Additional Material on Reachability

Ian Mitchell
Department of Computer Science
The University of British Columbia

research supported by
National Science and Engineering Research Council of Canada

Outline

- Alternative algorithms for reachability of continuous and/or hybrid systems
  - Further comments [Mitchell, HSCC 2007]
- Hybrid system reachability with HJ PDEs
- Reachable set sensitivity
  - [Mitchell, HSCC 2007]

Lots of Algorithms

[Chutinan & Krogh, IEEE TAC 2003]
[Kurzhanski & Varaiya, HSCC 2000]

Alternative Eulerian Approaches

- Static Hamilton-Jacobi
  - Falcone, Ferretti, Soravia, Sethian, Vladimirsky
  - Minimum time to reach
  - (Dis)continuous implicit representation
  - Solution provides information on optimal input choices
- Viability kernels
  - Saint-Pierre, Aubin, Quincampoix, Lygeros
  - Based on set valued analysis for very general dynamics
  - Discrete implicit representation
  - Overapproximation guarantee
- Backward approach typical of Eulerian algorithms
  - Representation not moving (although it may adapt)
  - Generally handle nonlinear and multiple inputs
  - No examples beyond four dimensions?
State Space Decomposition

- Partition state space and compute reachability over partition
- Examples
  - Uniform grids: Kurshan & MacMillan, Belta and many others
  - Timed Automata “Region Graph”: Alur & Dill
  - Cylindrical Algebraic Decomposition: Tiwari & Khanna
- Advantages: No need to integrate dynamics, direct control over size of representation
- Disadvantages: Restricted classes of dynamics, “wrapping” problem (discrete system has transitions that do not exist in continuous system)

Lyapunov-like Methods

- Invariant sets are isosurfaces of Lyapunov-like functions
- Examples:
  - Convex optimization: Boyd, Hindi, Hassibi
  - Sum of Squares: Prajna, Papachristodoulou, Parrilo
- Advantages: Short certificate proves analytic invariance, no need to integrate dynamics
- Disadvantages: Restricted class of dynamics, difficult to extract counterexamples

Lagrangian Approaches

- “Lagrangian” computation is performed along trajectories of the system
  - Compare with “Eulerian” computation, which occurs on a grid which does not move with the trajectories
- Typically defined in terms of forward reach sets & tubes
- Advantages: Compact representation of sets, overapproximation guarantees, demonstrated high dimensions
- Disadvantages: Restricted dynamics, reliance on trajectory optimization, restrictive set representation

Examples of Lagrangian Schemes

- Timed automata
  - Derivatives are zero or one; continuous variables are “stopwatches”
  - Uppaal [Larsen, Pettersson…], Kronos [Yovine,…], …
- Rectangular differential inclusions (“linear” hybrid automata)
  - Derivatives lie in some constant interval
  - Hytech, Hypertech [Henzinger, Ho, Horowitz, Wong-Toi, …]
- Polyhedra and (mostly) linear dynamics
  - Derivatives are linear (or affine) functions
  - Checkmate [Chutinan & Krogh], d/dt [Bournez, Dang, Maler, Pnueli, …], PHAVer [Frehse], Coho [Greenstreet, Mitchell, Yan], others [Bemporad, Morari, Torrisi, …], …
- Ellipsoids and linear dynamics
  - [Botchkarev, Kurzhanski, Kurzhanskiy, Tripakis, Varaiya, …]
- Zonotopes and linear dynamics
  - [Girard, Le Guernic & Maler]
Four Examples of Lagrangian Schemes

- CheckMate & convex polygons
- Zonotopes
- Ellipsoids
- Coho & projectagons

Note:
- Choices are heavily influenced by my expertise
- I may choose different (and potentially conflicting) variable names in these slides when compared with the papers on which they are based

CheckMate

- Designed to verify properties of Polyhedral Invariant Hybrid Automata (PIHA)
  - Hybrid automata with invariants/guards defined by conjunctions of linear inequalities (convex polyhedra)
- Works by computing an Approximate Quotient Transition System (AQTS)
  - Discrete transition system which conservatively simulates the hybrid automata’s evolution
- Released as an add-on to Mathworks’ Simulink / Stateflow
  - Model can be constructed graphically
  - Same model can be simulated and verified

