Eulerian Approaches to Backward Reachability in Continuous and Hybrid Systems

Ian Mitchell
Department of Computer Science
The University of British Columbia

Joint work with
Alex Bayen, Meeko Oishi & Claire Tomlin (Stanford)
Jon Sprinkle, Mike Eklund & Shankar Sastry (Berkeley)

research supported by
National Science and Engineering Research Council of Canada
Office of Naval Research
DARPA Software Enabled Control Project
Continuous Reach Sets
and the Hamilton-Jacobi Equation

An Eulerian
Dynamic Implicit Surface Framework
Outline

• Reach sets
  – Backwards vs forward, sets vs tubes, maximal vs minimal
  – Implicit surface representation of sets
  – Game of two identical vehicles

• Computing reach sets
  – Formulation as a time-dependent HJ PDE
  – The Toolbox of Level Set Methods

• The mixed implicit explicit formulation of reach sets
  – Dynamics with terminal integrators
  – Hamilton-Jacobi / optimal control formulation of terminal integrator’s reach set / tube
  – Examples: various double integrators and the pursuit of the oblivious evader

• Decoupling & other tricks
  – Example: Glideslope recapture
Typical Application: Safety Analysis

• Does there exist a trajectory of system $H$ leading from a state in initial set $I$ to a state in terminal set $T$ (under some policy for input $u(\cdot)$)?

• Assumption: given $z$, $t$ and $u(\cdot)$ the trajectory is unique

• Applicable to general dynamic systems: ODEs, OΔEs, discrete systems, hybrid systems, etc.

- $s \in \mathbb{T}$ is current time
- $z \in \mathbb{Z}$ is initial state
- $t \in \mathbb{T}$ is initial time
- $u(\cdot) \in \mathbb{U}$ is input signal
Nondeterministic, Nonlinear Systems

\[ \dot{x} = f(x, p) \]

- Systems with unknown parameters \( p(t) \)
- Bounded value inputs \( p(t) \in P \)
  - Controls: double integrator time to reach
  - Disturbances: robust reach sets
- Stochastic perturbations \( p(t) \sim P \)
  - Continuous state Brownian motion: double integrator with stochastic viscosity
  - Discrete state Poisson processes: stochastic hybrid system model of TCP communication protocol
Forward Reachability

- Start at initial conditions and compute forward
Forward Reachability Algorithms

• Forward approach typical of Lagrangian algorithms
  – Representation moves with the underlying dynamics
  – Fixed representation of sets
  – Varying ability to handle nonlinearity and/or inputs
  – Demonstrated ability to handle high dimensions

• Examples
  – [Henzinger, Ho & Wong-Toi, IEEE TAC 1998]
  – [Greenstreet & Mitchell, HSCC 1999]
  – [Bemporad, Torrisi & Morari HSCC 2000]
  – [Kurzhanski & Varaiya, HSCC 2000]
  – [Asarin, Dang & Girard, HSCC 2003]
  – [Girard, Guernic & Maler, HSCC 2006]
  – [Han & Krogh, HSCC 2006]
Backward Reachability

- Start at terminal set and compute backwards

Backward Reach Set
\[ B(H, T, t) \]

Backward Reach Tube
\[ B(H, T, [0, t]) \]

Terminal Set \( T \)
Backward Reachability Algorithms

• Backward approach typical of Eulerian algorithms
  – Representation not moving (although it may adapt)
  – More general set representation
  – Generally handle nonlinear and multiple inputs
  – No examples beyond four dimensions?

• Examples
  – [Broucke, Benedetto, Gennaro & Sangiovanni-Vincentelli, HSCC 2001]
  – [Saint-Pierre, HSCC 2002]
  – [Sethian & Vladimirsky, HSCC 2002]
  – [Gao, Lygeros & Quincampoix, HSCC 2006]
Dealing with Inputs

- Input $u(\cdot)$ can maximize or minimize the set or tube
  - Interpretation as best or worst case depends on context

Backward maximal reach set
(exists a control)

Backward minimal reach tube
(for all controls)
Determining Reach Sets

- Three primary challenges
  - How to represent set of reachable states
  - How to evolve set according to dynamics
  - How to compare sets of reachable states
- 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 $S(t)$ 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

\[
\phi : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}
\]

\[
S(t) = \{ x \in \mathbb{R}^n \mid \phi(x, t) \leq 0 \}
\]
Game of Two Identical Vehicles

