Recent Advances in Verification and Analysis of Hybrid Systems

Janan Zaytoon
CReSTIC, University of Reims
France
Hybrid Systems: Examples

- Systems with commutations: electrical circuits

- Electric Networks: manage & optimize system configuration through discrete connections/disconnections of parts of the net to regulate electrical energy
The Heterogeneity of Systems

An Engine Control System
Models of Computation

- **Continuous Time**
  - continuous functions
  - continuous signals

- **Finite State Machine**
  - states
  - transitions

- **Discrete Event**
  - operations on events
  - occurrence time

- **Controller**

- **Power Train**

- **Sensors**
Different Approaches

- Hybrid Systems: Dynamical systems with interacting continuous and discrete dynamics

Applications:
- ATMS - Air Traffic Management Systems
- AHS - Automated Highway Systems
- Power Networks
- UAV - Uninhabited Aerial Vehicles

J. Zaytoon, ICINCO’09, Milan
Research Issues in Hybrid Systems

• Modeling & Simulation
  – classify discrete phenomena, existence and uniqueness of execution, Zeno
  – composition and abstraction operations

• Analysis & Verification
  – avoid or attain forbidden states: algorithmic or deductive methods, abstraction
  – stability, Lyapunov techniques, LMI techniques

• Controller Synthesis
  – optimal control, hierarchical control, supervisory control, safety specifications, control
  mode switching
  – algorithmic synthesis, synthesis based on HJB

• IFAC Technical Committee on Discrete Event and Hybrid Systems
  – IFAC Conference on Analysis and Design of Hybrid Systems (ADHS’03 in France,
    ADHS’06 in Italy, ADHS’09 in Zaragoza – Spain)

• IEEE WG Hybrid Systems
• Nonlinear Analysis: Hybrid Systems (International Journal, Elsevier)
• National groups, NOE, European and International projects, Annual
  Workshop on Hybrid Systems
Outline

• Safety verification and reachability
  – Hybrid automaton

• Abstraction
  – Conserve hybrid nature of the system
  – Discrete-Event abstraction

• Characterizing reachable space

• Reachable space computation (overapproximation)
Hybrid Automaton

- \(<L, X, U, INV, F, E, Guard, Jump, l_0, x_0, u_0>\)

- \(\text{state } (l, x, u) \in L \times X \times U\)

- Composition

\(l_1\)
\[\begin{align*}
y &= f(l_1, x, u) \\
\dot{x} &= f(l_1, x, u)
\end{align*}\]
\((x, u) \in \text{Guard}(l_1, l_2)\)
\(x \in \text{Jump}((l_1, l_2), x, u)\)

\(l_2\)
\[\begin{align*}
y &= f(l_2, x, u) \\
\dot{x} &= f(l_2, x, u)
\end{align*}\]
\((x, u) \in \text{Guard}(l_2, l_3)\)
\(x \in \text{Jump}((l_2, l_3), x, u)\)

\(l_3\)
\[\begin{align*}
y &= f(l_3, x, u) \\
\dot{x} &= f(l_3, x, u)
\end{align*}\]
\((x, u) \in \text{Guard}(l_3, l_1)\)
\(x \in \text{Jump}((l_3, l_1), x, u)\)
Reachable Sets

- Execution: Admissible trajectories described by a succession of continuous & discrete evolutions
- State can advance by progression of time in the current location or by an instantaneous transition to a new location
- Continuous & discrete successors (predecessors) for a point or a region

$\text{Inv}(l_1)$, $\text{Guard}(l_1,l_2)$, $\text{Guard}(l_3,l_5)$, $\text{Inv}(l_3)$, $\text{Jump}((l_1,l_3),x,u)$, $\text{Jump}((l_1,l_2),x,u)$, $\text{Jump}((l_2,l_4),x,u)$
Algorithmic Verification: Safety verification

- Since the state space of HS implicitly includes time, many properties of HS can be expressed as reachability properties.
- Safety properties (is the system dangerous to itself or to its environment): Verify, through reachability computation, that for any initial condition, the hybrid state can never enter some unsafe region.
- Decidability is a central issue in algorithmic analysis because of the uncountability of the hybrid state space.
Hybrid Reachability based Verification