Continuous Algorithm

- Start with an initial set $X_0$
- Reach set $V_t(X_0)$ at a later time $t_k$ can be determined by simulating trajectories from each vertex of $X_0$
- Given reach set at $t_k$ and $t_{k+1}$, initial approximation of reach tube for $[t_k, t_{k+1}]$ is convex hull of $V_{t_k}(X_0)$ and $V_{t_{k+1}}(X_0)$
- Trajectories may curve, so use optimization to push edges of convex hull outward until reach tube contains all reachable states
- For linear dynamics $\dot{x} = Ax$, analytic solution is $\xi(t; x_0, t_0) = e^{A(t-t_0)}x_0$, so optimization is a linear program for any fixed $t$ (easy to solve)

Continuous Algorithm’s Issues

- Global nonlinear optimization provides no guarantees
  - Dilated convex hull may not contain all possible trajectories
- Trajectories are approximated numerically
- Accommodating inputs requires additional trust in optimization procedure

CheckMate reach tube examples for 2D Van der Pol model and a 3D linear model
Constructing the AQTS

- Reach tube construction is used to determine what set of states on an incoming polyhedral invariant face maps to which set of states on an outgoing polyhedral invariant face.
- Sets of states on face are mapped to discrete states in the AQTS (with possible subdivisions).

Primary CheckMate Papers

  - Details of the scheme for approximating continuous “flow pipes” (forward reach tubes).
  - Procedure for constructing the AQTS and hence verifying a model for a continuous system, assuming a scheme for computing continuous reachable sets.
  - Journal version of CDC paper, including proof of flow pipe approximation convergence & detailed batch evaporator example.
- Numerous other papers (see CheckMate web site).

CheckMate Outcomes

- Most complete tool for hybrid systems with non-constant dynamics.
  - (Partially) integrated with commercial design package.
  - Handles hybrid system verification, not just continuous reachability.
  - Generates counter-examples on failure.
  - Later work integrated Counter-Example Guided Abstraction Refinement (CEGAR) [Clarke, Fehnker, Han, Krogh, Stursberg, Theobald, TACAS 2003].
- Unable to move beyond low dimensions.
  - Polyhedral representation grows too complex.
  - One proposal: Oriented Rectangular Hull representation [Krogh & Stursberg, HSCC 2003].

A Brief Description of d/dt

- Similar basic idea to CheckMate.
  - Encorporates “griddy polyhedron” construction to control complexity of full reach set representation.
  - Various continuous reachability extensions: competing inputs, projections, …
- Many publications.
  - Fig. 2, p. 25 shown at right.
Zonotopes

- Representation of general convex polyhedra is too complex in higher dimensional spaces
- Instead, choose a category of sets that can be efficiently represented
- Zonotopes:
  - Image of a hypercube under an affine projection
  - Minkowski sum of a finite set of line segments

\[ Z = (c, <g_1, \ldots, g_p>) \]

\[ Z = \left\{ x \in \mathbb{R}^n \mid x = c + \sum_{i=1}^{p} \lambda_i g_i, \lambda_i \in [-1, +1] \right\} \]

where \( c, g_1, g_2, \ldots, g_p \) are vectors in \( \mathbb{R}^n \)

Zonotope Features

- Compact representation: storage cost \( n(p + 1) \)
- Closed under linear transformation: if \( Lx = Ax + b \) then
  \[ LZ = (Lc, <Lg_1, \ldots, Lg_p>) \]
- Closed under Minkowski sum:
  \[ Z^{(1)} + Z^{(2)} = (c^{(1)} + c^{(2)}, <g^{(1)}_1, \ldots, g^{(1)}_p, g^{(2)}_1, \ldots, g^{(2)}_p>) \]
- Conversion to other representations can be expensive; for example, a zonotope may have \( 2p \) choose \( n - 1 \) facets
- Computation of intersection and union may be difficult; for example, see [Girard & Le Guernic, HSCC 2006]