- Classical collision avoidance example
  - Collision 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 $v_e = v_p = 5$

\[ \begin{aligned}
\frac{d}{dt} \begin{bmatrix} x_p \\ y_p \\ \theta_p \end{bmatrix} &= \begin{bmatrix} v_p \cos \theta_p \\ v_p \sin \theta_p \\ b \end{bmatrix}
\end{aligned} \]

evader aircraft (control)  pursuer aircraft (disturbance)
Collision Avoidance Computation

- Use relative coordinates with evader fixed at origin
  - State variables are now relative planar location \((x,y)\) and relative heading \(\psi\)

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

target set description
\[ h(x) = \sqrt{x^2 + y^2} - 5 \]

evader aircraft (control)   pursuer aircraft (disturbance)
Outline

- Reach sets
  - Backwards vs forward, sets vs tubes, maximal vs minimal
  - Implicit surface representation of sets
  - Game of two identical vehicles

- Computing reach sets
  - Formulation as a time-dependent HJ PDE
  - The Toolbox of Level Set Methods

- The mixed implicit explicit formulation of reach sets
  - Dynamics with terminal integrators
  - Hamilton-Jacobi / optimal control formulation of terminal integrator’s reach set / tube
  - Examples: various double integrators and the pursuit of the oblivious evader

- Decoupling & other tricks
  - Example: Glideslope recapture
Evolving Reachable Sets

- Modified Hamilton-Jacobi partial differential equation

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

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

and terminal conditions: \(\phi(x, 0) = h(x)\)

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

and \(\dot{x} = f(x, a, b)\)
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]
Alternative Eulerian Approaches

• Static Hamilton-Jacobi (Falcone, Sethian, …)
  – Minimum time to reach
  – (Dis)continuous implicit representation
  – Solution provides information on optimal input choices

• Viability kernels (Aubin, Saint-Pierre, …)
  – Based on set valued analysis for very general dynamics
  – Discrete implicit representation
  – Overapproximation guarantee

• Time-dependent Hamilton-Jacobi (this method)
  – Continuous solution
  – Information on optimal input choices available throughout entire state space
  – High order accurate approximations

• All three are theoretically equivalent
Time-Dependent Hamilton-Jacobi Eq’n

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

• First order hyperbolic PDE
  – Solution can form kinks (discontinuous derivatives)
  – For the backwards reachable set, find the “viscosity” solution [Crandall, Evans, Lions, …]

• Level set methods
  – Convergent numerical algorithms to compute the viscosity solution [Osher, Sethian, …]
  – Non-oscillatory, high accuracy spatial derivative approximation
  – Stable, consistent numerical Hamiltonian
  – Variation diminishing, high order, explicit time integration
Level Set Methods

• Numerical algorithms for dynamic implicit surfaces and time dependent Hamilton-Jacobi partial differential equations

• Applications in
  – Graphics, Computational Geometry and Mesh Generation
  – Financial Mathematics and Stochastic Differential Equations
  – Robotics, Control and Dynamic Programming
  – Fluid and Combustion Simulation
  – Image Processing and Computer Vision
The Toolbox of Level Set Methods

• A collection of Matlab routines to approximate the viscosity solution of time-dependent HJ PDEs
  – Fixed Cartesian grids
  – Arbitrary dimension (computational resource limited)
  – Vectorized code achieves reasonable speed
  – Direct access to Matlab debugging and visualization
  – Source code is provided for all toolbox routines

• Underlying algorithms
  – Solve various forms of Hamilton-Jacobi PDE
  – First and second spatial derivatives
  – First temporal derivatives
  – High order accurate approximation schemes
  – Explicit temporal integration
Using the Toolbox

• Similar to Matlab’s ODE integrators
  – More parameters to specify
  – Formulation and scaling must be considered
  – Many examples are available