- Computation of the reachable set: starting at Init, determine the limit of the series of regions defined by
  \[ R_i = \text{Succ}_C(\text{Init}) \]
  \[ R_{i+1} = R_i \cup \text{Succ}_C(\text{Succ}_D(R_i)) \]

- exactly for some very simple classes of systems: Piecewise constant differential inclusions, some linear systems
- approximately for other classes: over-approximation algorithms, set-based simulation
Outline

• Verification and reachability
• **Abstraction**
• Characterizing reachable space
• Reachable space computation
Abstraction

- \( S_2 \) is an abstraction of \( S_1 \) iff the image of each trajectory of \( S_1 \) is also a trajectory of \( S_2 \) (but some executions in \( S_2 \), introduced by the abstraction process, may not be related to trajectories in \( S_1 \)).

- If \( S_2 \) is safe then \( S_1 \) is safe.

- Linear differential inclusion abstraction
- Discrete event abstraction
Linear differential inclusion abstraction or hybridization (Henzinger et al., 98; Frehse, 05; Lefebvre, Gueguen & Zaytoon, 06):

- Approximation of complex continuous dynamics by simpler hybrid dynamics
- Calculate differential inclusion that includes the derivative vector defined by the continuous dynamics at each point of the invariant of a location
- Use the differential inclusion (derivative vectors $\gamma_1$ and $\gamma_2$) to compute the reachable space from $P_0$
- The resulting abstraction (resulting HLA) is generally too coarse, and hence the overapproximated reachable space does not allow us to conclude for safety verification

J. Zaytoon, ICINCO’09, Milan
Hybridization: Refine the abstraction

- Partition the invariant of a location into n subsets and replace the location with n locations whose reachable spaces are over-approximations of the corresponding subset region.

\[ \dot{x} \in F_l(x) + Bu \]

Over-approximated region (polyhedron) containing the derivative vector defined by the continuous dynamics at each point of the subset.
Linear differential inclusion abstraction

- Include a transition between two sub-locations of a location if there exists a continuous trajectory crossing the boundary between the corresponding elements of the partition
- For each $e(l_i \rightarrow l_j)$, include a transition from each sub-location of $l_i$ intersecting $\text{Guard}(e)$ to each sub-location of $l_j$ intersecting $\text{Jump}(e)$
- Then calculate reachability using the resulting abstraction
Reachability

- Refine abstractions if resulting regions are too coarse
- No guarantee that this abstraction will eventually allow to conclude
- Difficulty: determine a pertinent criteria to refine the partition to improve the efficiency of reachability calculation
  - Continuous dynamics can be used to determine the regions defining the partition of the state space (tradeoff: precision of abstraction vs. simplicity of calculation)

\[ \begin{align*}
  x_1 & = \frac{z_1}{z_2} \\
  x_2 & = \frac{z_3}{z_4}
\end{align*} \]
Linear differential inclusion abstraction: Lefebvre, Guéguen, Zaytoon

- Simple case: Affine planar systems: \( \dot{x} = Ax + b \)

\[
\begin{align*}
& w_1^t (x - x_c) \leq 0 \quad \Rightarrow \quad v_1^t \dot{x} \leq 0 \\
& w_2^t (x - x_c) \geq 0 \quad \Rightarrow \quad v_2^t \dot{x} \leq 0
\end{align*}
\]

- Half lines defined by the equilibrium point are very useful in specifying the partition: at all points of this line, the derivative vector is collinear to a unique vector and, so, the trajectories cross the half-line in the same direction, leading to a very simple structure for the abstraction.