Linear Dynamics with Bounded Inputs

Restrict class of ODEs to the form

\[ \dot{x} = Ax + u, \quad u \in U \]

where \( U \) in this case is a hypercube

\[ U = \{ u \in \mathbb{R}^n \mid ||u||_{\infty} \leq \mu \} \]

- \( f(x, u) = Ax + u \) is Lipschitz in \( x \), so standard existence and uniqueness results apply
- Trajectories now denoted by \( \xi(t; x_0, t_0, u(.)) \) where function \( u(.) : \mathbb{R} \to U \) is an input signal
- Reach set with fixed (but not necessarily constant) input signal is the same as the input-free case
- Reach set with general input signal is the union over all possible fixed input signals

Continuous Algorithm

- Decompose full reach tube into segments
  \[ F(I, [0, T]) = \bigcup_i F(I, [i\tau, (i+1)\tau]) \]
  for some small timestep \( \tau \)
- Time-independent ODEs have the semigroup property
  \[ \xi(t_1 + t_2; x_0, 0) = \xi(t_2; \xi(t_1; x_0, 0), t_1) \]
  We can use the semigroup property to deduce
  \[ F(I, [ir, (i + 1)r]) = F(F(I, [i\tau, (i+1)\tau]), r) \]
- Therefore, if we can conservatively approximate \( F(I, [0, r]) \) and \( F(Z, r) \) for any \( Z \), we can conservatively approximate \( F(I, [0, T]) \)
Conservative Approximations

- Let $\| \cdot \| = \| \cdot \|_\infty$, “$+$” for sets be interpreted as the Minkowski sum and $\boxplus(\rho) = \{ x \in \mathbb{R}^n | \| x \|_\infty \leq \rho \}$ (which is a zonotope)
- $F(Z, r) \subseteq e^{tA}Z + \boxplus(\beta_r)$ where
  - $\beta_r = e^{r \| A \|} \frac{1}{\| A \|} - I$
- $F(Z, l, r) \subseteq P + \boxplus(\alpha_r)$ where
  - $\alpha_r = (e^{r \| A \|} - 1 - r \| A \|) \sup_{Z} \| x \|$
  - $P = \frac{1}{2} \left( \begin{array}{c} g_1 + e^{A}g_1, \ldots, g_p + e^{A}g_p \\ c - e^{A}c_1, \ldots, g_p - e^{A}g_p \end{array} \right)$
- These approximations can be shown to converge (in the Hausdorff metric) as $r \to 0$

Further Work

- Complexity of zonotopes in basic algorithm grows with time
  - Can conservatively constrain the order of the zonotope
- [Girard, Le Guernic & Maler, HSCC 2006]
  - Refactorizes the Minkowski sum to avoid growth of order
  - Constructs underapproximations and interval hull approximations
  - Discusses extension to hybrid automata (requires set intersection)
- [Girard & Le Guernic, HSCC 2008]
  - “Efficient” Algorithm for zonotope intersection with hyperplane

Zonotope Outcomes

- Still primarily a research project
  - MATISSE tool implements the continuous reachable set computation (including HSCC 2006?)
- Demonstrated on continuous toy examples in dimension 100 (HSCC 2005) and 200 (HSCC 2006)
- Demonstrated on low dimensional hybrid examples
- Zonotope representation has interesting trade-offs
  - Difficulty of computing set intersection and (presumably) union may make abstraction refinement challenging
  - Infinity norm bounds require well scaled system dynamics and inputs

Ellipsoids

- An alternative class of sets which can be efficiently represented in high dimensions
- Represent as the zero level set of a quadratic function
  - So computational costs in a given dimension are similar to LQR or Kalman filtering

Ellipsoid $E(z, Z) \subset \mathbb{R}^n$ is specified by

$$ E(z, Z) = \{ x \in \mathbb{R}^d | (x - z)^T Z^{-1} (x - z) \leq 1 \} $$

- $z \in \mathbb{R}^n$ is the center
- $Z \in \mathbb{R}^{n \times n}$, $Z = Z^T > 0$ is the shape matrix
- Eigenvectors of $Z$ specify the directions of the axes
- Eigenvalues of $Z$ specify the lengths of the axes
Ellipsoid Features