• PDE forms applicable to systems analysis
  – [Mitchell & Templeton, HSCC 2005]
    \[0 = D_t \varphi(x, t) + v(x, t) \cdot \nabla \varphi(x, t) + H(x, t, \varphi, \nabla \varphi) - \text{trace}[L(x, t)D^2_x \varphi(x, t)R(x, t)] + \lambda(x, t)\varphi(x, t) + F(x, t, \varphi),\]
    \[D_t \varphi(x, t) \geq 0, \quad D_t \varphi(x, t) \leq 0,\]
    \[\varphi(x, t) \leq \psi(x, t), \quad \varphi(x, t) \geq \psi(x, t),\]
Example: Continuous Reachable Sets

- Game of two identical vehicles
- Nonlinear dynamics with adversarial inputs

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

\[ H(x, p) = \max_{a \in \mathcal{A}} \min_{b \in \mathcal{B}} [p \cdot f(x, a, b)] \]

\[
\begin{bmatrix}
\frac{dx_1}{dt} \\
\frac{dx_2}{dt} \\
\frac{dx_3}{dt}
\end{bmatrix} =
\begin{bmatrix}
-v_a + v_b \cos x_3 + ax_2 \\
v_b \sin x_3 - ax_1 \\
 b - a
\end{bmatrix}
\]

\[
= f(x, a, b)
\]

\[ a \in \mathcal{A} = [-1, +1] \]

\[ b \in \mathcal{B} = [-1, +1] \]

\[ v_a, v_b \text{ constant} \]
Example: Continuous Reachable Sets

- Two vehicle collision avoidance
- Nonlinear dynamics with cooperative inputs

\[ D_t \phi(x, t) + \min \left[ 0, H(x, \nabla \phi(x, t)) \right] = 0 \]

\[ H(x, p) = \max_{a \in A} \min_{b \in B} [p \cdot f(x, a, b)] \]

\[
\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -v_a + v_b \cos x_3 + ax_2 \\ v_b \sin x_3 - ax_1 \\ b - a \end{bmatrix} = f(x, a, b) 
\]

\[ a \in A = [-1, +1] \]

\[ b \in B = [-1, +1] \]

\[ v_a, v_b \text{ constant} \]
A Different Continuous Reachable Set

  - Variation on homicidal chauffeur, where evader must reduce speed when near pursuer

\[
\frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = W_p \begin{bmatrix} 0 \\ -1 \end{bmatrix} + \frac{W_p}{R} \begin{bmatrix} y \\ -x \end{bmatrix} b \\
+ 2W_e \min \left( \sqrt{x^2 + y^2}, S \right) a
\]

\[
a \in \mathbb{R}^2, \|a\| \leq 1
\]

\[
b \in [-1, +1]
\]

\[
W_p, W_e, R, S \text{ constant}
\]
Implemented in the Toolbox

• Part of the standard toolbox distribution (version 1.0)
  – Examples/Reachability/

• PDE terms utilized

\[ 0 = D_t \varphi(x, t) + v(x, t) \cdot \nabla \varphi(x, t) + H(x, t, \varphi, \nabla \varphi) \]

\[ - \text{trace}[L(x, t)D_x^2 \varphi(x, t)R(x, t)] + \lambda(x, t)\varphi(x, t) + F(x, t, \varphi), \]

\[ D_t \varphi(x, t) \geq 0, \quad D_t \varphi(x, t) \leq 0, \]

\[ \varphi(x, t) \leq \psi(x, t), \quad \varphi(x, t) \geq \psi(x, t), \]
The Toolbox: How to Use It

- Cut and paste from existing examples
- Most code is for initialization and visualization

- Cartesian grid: dimension, range, cell size
- Boundary conditions (periodic, Dirichlet, Neumann, extrapolated)
- Initial conditions
- Basic shapes (sphere, cylinder, hyperplane)
- Constructive solid geometry (union, intersection, complement)
- Integrators (explicit TVD RK order 1–3)
- CFL number, post-timestep processing (e.g., masking)
- Term approximations (convection, mean curvature, general HJ, forcing, sum, etc.)
- Term parameters (dissipation, speed, flow field, etc.)
- Spatial derivatives
- Upwinded first (minmod / ENO / WENO order 1–5)
- Centered second (curvature & Hessian order 2)
- Visualization (contour, isosurface, etc.)

User supplied  Toolbox supplied  Matlab supplied

Ian Mitchell (UBC Computer Science)
The Toolbox is not a Tutorial

• Users will need to read the literature
• Two textbooks are available
  – Osher & Fedkiw (2002)
  – Sethian (1999)
Why Use It?

