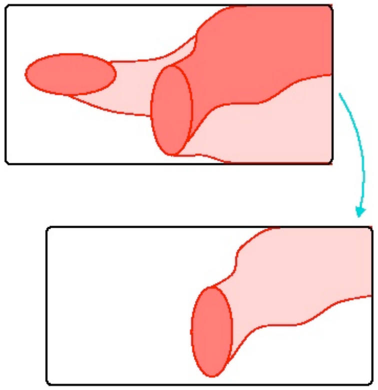
EECE 571M / 491M Winter 2007

Hybrid reachability



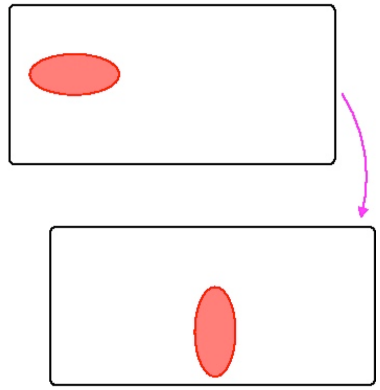
EECE 571M / 491M Winter 2007

Hybrid reachability



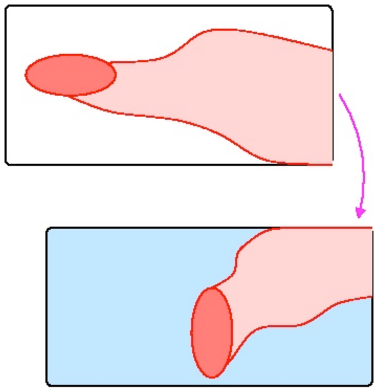
EECE 571M / 491M Winter 2007

Hybrid reachability

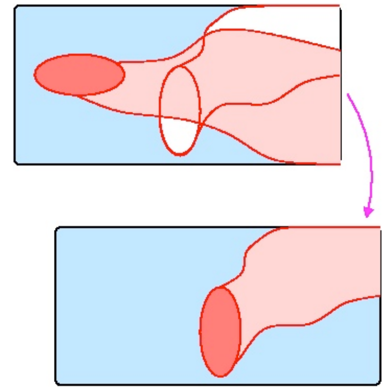


EECE 571M / 491M Winter 2007

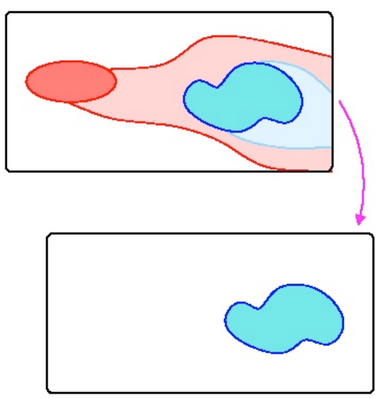
Hybrid reachability



Hybrid reachability



Hybrid reachability



Hybrid reachability

- **Controllable predecessor**

  $$\mathrm{Pre}_u(K) = \{(q,x) \in K : \exists(\sigma_u, u) \in \Sigma_{\mathbf{u}} \times \mathcal{U}\ \forall(\sigma_d, d) \in \Sigma_{\mathbf{d}} \times \mathcal{D}\ R(q,x,\sigma_u,\sigma_d,u,d) \subseteq K\}$$

  Those states for which there exists a control (discrete or continuous) that will keep the state in K for one iteration

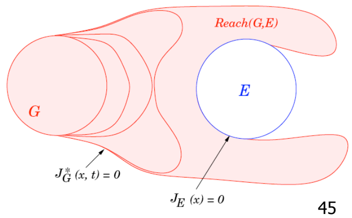
- **Uncontrollable predecessor**

  $$\mathrm{Pre}_d(K^c) = \{(q,x) \in K : \forall(\sigma_u, u) \in \Sigma_{\mathbf{u}} \times \mathcal{U}\ \exists(\sigma_d, d) \in \Sigma_{\mathbf{d}} \times \mathcal{D}\ R(q,x,\sigma_u,\sigma_d,u,d) \cap K^c \neq \emptyset\} \cup K^c$$

  Those states for which no control exists that will prevent the state from being driven to K^c in one iteration (and those states already in K^c)

Hybrid reachability

- **Reach-Avoid operator**
  Given two subsets
  $$G \subseteq Q \times \mathbb{R}^n \text{ and } E \subseteq Q \times \mathbb{R}^n \text{ such that } G \cap E = \emptyset.$$
  where *G* is the target and *E* is the "escape set", define the operator
  $$\text{Reach}(G, E) = \{(q,x) \in Q \times \mathbb{R}^n \mid \forall u \in \mathcal{U}\ \exists d \in \mathcal{D} \text{ and } t \geq 0 \text{ such that } (q, x(t)) \in G \text{ and } (q, x(s)) \in \text{Dom} \setminus E \text{ for } s \in [0,t]\}$$
  as those states which will inevitably be driven to the target *G* without first reaching the escape set *E.*



Hybrid reachability

- **Hybrid reachability algorithm**

Initialization:
$W^0 = F,\ W^1 = \emptyset,\ i = 0$
while $W^i \neq W^{i+1}$ **do**
begin
$W^{i-1} = W^i \setminus \text{Reach}(\text{Pre}_2(W^i), \text{Pre}_1(W^i))$
$i = i - 1$
end

- In the first step, remove from F those states for which the disturbance (discrete or continuous) can force the state to leave F, while also preventing the state from entering the set of states for which there exists a control action to keep the system inside F.