- Compact representation: $\frac{1}{2}n^2 + O(n)$
- Operations (union, intersection, Minkowski sum, etc.) on ellipsoids rarely give rise to ellipsoids
  - However, inner and/or outer bounding ellipsoids of the results can often be constructed analytically or by convex optimization
  - See various works by Kurzhanski, Kurzhanskiy, Vályi, Varaiya and many others

Two ellipsoids (red & blue) and an ellipsoid bounding their intersection (green)

Two ellipsoids (red & green), their actual Minkowski sum (black), and two ellipsoids bounding their Minkowski sum (cyan & blue)

Ellipsoidal Reachability

- Restrict dynamics to be linear
  \[ \dot{x} = Ax + Bu \]
  where $A$, $B$ are matrices, and input $u \in U = \mathcal{E}(p, P)$
- Even if $I = \mathcal{E}(q, X_0)$, reach set $F(I, t)$ is not an ellipse
- It is possible to construct tight external and internal bounding ellipsoids which touch the reach set at known points $\ell^*(t)$
  - Choose $\ell^*(t)$ as a solution to the adjoint of the homogenous dynamics
    \[ \ell^* = -A^T\ell^* \]
    for some $\ell^*(0) = \ell_0$.
  - We can write a recurrence for the tight ellipsoids' parameters

Ellipsoid Features

- Construct outer bounding ellipsoid
  \[ X^+_I(t) = \mathcal{E}(x_c(t), X^+_I(t)) \] such that $F(I, t) \subseteq X^+_I(t)$
- Center is just a trajectory (remember $u \in \mathcal{E}(p, P)$)
  \[ x_c(t) = Ax_c(t) + Bp \quad x_c(0) = x_0 \]
- Shape satisfies a matrix ODE
  \[ X^+_I(t) = AX^+_I(t) + X^+_I(t)A^T + \pi(t)X^+_I(t) + \frac{DPD^T}{\pi(t)} \]
  \[ X^+_I(0) = X_0 \]
  \[ \pi(t) = \left( \frac{\ell^T(t)DPD^TY^T(t)\ell(t)}{\ell^T(t)X^+_I(t)Y^T(t)\ell(t)} \right)^{\frac{1}{2}} \]
  \[ Y(t) = e^{At} \]
- A similar recurrence can be defined for an inner ellipsoid $X^-_I(t)$ such that $X^-_I(t) \subseteq F(I, t)$

External Bounding Ellipses

- Actual derivation allows dynamics and input set to be time-dependent
- Also derived for systems with two inputs: “control” and “disturbance”
  - State $x$ is reachable if there exists an initial condition in $J$ and a feedback control signal $u(t)$ that drives a trajectory to $x$ for every possible disturbance signal $v(t)$
  - In practice, compute bounding ellipsoids for several different $\ell$
    - For verification, test if all (outer) or any (inner) ellipsoid intersects with the target
    - For visualization and other operations, can compute bounding ellipsoids for intersections and unions
    - Shown at right: two outer and three inner bounding ellipsoids; actual reach set is contained in the intersection of the outer and the union of the inner

Further Work

- From Kurzhanski & Varaiya, HSCC 2000, Fig. 2 & 3, p. 212–213
Ellipsoid Outcomes

- Described in a whole series of papers by Kurzhanski & Varaiya
- Implemented in Ellipsoidal Toolbox (ET) by Kurzhanski
  - Documentation for ET provides concise summary of previous work
- Demonstrated in dozens of dimensions
- Demonstrated on low dimensional hybrid problems [Botchkarev & Tripakis, HSCC 2000] and ET
- Ellipsoid representation has different trade-offs
  - Extensive historical work on geometric operations makes extension to hybrid system reachability seem more feasible
  - Complexity of representation cannot be tuned: always $\frac{1}{2}n^2 + O(n)$
  - General linear input with ellipsoidal bounds adds flexibility