• Does not escape Bellman’s curse of dimensionality
  – Dimensions 1–3 interactively, 3–5 slow but feasible

• Pedagogical tool
  – Experiment with optimal control and differential game problems that have no analytic solution
  – Access to Matlab’s visualization & debugging
  – Source code for all routines and examples
  – Reasonable speed with vectorized code

• Validation of faster but more specialized algorithms
  – For example, reduced order TCP model assumed form of high order moments of the distribution [HSCC 2005]

• Study low dimensional systems
  – Mobile robots in 2–3 spatial dimensions

• Free (google “toolbox level set methods”)
The Goal: Reproducible Research

“[a]n article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.”

[Jon Claerbout, as quoted by Buckheit & Donoho, 1995]

• Reproducible research: scientific results should be independently replicable, given appropriate resources
• Key challenges in computational science
  – Appropriate licensing standards
  – Disciplined software development processes
  – Software infrastructure and tools
• More details
  – Reproducible research workshop, July 13–16, 2011 at UBC in Vancouver (just before ICIAM 2011)
Outline

• Reach sets
  – Backwards vs forward, sets vs tubes, maximal vs minimal
  – Implicit surface representation of sets
  – Game of two identical vehicles

• Computing reach sets
  – Formulation as a time-dependent HJ PDE
  – The Toolbox of Level Set Methods

• The mixed implicit explicit formulation of reach sets
  – Dynamics with terminal integrators
  – Hamilton-Jacobi / optimal control formulation of terminal integrator’s reach set / tube
  – Examples: various double integrators and the pursuit of the oblivious evader

• Decoupling & other tricks
  – Example: Glideslope recapture
Systems with Terminal Integrators

- Common form of system dynamics
  \[ \dot{y} = f(y, u) \text{ coupled states } y \in \mathbb{R}^{d_y}, \]
  \[ \dot{x}_i = b(y) \text{ terminal integrator } x_i \in \mathbb{R} \]
  \[ \text{for } i = 1, \ldots, d_x \]

- Computational cost of reachability for full system with \( n \) grid points is \( \mathcal{O}(n^{(d_y + d_x)}) \)

- Instead
  - Run two modified HJ PDEs on \( \mathbb{R}^{d_y} \) for each of the \( x_i \) variables
  - States are inside overall reach set only if inside every PDE’s reach set
  - Computational cost \( \mathcal{O}(2d_x n^{d_y}) \)
Mixed Implicit Explicit Formulation

- Traditional *implicit* formulation represents sets with an implicit surface function

\[ S = \{(x, y) \mid \psi(x, y) \leq 0\} \]

- New *mixed implicit explicit* (MIE) formulation represents sets as an interval in \( x_i \) for every \( i \) and \( y \)

\[ S = \left\{(x, y) \mid \underline{\psi}_i(y) \leq x_i \leq \overline{\psi}_i(y)\right\} \]

upper bound \( x \leq \overline{\psi}_0(y) \)  lower bound \( x \leq \underline{\psi}_0(y) \) overall target set
Terminal Integrator’s HJ PDEs

- For scalar terminal integrator $d_x = 1$ define target set

$$S = \left\{ (x, y) \mid \psi_0(y) \leq x \leq \overline{\psi}_0(y) \right\}$$

- If $x(t, y) = \overline{\psi}(t, y)$ is the upper boundary of the reach set, then formally

$$b(y) = \frac{d}{dt} x(t, y) = \frac{d}{dt} \overline{\psi}(t, y) = D_t \overline{\psi}(t, y) + D_y \overline{\psi}(t, y) \cdot f(y, u)$$

- Rearrange to find terminal value HJ PDE

$$D_t \overline{\psi}(t, y) + H \left( t, y, D_x \overline{\psi}(t, y) \right) = 0 \quad \overline{\psi}(0, y) = \overline{\psi}_0(y)$$

with $H(t, y, p) = \max_{u \in U} (p \cdot f(y, u) - b(y))$

- Repeat with $x(t, y) = \psi(t, y)$ for lower boundary (with adjustment of optimizations)

- Yields backwards reach set

$$B(S, t) = \left\{ (x, y) \mid \psi(t, y) \leq x \leq \overline{\psi}(t, y) \right\}$$
Double Integrator