- The derivative vector of each point between 2 such half lines, is included in the convex hull of the 2 vectors characterizing the borderer lines, and this defines the differential inclusion of the abstraction.
Linear differential inclusion abstraction: Lefebvre, Guéguen, Zaytoon

- Resulting HA for a partition of 8 elements:
  - continuous dynamics in each location given by the differential inclusion representing the border line of the corresponding region
  - transition guards given by the border lines

\[
\begin{align*}
q_i^T(x-x_0) \geq 0 \\
(y_i^T \tilde{x}) \geq 0 \\
(y_i^T \tilde{x} \leq 0)
\end{align*}
\]
Affine systems: \( \dot{x} = Ax + b \)

\[ H = \{ x \mid q^T x = k \}, \text{ where } \exists \gamma \text{ s.t. } q = A^T \gamma, k = -\gamma^T b \]

- For higher dimension affine systems, it is possible to consider families of hyperplanes with certain constraints s.t. all trajectories cross the hyperplanes in the same direction, leading to a very simple transition structure for the abstraction.

- Extension to systems defined by: \( \dot{x} = Ax + Bu \)

\( U : \) space of continuous inputs is a polytope (Nasri et al., 06)
Discrete Event Abstraction: Alur et al. 03, Chutinan & Krogh 03, Tiwari & Khanna 04, Ratschan & She 05, Blouin et al. 03, Kloetzer & Belta 06

- Construction
  - partition of state space (consider specific regions: guard, invariants, $R_{init}, R_{unsafe}$, and other regions linked to the property or sometimes their borders)
  - associate an abstract discrete-state to each element of the partition
  - Calculate the transitions: constraint to satisfy
    $$(l_k, x_k) \in \text{Reach}(l_n, x_n) \Rightarrow \pi((l_k, x_k)) \in \text{Succ}((l_n, x_n)))$$
    
    $$R_{unsafe} \cap \text{Reach}(R_{init}) = \phi$$
    $$q_{unsafe} \notin \text{Succ}(q_{init})$$
  - If safety condition is not satisfied, iterate the abstraction

$J. Zaytoon, ICINCO’09, Milan$
DE Abstraction

• Choice of discrete states

Abstraction on guards

\[ q_e = \pi(l_k, G_e) \]

transition from \( q_1 \) to \( q_3 \) stems from the continuous reachability of \( G_3 \) from \( D_2 \)

Include a transition from \( q_a \) to \( q_b \) if \( G_b \subseteq \text{Succ}_C(\text{Succ}_D(G_a)) \)

Abstraction on borders:

Include a transition from \( q_a \) to \( q_b \) if \( b_b \subseteq \text{Succ}_C(\text{Succ}_D(\text{Succ}_C(b_a))) \)
Abstraction

- Spurious transitions due to abstraction
- Iterative algorithm to refine the abstraction (Tabuada et al., 2002)
Consider a discrete transition & partition the continuous domain of the region mapped to the source location
If \( \text{Pred}_D(\text{Pred}_C(l_p, D_p)) \cap D_k \neq D_k \), split \( D_k \) to:
  \[
  \begin{align*}
  D_{k1} &= \text{Pred}_D(\text{Pred}_C(l_p, D_p)) \cap D_k \\
  D_{k2} &= D_k - (\text{Pred}_D(\text{Pred}_C(l_p, D_p)) \cap D_k)
  \end{align*}
  \]
If \( \text{Pred}_D(\text{Pred}_C(l_p, D_p)) \cap D_k = D_k \), no change

- Difficulty: choice of transition to refine:
  - transitions leading to regions close to forbidden area
  - Transitions close to counter-example trajectory provided by verification
Outline

• Verification and reachability

• Abstraction

Building an abstraction requires the determination of reachable regions: 2 types of answers

- if the problem is to decide whether there is a discrete transition between 2 locations in case of hybridization or 2 discrete states in case of DE abstraction, use methods that give a yes/no answer

• Characterizing reachable space

- to refine the DE abstraction

• Reachable space computation

In both approaches, reachability calculation is only related to 1 location or 2 successive locations

J. Zaytoon, ICINCO’09, Milan
Characterizing reachable space

• Is it possible to reach a region $P_c$ from region $P_0$ without explicitly computing the reachable space?