Coho & Projectagons

- Two dimensions is easy: Lots of fast, powerful algorithms
  - Can we design an algorithm that primarily works in two dimensional subsets of the full state space?
- “Projectagons”
  - Subset of high dimensional polyhedrons which can be represented as the intersection of a collection of prisms
  - Each prism is the infinite extension (into the other dimensions) of a bounded (potentially nonconvex) two dimensional polygon
  - We actually track only the two dimensional projections

Ellipsoid Outcomes

- Two dimensions is easy: Lots of fast, powerful algorithms
  - Can we design an algorithm that primarily works in two dimensional subsets of the full state space?
- “Projectagons”
  - Subset of high dimensional polyhedrons which can be represented as the intersection of a collection of prisms
  - Each prism is the infinite extension (into the other dimensions) of a bounded (potentially nonconvex) two dimensional polygon
  - We actually track only the two dimensional projections

Evolving a Projection (1)

- Let projectagon be $P$ and the prism represented by projection $j$ be $P_j$, so $P = \cap_j P_j$
- Then $CH(P) \subseteq \cap_j CH(P_j)$, where $CH(P)$ is the convex hull of $P$
- $CH(P_j)$ is easily computed and can be represented by the conjunction of a set of linear inequalities
- Loosen (“blow”) all inequalities by $\epsilon$
- Now consider an individual edge $e_i$ in the two dimensional projection of $P_j$, which corresponds to a face of $P$
- Construct a box bounding all states within $\epsilon$ of $e_i$ (also a conjunction of linear inequalities)
- The conjunction of all of the inequalities represents all states within $\epsilon$ of the face of $P$ corresponding to $e_i$

Evolving a Projection (2)

- Construct an affine plus error model $\dot{x} = Ax + b + u$ for $u \in U$ and $U$ a hyperrectangle that is valid within the conjunction of all these linear inequalities
- The forward time mapping of states under this dynamic is linear (if $b = 0$ and $U = \emptyset$ then it is $e^{At}$)
- Use linear programming to compute the polygonal projection of the forward time mapping of $e_i$
- Repeat for all edges in the projection
- Compute the union of all forward time polygons (and all states inside that union)
- Simplify if necessary
- Repeat for all projections
- Repeat for next timestep
Practical Aspects

- Geometry and mathematics are well separated
  - Geometry operations in Java, linear programs (LPs) and model computation in Matlab
- LPs are nasty
  - Lots of (nearly) redundant and (nearly) degenerate inequalities
  - Lots of sparsity (only two nonzeros per row)
  - Need to walk the projection (start from nearly optimal point)
  - Need guaranteed optimum for guaranteed overapproximation
  - Led to specialized LP implementation by Laza & Yan: takes advantage of special structure, uses regular floating point calculations to start but guarantees solution accuracy through interval arithmetic and if necessary arbitrary precision arithmetic
- Careful simplification of projections is important
  - Need to keep number of edges under control, but accuracy degrades significantly if nonconvexity is removed
- Choice of projections is not always obvious

Coho Outcomes

- Implemented at UBC in Coho toolset
- Demonstrated on seven dimensional realistic circuit model of a toggle element [Yan & Greenstreet, FMCAD 2007]
  - Included verification of composability to construct a ripple counter
- Projectagons are not as scalable as zonotopes & ellipsoids, but can represent nonconvex reach sets
  - Ample opportunity for parallelization
- Algorithm has considered automatic construction of linear plus error models from nonlinear circuit models

Outline

- Alternative algorithms for reachability of continuous and/or hybrid systems
  - Further comments [Mitchell, HSCC 2007]
- Hybrid system reachability with HJ PDEs
- Reachable set sensitivity
  - [Mitchell, HSCC 2007]

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

\[ f_S \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -v + v \cos \psi \\ v \sin \psi \end{bmatrix}, \quad f_A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -v + v \cos \psi - x_2 \\ v \sin \psi \end{bmatrix} \]

[Tomlin, Mitchell & Ghosh, 2001]