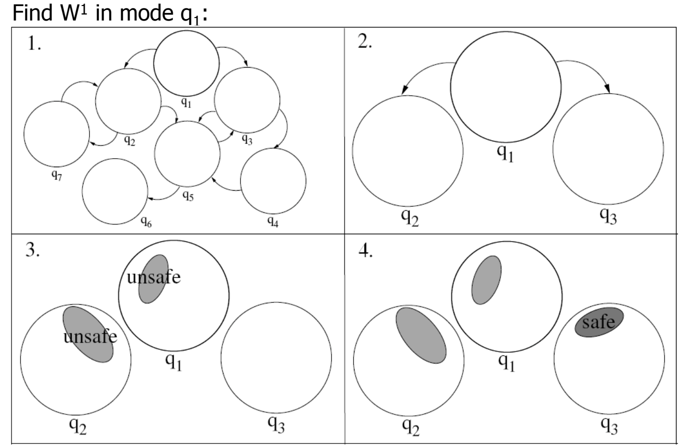
Hybrid reachability

- *Reach* is a computation on the continuous evolution of the state, done independently in each mode
- The *Reach* computations for each mode can be done in parallel
- To solve the *Reach* computation, define *G* and *E* implicitly:
  $$G(t) = \{x \in X : J_G(x,t) \leq 0\}$$
  $$E = \{x \in X : J_E(x) \leq 0\}$$
- Modify the HJ equation
  $$\frac{\partial J_G(x,t)}{\partial t} + \min(0, H(x, \frac{\partial J_G(x,t)}{\partial x})) = 0$$
  $$\text{subject to } J_G(x,t) \geq J_E(x)$$

so that the evolution of $J_G(x,t)$ is frozen once trajectories enter *E*.

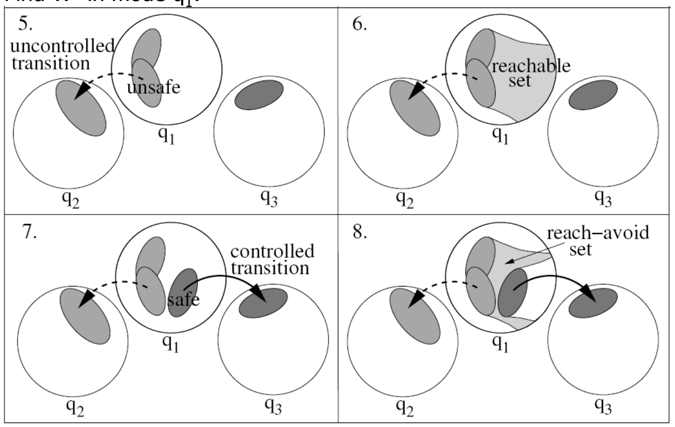
Hybrid reachability

- Find W^1 in mode q_1:



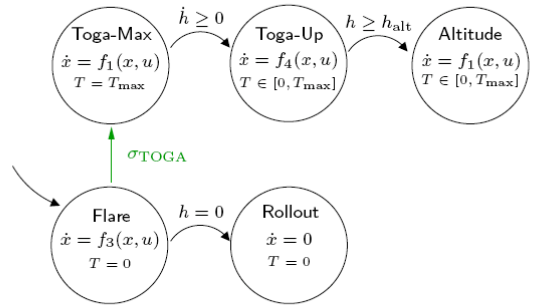
Hybrid reachability

- Find W^1 in mode q_1:



Hybrid reachability

Automatic landing/go-around example

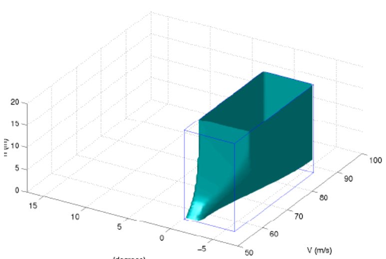


Mode	V [m/s]	γ [degrees]	α [degrees]
Flare	[55.57, 87.46]	[−6.0°, 0.0°]	[−9°, 15°]
Toga-Max	[63.79, 97.74]	[−6.0°, 0.0°]	[−8°, 12°]
Toga-Up	[63.79, 97.74]	[0.0°, 13.3°]	[−8°, 12°]
Altitude	[63.79, 97.74]	[−0.7°, 0.7°]	[−8°, 12°]