- Dynamics

\[ \dot{y} = f(y, u) = u \quad \dot{x} = y \quad |u| \leq u_{\text{max}} \]

yields terminal integrator Hamiltonian (for upper bound)

\[ H(t, y, r, p) = \max_{|u| \leq u_{\text{max}}} (p \cdot u - y) = (|p| u_{\text{max}} - y) \]

Regular implicit surface formulation
One HJ PDE in 2D

Terminal integrator formulation
Two HJ PDEs in 1D
Finite Horizon Optimal Control

- Terminal integrator’s dynamics (for $t < 0$) are

\[ x(0, y(0)) = x(t, y(t)) + \int_t^0 b(y(s)) ds \]

or

\[ x(t, y(t)) = \int_t^0 -b(y(s)) ds + x(0, y(0)) \]

- Can be interpreted as a finite horizon optimal control problem with associated HJ PDE

\[ D_t \overline{\psi}(t, y) + H \left(t, y, D_x \overline{\psi}(t, y)\right) = 0 \quad \overline{\psi}(0, y) = \overline{\psi}_0(y) \]

with \( H(t, y, p) = \max_{u \in U} (p \cdot f(y, u) - b(y)) \)

- Solution \( \overline{\psi}(t, y) \) provides smallest \( x(t, y(t)) \) giving rise to a trajectory which reaches the upper boundary \( x(0, y(0)) = \overline{\psi}_0(y(0)) \) of the target set at \( t = 0 \)
Rotating Double Integrator

- Let $u \in U = \{ u \in \mathbb{R}^2 \mid \|u\|_2 \leq u_{\text{max}} \}$ and
  \[
  \begin{bmatrix}
    \dot{y}_1 \\
    \dot{y}_2
  \end{bmatrix} = \begin{bmatrix}
    -y_2 \\
    +y_1
  \end{bmatrix} + \mu(\|y\|_2) \begin{bmatrix}
    u_1 \\
    u_2
  \end{bmatrix}, \quad \dot{x} = \|y\|_2
  \]

- Behaves radially like first quadrant of traditional double integrator for $\mu(\alpha) \equiv 1$
- For this experiment, $\mu(\alpha) = 2 \sin(4\pi\alpha)$
Terminal Integrators with Linear Self-Coupling

- Generalization of terminal integrator’s dynamics
  \[ \dot{x} = a(y)x + b(y) \]

- Leads to an HJ PDE
  \[ D_t \overline{\psi}(t, y) + H\left(t, y, \overline{\psi}(t, y), D_x \overline{\psi}(t, y)\right) = 0 \quad \overline{\psi}(0, y) = \overline{\psi}_0(y) \]
  \[
  \text{with } H(t, y, q, p) = \max_{u \in U} (p \cdot f(y, u) - a(y)q - b(y))
  \]
  - HJ PDEs of this form arise in discounted finite horizon optimal control
  - In this interpretation, \( a(y) \) is the discount factor

- Existing viscosity solution theory supports positive constant \( a(y) \equiv a > 0 \)
  - Stable systems will have \( a(y) < 0 \) (for backwards reachability)
  - It may be possible to extend the theory
This Ain’t No Double Integrator

- Theoretically supported but unstable dynamics

\[ \dot{x} = ax + y + v \quad \dot{y} = u \]

with \( a = 1 > 0, \quad |u| \leq u_{\text{max}} = 0.25, \quad |v| \leq v_{\text{max}} = 0.5 \)

eyields Hamiltonian

\[ H(t, y, q, p) = \max_{u,v}(p \cdot u - aq - y - v) = (|p| u_{\text{max}} - aq - y + v_{\text{max}}) \]
MIE for the Collision Avoidance Example?