• Display borders separating the two domains and uncrossable by continuous trajectories

• Constraints inconsistency: determine partial (easier to compute) characteristics of reachable and goal region and prove their inconsistency

• Existence of Trajectories from $P_0$ to $P_c$??
Uncrossable borders: Use structural properties of continuous dynamics to define borders characterising invariant domains that continuous trajectories never leave & include initial region (Tiwari, 03, Rodriguez & Tiwari 05)

- Example: linear dynamics $\dot{x} = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix} x$
  +ve real eigenvalues $\lambda \ (2, 4)$

- $c_1=(1 \ 0)^T, \ c_2=(0 \ 1)^T \rightarrow c^T x \geq \min_{P_0} (c^T x)$ if $\lambda > 0$
  $\rightarrow$ reachable space characterized by: $c_1^T x \geq 1, \ c_2^T x \geq 1$
  $\rightarrow P_{C1}$ unreachable, $P_{C2}$??

- Extension to complex $\lambda$
Inconsistent Temporal constraints on reachability in eigenspaces (Yazarel & Pappas 04)

- \( \dot{x} = Ax \): Projection of trajectory from \( x_0 \) on eigenspaces (of dimension 1) associated with real eigenvalues
- Compute min & max time necessary (through linear programming) to go from projection of \( P_0 \) to projection of \( P_C \) for each eigenspace
- Check for –ve value of max time or check emptiness of intersection of time intervals from different eigenvectors
- Projections of \( P_0 \) & \( P_{C_1} \) on subspace defined by eigenvector (1,0):
  - bounds: \((-\infty \ 0.5\ln0.5)\)
  - \( t_u < 0 \rightarrow P_{C_1} \) unreachable from \( P_0 \)
- Projections of \( P_0 \) & \( P_{C_2} \) on (1,0):
  - bounds: \((0.5 \ln1.25 \ 0.5 \ln3.5)\)
  - Projections of \( P_0 \) & \( P_{C_2} \) on (0,1):
    - bounds: \((0 \ 0.25 \ln1.5)\)
    - since \(0.25 \ln1.5 < 0.5 \ln1.25 \rightarrow P_{C_2} \) unreachable from \( P_0 \)
- The more the number of eigenvalues associated with eigen subspace of dimension 1, the more the chances to conclude that \( P_c \) is unreachable

J. Zaytoon, ICINCO’09, Milan
Inconsistent Spatial (polynomial) constraints on reachability in eigenspaces (Yazarel et al. 04)

- \( \dot{x} = Ax \), \( A \) diagonalizable with rational eigenvalues \( \lambda_i \) or nilpotent with pure imaginary eigenvalues
- reachable points on eigenspace of \( \lambda_i \) can be characterized with a set of polynomial constraints
- Check that no point fulfills all constraints through SOS optimization \( \rightarrow \) goal region unreachable from initial region

- no point in \( P_{c2} \) fulfills \( C2, C3 \) \( \rightarrow \) \( P_{c2} \) unreachable
- Constraint on positivity of time: \( x_1^2 + 2x_2^2 \geq 3 \)
- no point in \( P_{c1} \) fulfills \( C2, C3, C4 \) \( \rightarrow \) \( P_{c1} \) unreachable
Barrier certificates (e.g. Prajna et al. 07, Glavaski et al. 05)

\[ \forall x \in X, \forall u \in U : B(x) = 0 \Rightarrow \frac{\partial B(x)}{\partial x} f(x, u) \leq 0 \]

- Choice of type of \( B(x) \)
- SOS Optimization if \( B \) and dynamics are polynomial

Existence of a trajectory: reachability certificate (Prajna & Rantzer, 05)

- For \( \dot{x} = f(x) \), \( \exists \) a trajectory from \( P_0 \) to \( P_C \) if \( \exists \) a function \( \rho \) st:
  \[ \int_{P_0} \rho(x) dx > 0 \]
  \( \forall x \in \text{closure}(\text{bound}(\text{Inv}) - \text{bound}(P_C)) \),
  \( \forall x \in \text{closure}(\text{Inv} - P_C) \),
  \( \text{div}(\rho f)(x) < 0 \)
Outline

• Verification and reachability
• Abstraction
• Characterizing reachable space
• Reachable space computation
Reachable space calculation

• When refining a DE abstraction

• Difficulty: integration of differential equations (infinite set of trajectories to simulate), time elimination

• Over-approximation to preserve safety property

For continuous systems specified by linear differential inclusions, the overapproximated regions can be determined with geometric considerations and polytopes computations.