Seven Mode Safety Analysis
- Ability to choose maneuver start time further reduces unsafe set

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]

- 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

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]
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

Landing Example: Mode Switches

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
Outline

- Alternative algorithms for reachability of continuous and/or hybrid systems
  - Further comments [Mitchell, HSCC 2007]
- Hybrid system reachability with HJ PDEs
- Reachable set sensitivity
  - [Mitchell, HSCC 2007]

Trajectory Sensitivity

- To approximate reach sets and tubes, direct algorithms integrate trajectories
- Small(?) perturbations occur in representation
  - Floating point roundoff
  - Simplified dynamics
  - Approximating the true set with a larger set from the appropriate class
- How might the interaction of perturbations and dynamics affect the quality of the approximation?

\[ \xi_H(t; z_0 + \delta x, 0, u(\cdot)) \]
Sensitivity of Forward Reachability

\[ \| \zeta_{\text{H}}(s; z) \| \leq \| \Phi(s) \| \| \zeta_{\text{H}}(s; z) \| \| \delta x \|. \]

- Sensitivity matrix can become large via
  - Deterministic models are rarely used for such systems

Sensitivity of Backward Reachability

- System dynamics are reversed
  - Sensitivity matrix can become large via

\[ \lambda(\Phi) \approx 0 \text{ backward continuous evolution} \]

\[ \lambda(\Phi) \approx 1 \text{ backward discrete jumps} \]

\[ \nabla_{\psi^T} f \approx 0 \text{ backward grazing contact with switching surface} \]

- Systems which show contraction are likely to be ill-conditioned for backward reachability
  - Such systems are commonly encountered, because their models are well-conditioned in forward time

Continuous System Sensitivity Example

- Toggle circuit [Yuan & Svensson, IEEE JSSC 1998]
  - Period of output is double period of input
  - Short channel transistor model with velocity saturation, all capacitance to ground and interconnect capacitance is ignored
  - Forward verification that chain of toggles can operate as a counter
  - Thanks: Mark Greenstreet, Chao Yan & Suwen Yang for simulation

Toggle Circuit Sensitivity

- System dynamics has components which are strongly contractive
  - Sensitivity matrix of continuous dynamics has eigenvalues with large negative real component

\[ F(q, x, u) \Delta D_x f(q, x, u) \]

- Backward reachability will be ill-conditioned
Discrete System Sensitivity Example

• Adapted from rocking block in [Lygeros, Johansson, Simic, Zhang & Sastry, IEEE TAC 2003]
  – Discrete control input can change location of center of mass

\[
f_L(x, \alpha) = \frac{1}{\alpha} \sin(\alpha(1 + x_1)) \quad f_R(x, \alpha) = \frac{1}{\alpha} \sin(\alpha(1 - x_1))
\]

\[
\alpha_{\text{big}} = \frac{\pi}{3}, \quad \alpha_{\text{small}} = \frac{\pi}{6}, \quad \rho = 0.8, \quad \gamma = \frac{\alpha_{\text{big}}}{\alpha_{\text{small}}} = 2
\]

Rocking Block Sensitivity

• Two typical trajectories
  – Constant center of mass (blue) or switched (red)

• Forward behaviour
  – Final state is sensitive to initial conditions (tipped or not)
  – Switching (controlled or autonomous) is not locally sensitive

• Backward behaviour
  – Controlled switch is sensitive through interaction with reset
  – Reset is sensitive for \( \rho \ll 1 \)

Switching Surface Sensitivity

• Backward switching sensitivity is not obvious

\[
\nu = \frac{\partial h(x_0)}{\partial x_0} = \frac{\nabla \psi(x^+)^T \Xi_0(f^+(t^+, x^+, u))}{\nabla \psi(x^+)^T f(q^+, x^+, u)}
\]

\[
\nabla \psi(x^+)^T f(q^+, x^+, u) = 0 \text{ at } x^+ = \begin{pmatrix} 1 \\ 23 \\ 19 \end{pmatrix}
\]

Switching Surface on Reachability

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