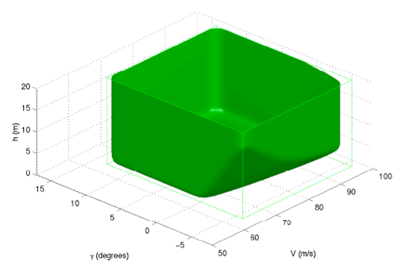
Hybrid reachability

Automatic landing/go-around example

Safe region for landing



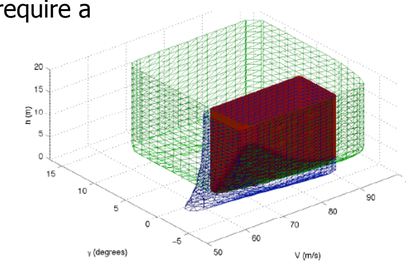
Safe region for go-around



Hybrid reachability

Automatic landing/go-around example

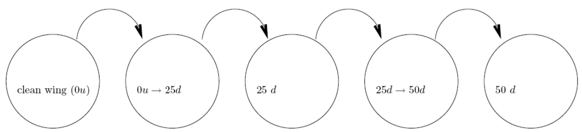
- Intersection of 'safe landing' and 'safe go-around' sets
- The type of event which triggers a go-around will change the shape of these sets
- A disturbance event will require a reachability computation on the red region under landing dynamics
- Note that the red region does not intersect h=0!



Hybrid reachability

Automatic landing (flap control) example

- Flaps are ‘deflected’ or ‘undeflected’
- 0 degrees, 25 degrees, or 50 degrees



- Transitions are assumed to be controlled events
- Time-delay (10 seconds) for flaps to move to new position

Hybrid reachability

Automatic landing (flap control) example

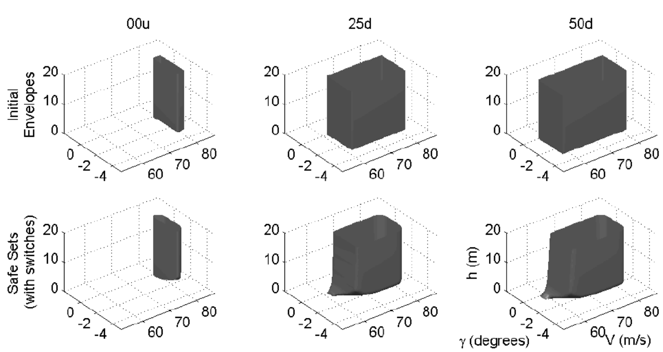
- State-space constraints which define the initial set W^0:

$$\left\{\begin{array}{ll} h \leq 0 & \text{landing or has landed} \\ V > V_\delta^{\text{stall}} & \text{faster than stall speed} \\ V < V^{\max} & \text{slower than limit speed} \\ V \sin\gamma \geq \dot{z}_0 & \text{limited TD speed} \\ \gamma \leq 0 & \text{monotonic descent} \end{array}\right. \cup \left\{\begin{array}{ll} h > 0 & \text{aircraft in the air} \\ V > V_\delta^{\text{stall}} & \text{faster than stall speed} \\ V < V^{\max} & \text{slower than limit speed} \\ \gamma > -3^\circ & \text{limited descent flight path} \\ \gamma \leq 0 & \text{monotonic descent} \end{array}\right.$$

- Nonlinear flight dynamics in (V, γ, z) -- speed, flight path angle, and altitude -- for a DC9-30

Hybrid reachability

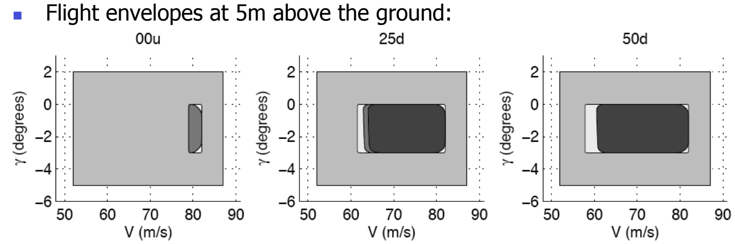
Automatic landing (flap control) example



Hybrid reachability

Automatic landing (flap control) example

- Flight envelopes at 5m above the ground:



- Dark grey: subset of the initial escape set that is also safe in current mode
- Mid-grey: initial escape set
- Light grey: Target (Complement of W0)
- White: Reach set (States from which the system cannot remain in current mode and cannot switch to safety)

Summary

- Continuous reachability
  - Level set methods
  - Hamilton-Jacobi formulation
- Discrete reachability
  - Invariant set algorithm
- Hybrid reachability
  - Reach-Avoid operator
  - Invariant set algorithm
- Examples