Complex and difficult to implement: pay attention to the choice of regions.
Finite discrete time integration (Dang, Chutinan & Krog, Asarin et al., Gérard)

- Calculation of series of finite time successor regions, using sample-time computation
  - Guaranteed integration: Time step $\delta$, Finite number of steps

\[
\dot{x} = Ax \\
P_i = e^{A\delta} P_{i-1}
\]

\[
\dot{x} = Ax + u \\
P_i = e^{A\delta} P_{i-1} \oplus V
\]

where $A \oplus B = \{ a+b \mid a \in A \land b \in B \}$

J. Zaytoon, ICINCO’09, Milan
Space regions

• Choice of a type of sets for continuous space regions:
  – efficiency of their set representation
  – complexity of computation on this type of set (intersection, union, dynamic evolution, Minkowski sum)
  – Closure of this type of set wrt operations needed for reachability calculation to reduce complexity and approximation

• Polynomial regions (e.g. Dang, 2006)
• Ellipsoids (e.g. Kurzhanski & Varyia, 2000)
  – Compact and closed for transformations induced by linear dynamics
  – Not closed for other operations (ex: Minkowski sum), inducing important approximations

• Polyhedral sets
  – hyperrectangles – interval computation (Nedialkov et al., 1999)
  – Polyhedrons (linear constraints, vertex)
  – Zonotopes
Closure

- Hyperrectangles: all borders are normal to one of the basis vectors.
- Difficulty: hyperrectangles are not closed for continuous dynamics changes (wrapping effect).
- Express intermediate results in intermediate basis to overcome wrapping effect.
Polyhedral sets: Polyhedrons

- Complexity of representation due to iterative computation
  - Tight overapproximation to reduce number of constraints

- Efficient coding of constraints
  (Asarin et al., 06): overapproximation to encode constraints with lower number of bits
Polyhedral sets: Zonotopes

- Use for high dimension state space due to compact representation
- Closed for most operations involved in reachability computation (linear transformation, Minkowski sum)
- Problems: reduction of number of generators further to iteration of reachability computation, and computation of intersection with guards

Planar zonotope
Defined by its center and 3 generators
Complexity reduction: Continuous space dimension reduction

- Projection & uncertainty (e.g. Asarin & Dang, 04; Han & Krogh, 05): identify subspaces of state space st projection of state in one subspace has low influence on the projection of the state of the other
- Trajectories similarities (Girard, Pappas et al., 2006):
  - Approximation as a relaxation of the notion of abstraction
  - distance between trajectories rather than an inclusion relation
  - simulation functions defining approximate simulation relations: Lyapunov-like characterization, Algorithms (LMIs, SOS, Optimization)
  - reachability computations based on zonotopes
Analysis of complex systems

Abstraction methods for complexity reduction of systems.

Abstraction

Dimension of the continuous state space

Model complexity

Linear systems → Piecewise affine systems → Nonlinear systems → Hybrid system

Complex system

Dimension reduction

Hybridization

Abstraction
Conclusion: Structured presentation of formal verification techniques for Hybrid Systems

• Guaranty correct behavior
  – Methods and tools
• Safety properties: reachability and abstraction
• Non decidability results
• Various propositions
  – General principles
  – Representation of regions
  – Algorithms
doi:10.1016/j.arcontrol.2009.03.02

J. Zaytoon, ICINCO’09, Milan
Perspectives

• Safety verification for real-size applications require complementary approaches alternating overapproximation, characterization of reachable space, dimension reduction
• Methodology based on clear criteria to guide the choice of the approaches and their cooperation for a given class of applications and properties
• Integrating such approaches with other control design tools