- No terminal integrators: all variables are coupled for evader input $a \neq 0$
  
  - Also, target set has coupling of $x$ and $y$

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

Target set description:

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

evader aircraft (control)  pursuer aircraft (disturbance)
Pursuit of an Oblivious Evader

• The oblivious evader has no input
• Relative position variables are terminal integrators
  – Decouple target set by using a box
• Allow pursuer to adjust both heading and speed
  – \( \omega_p \in \Omega_p \) and \( a_p \in A_p \)

\[
\begin{aligned}
\text{target set description} & \quad \{ (x_1, x_2) \mid |x_1| \leq 1 \land |x_2| \leq 1 \} \\
\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ \theta \\ v_p \end{bmatrix} & = \begin{bmatrix} -v_e + v_p \cos \theta \\ v_p \sin \theta \\ \omega_p \\ a_p \end{bmatrix}
\end{aligned}
\]

evader aircraft (no input) \qquad \text{pursuer aircraft (disturbance)}
Aerial Refueling Scenario

- [Ding & Tomlin, CDC 2010]
  - Tanker flies straight at constant speed
  - UAV attempts to reach small rectangular refueling zone without crashing into the tanker

\[
\begin{align*}
(x_1, x_2) & \quad \left| x_1 - \hat{x}_1 \right| \leq 1 \\
\wedge |x_2| & \leq 1
\end{align*}
\]

\[
\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ \theta \\ v_p \\ w_p \\ a_p \end{bmatrix} = \begin{bmatrix} -v_e + v_p \cos \theta \\ v_p \sin \theta \\ \omega_p \\ a_p \end{bmatrix}
\]

tanker aircraft (no input) \quad \text{UAV aircraft (control)}
Multiple Terminal Integrators

• Three approaches
  – Solve one HJ PDE in \( d_y + d_x \) dimensions (\( y, x_1, \ldots, x_{d_x} \)) to get a full implicit representation of the reach set / tube
  – Solve one HJ PDE in each of \( d_x \) separate subspaces (\( y, x_i \)) of dimension \( d_y + 1 \) to get implicit representations of the projections of the reach set / tube [Mitchell & Tomlin, JSC 2003]
  – Solve two HJ PDEs for each of \( d_x \) separate dimensions in the subspace (\( y \)) of dimension \( d_y \) to get MIE representation of the projections of the reach set / tube

• Projection-based representations require decoupling of the inputs
  – Independent choice of input in each projection could be pessimistic but sound, or could introduce leaky corners
Pursuit of the Oblivious Evader

- Parameters for this run were $v_e = 1$, $A_p = [ -0.2, +0.2 ]$, $\Omega_p = [ -0.2, +0.2 ]$, $v_p \in [ 1.0, 3.0 ]$, $t_{\text{max}} = 2.0$
- Projections into $(x_1, v_p, \theta)$
Pursuit of the Oblivious Evader

- Parameters for this run were $v_e = 1$, $A_p = [-0.2, +0.2]$, $\Omega_p = [-0.2, +0.2]$, $v_p \in [1.0, 3.0]$, $t_{\text{max}} = 2.0$
- Projections into $(x_2, v_p, \theta)$

Nov 2011
Ian Mitchell (UBC Computer Science)
Aside: Projective Overapproximation

- Overapproximate reachable set of high dimensional system as the intersection of reachable sets for lower dimensional projections
  - [Mitchell & Tomlin, JSC 2003]
  - Example: rotation of “sphere” about z-axis
Example: Projective Collision Avoidance

- Work strictly in relative $x-y$ plane
  - Treat relative heading $\psi \in [0, 2\pi]$ as a disturbance input
  - Compute time: 40 seconds in 2D vs 20 minutes in 3D
  - Compare overapproximative prism (mesh) to true set (solid)
MIE: Pros & Cons

• Slices of reach tube at $v_p = 2$
  – Projection formulations cannot represent true reach tube, must overapproximate

• Inputs calculated separately in each projection
  – Results are pessimistic but sound
  – Not appropriate for refueling scenario

• Computational cost
  – MIE: $65 \times 100$ grid, 3.1 seconds
  – Decoupled implicit: $151 \times 65 \times 100$ grid, 541 seconds
  – Full dimensional implicit: $151^2 \times 65 \times 100$ grid estimated at 30 hours
Comparing Formulations

- Memory and computational cost
  - Full implicit: $\mathcal{O}(n^{(d_y+d_x)})$
  - Decoupled implicit: $\mathcal{O}(d_x n^{(d_y+1)})$
  - Decoupled MIE: $\mathcal{O}(2d_x n^{d_y})$

- Benefits of implicit representations
  - Implicit surface function remains continuous: easier theory, numerics, application of constraints
  - Narrowband or local level set schemes reduce cost
  - Artificial boundary conditions at the edge of the computational domain are dealt with more easily
  - Optimal control may be extracted from gradient of solution

- ToolboxLS implementation provides no guarantee on the sign of the error, but the MIE approach can be used with other implementations
Safely Switching Control Modes

• One application of reach sets is to determine when it is safe to switch between distinct control modes
  – Final mode has region $S_0$ within which final mode’s controller is known to be stable
  – Compute $B(S_0, [0, t_0])$ using dynamics for final mode’s controller to determine region within which switch to final mode is safe
  – Pick $S_1 \subset B(S_0, [0, t_0])$ as target for second to last mode
  – Compute $B(S_1, [0, t_1])$ using dynamics for second to last mode’s controller, and so on
Three Dimensions is Child’s Play

• Simplified longitudinal quadrotor dynamics are six dimensional
  – Assumes that out-of-plane dynamics can be stabilized
  – Analysis performed separately on three position / velocity pairs of variables

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix}
= 
\begin{bmatrix}
-\frac{1}{m} C^v_D \ddot{x} \\
-\frac{1}{m} (mg + C^v_D \dot{y}) \\
-\frac{1}{I_{yy}} C^\theta_D \dot{\theta}
\end{bmatrix}
+ 
\begin{bmatrix}
0 & 0 \\
-\frac{1}{m} \sin \theta & -\frac{1}{m} \sin \theta \\
\frac{1}{m} \cos \theta & \frac{1}{m} \cos \theta
\end{bmatrix}
\begin{bmatrix}
T_1 \\
T_2
\end{bmatrix}
\]
Outline

• Reach sets
  – Backwards vs forward, sets vs tubes, maximal vs minimal
  – Implicit surface representation of sets
  – Game of two identical vehicles

• Computing reach sets
  – Formulation as a time-dependent HJ PDE
  – The Toolbox of Level Set Methods

• The mixed implicit explicit formulation of reach sets
  – Dynamics with terminal integrators
  – Hamilton-Jacobi / optimal control formulation of terminal integrator’s reach set / tube
  – Examples: various double integrators and the pursuit of the oblivious evader

• Decoupling & other tricks
  – Example: Glideslope recapture
Glideslope Recapture

(a) safe-set of operation relative to the desired point of landing on the virtual runway (f)
(b) vector-off maneuver requested
(c) command to land (if possible) is given
(d) aircraft will continue to vector-off (if landing is unsafe) or will issue commands to recapture the glideslope at some point (e)
Simple Model, Use Some Tricks

• Aircraft acts like a kinematic cart in longitudinal and lateral dynamics
  - Separate \((x, y, \theta)\) and \((x, z, \psi)\) calculations
  - Swap integration variable \(t\) to \(x\)

• Reduce 5D state space to two 2D calculations

• Variable resolution grid (two methods)

• [Sprinkle, Eklund & Sastry, *Proc. ACC*, 2005]


\[
\begin{bmatrix}
    x \\
    y \\
    z \\
    \theta \\
    \psi
\end{bmatrix}
= \begin{bmatrix}
    v \cos \psi \cos \theta \\
    v \sin \psi \\
    v \sin \theta \\
    u_\theta \\
    u_\psi
\end{bmatrix}
\]

- \(x\) distance along runway
- \(y\) distance across runway
- \(z\) height
- \(\theta\) yaw
- \(\psi\) roll

\[u_\theta \in U_\theta \quad u_\psi \in U_\psi\]

\[\theta \in [\theta_{\text{min}}, \theta_{\text{max}}] \quad \psi \in [\psi_{\text{min}}, \psi_{\text{max}}]\]
Implementation and Results

- ACC results shown; ISSE results difficult to visualize
- All pieces fit together, grid resolution changes by factor of ten
- ISSE uses nonlinear coordinate transform to avoid the need for separate grids

\[
[x, z, \theta] \text{ projection}
\]

\[
[0,3) \quad \quad [3,10)
\]
Implementation and Results

$[1,3)$ Safe Set from $[1,3)$ nautical miles

$[x, y, \psi]$ projection

$[3,10)$ Safe Set from $[3,10)$ nautical miles

$[0,1)$ Safe Set from $[0,1)$ nautical miles
Implementation

• Use backward reach set to make one lookup table for each projection
  – ~7MB total size
  – Lookup time: ~10ms (~5ms each)
  – Time to generate:
    • ~15 mins for the reach set
    • ~2 hours to compile into an executable (due to compiler issues)

• Additional issues
  – Smooth control laws
  – Decision delay

• Total development time
  – ~2 man months of coding, plus design and research required for safe sets
Demonstration & Results

• Flown on live T-33 aircraft
• Landing on “virtual” runway at a high altitude
• Ground controller gives vector-off and recapture commands
  – 1 successful landing
  – 1 go-around after “unsafe” answer (later verified offline as a correct result)
Hybrid System Reach Sets

Combining Continuous and Discrete Evolution
Outline

• Hybrid System example
  – Seven mode collision avoidance and results

• Hybrid Reachability
  – Implementing the reach-avoid operator

• Example applications
  – Discrete abstraction
  – Display analysis
  – Autolander
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

\[ \sigma_1 = \text{initiate maneuver} \]

\[
\begin{align*}
\dot{x} &= f_S(x, \nu) \\
q_1
\end{align*}
\]

\[
\begin{align*}
\dot{x} &= f_A(x, \nu) \\
q_2
\end{align*}
\]

\[ t = \frac{\pi}{4} \]

\[
\begin{align*}
\dot{x} &= f_S(x, \nu) \\
q_3
\end{align*}
\]

\[ t = T \]

\[
\begin{align*}
\dot{x} &= f_A(x, \nu) \\
q_4
\end{align*}
\]

\[ \sigma_1 = \text{initiate maneuver} \]

\[
\begin{align*}
\dot{x} &= f_S(x, \nu) \\
q_7
\end{align*}
\]

\[ t = \frac{\pi}{4} \]

\[
\begin{align*}
\dot{x} &= f_A(x, \nu) \\
q_6
\end{align*}
\]

\[ t = T \]

\[
\begin{align*}
\dot{x} &= f_S(x, \nu) \\
q_5
\end{align*}
\]

\[ t = \frac{\pi}{2} \]

\[ f_S \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -\nu + \nu \cos \psi \\ \nu \sin \psi \end{bmatrix} \]

\text{dynamics in straight modes}

\[ f_A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -\nu + \nu \cos \psi - x_2 \\ \nu \sin \psi + x_1 \end{bmatrix} \]

\text{dynamics in arc modes}
Seven Mode Safety Analysis

unsafe set without maneuver

Unsafe Set without Maneuver

unsafe

safe

unsafe set with maneuver

Unsafe Set with Maneuver

unsafe

unsafe set with choice to maneuver or not?

Intersection of Unsafe Sets

unsafe set with choice to maneuver or not?
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

[Tomlin, Lygeros & Sastry, 2000]
Reach-Avoid Operator

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

\[ G(t) = \{ x \in \mathbb{R}^n \mid \phi_G(x, t) \leq 0 \} \]
\[ E = \{ x \in \mathbb{R}^n \mid \phi_E(x) \leq 0 \} \]

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

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

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

![Diagram showing states and transitions]
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.

[Diagram showing controllable TOGA and flare envelopes, with intersection points labeled.

- Existing interface:
  - Flare: Flaps extended, minimum thrust
  - TOGA: Flaps retracted, maximum thrust
  - Rollout: Flaps extended, reverse thrust

- Revised interface:
  - Flare: Flaps extended, minimum thrust
  - TOGA: Flaps retracted, maximum thrust
  - Rollout: Flaps extended, reverse thrust
  - Slow TOGA: Flaps extended, maximum thrust]
Application: Aircraft Autolander

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

\[
\begin{align*}
\frac{d}{dt} \begin{pmatrix} V \\ \gamma \\ z \end{pmatrix} &= \begin{pmatrix} m^{-1}[T \cos \alpha - D(\alpha,V) - mg \sin \gamma] \\ (mV)^{-1}[T \sin \alpha + L(\alpha,V) - mg \cos \gamma] \\ V \sin \gamma \end{pmatrix} \\
\end{align*}
\]
Landing Example: Discrete Model

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

• 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
Eulerian Approaches to Backward Reachability in Continuous and Hybrid Systems

For more information contact

Ian Mitchell
Department of Computer Science
The University of British Columbia

mitchell@cs.ubc.ca
http://www.cs.ubc.ca/~mitchell