Towards a Compositional Framework for Hybrid Differential Inclusions

Dan P. Guralnik  
University of Florida/NCR Lab

Jared Culbertson  
AFRL/ACT3

April 14, 2020
**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

For example, in networks where comms are constrained by distance,
**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

Switching between comms structures (e.g. spanning trees) is useful.
Motivation

**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

Coordinated motion under a fixed controller...
**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

...may run into obstacles...
**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

...suggesting a reassessment of the comms structure...
Challenge: Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

...including temporary disconnects with the aim of reconnecting soon thereafter...
**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

... using a different connectivity structure.
**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

In the presence of additional resources...
**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

...a reactive control paradigm may provide alternative solutions...
**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.
Motivation

**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

[rendevous generates new comms connections]
Motivation

**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

[less risky strategy becomes available]
Challenge: Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

[resolution through edge-creation and edge-exchanges]
**Challenge:** Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.
Challenge: Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

[they live happily ever after]
Challenge: Autonomous generation of complex distributed cooperative behaviors requires reasoning over very large combinatorial structures.

[they live happily ever after]

Here, “very large combinatorial structure” = the space of all spanning trees over a varying set of agents.
Emerging requirements:

- A rich formal “substrate” for symbolic representations of task domains

  ▶ Do not treat tasks on a case-by-case basis

  ▶ Logic is not easily made mindful of geometry/topology
Motivation

Emerging requirements:

▶ A rich formal “substrate” for symbolic representations of task domains

----> Do not treat tasks on a case-by-case basis
----> Logic is not easily made mindful of geometry/topology

▶ Mitigate explosive complexities through combinatorial/topological analysis of the underlying space of discrete structures

- Example: understandings about binary trees [AGK17] enabling efficient reactive collision-free navigation [AGK16] using a covering obtained by hierarchical clustering of configurations.
Emerging requirements:

- A rich formal “substrate” for symbolic representations of task domains
  \(\rightarrow\) Do not treat tasks on a case-by-case basis
  \(\rightarrow\) Logic is not easily made mindful of geometry/topology

- Mitigate explosive complexities through combinatorial/topological analysis of the underlying space of discrete structures
  - Example: understandings about binary trees [AGK17] enabling efficient reactive collision-free navigation [AGK16] using a covering obtained by hierarchical clustering of configurations.

- More generally, underlying discrete structures must be mindful of local-to-global interactions between task and constraints.
  \(\rightarrow\) We strongly suspect that these are functorial, and generalize broadly
We seek a framework combining:

- differential inclusions (continuous dynamics)
- jump/reset relations (discontinuous/switched dynamics)
- sequential and parallel composition (concatenation/coupling)
- maps between hybrid systems (HS)
- trajectories as such maps
We seek a framework combining:

- differential inclusions (continuous dynamics)
- jump/reset relations (discontinuous/switched dynamics)
- sequential and parallel composition (concatenation/coupling)
- maps between hybrid systems (HS)
- trajectories as such maps

These accentuate the need in a CATEGORICAL FRAMEWORK, to enable programming of behaviors using TYPE-THEORETIC tools.
“No Abstract Nonsense” Pledge. The proposed framework must enable the operationalization of the following:

- Refinement/coarsening arguments to identify behaviors/tasks
  - Template–Anchor pairs [FK99, CGKS19]
  - Other hierarchical compositions [RK18, Rev19b, Rev19a]

- Stability arguments for formal guarantees of robustness
  - The hybrid differential inclusions framework [GST09] is an example

- Computable invariants of task achievability
  - Homological invariants à-la Erdmann?

- Temporal tameness analysis (noZeno / goodZeno / badZeno & worse...)
  - Generalized hybrid time domains / hybrid arcs
  - Weaker topology on the space of hybrid arcs, à-la Conley
Initial Goal: Fuse the Hybrid Differential Inclusions (HDI) framework [GST09] with the categorical formulations of Culbertson et. al. [CGKS19] and Lerman–Schmidt [LS20].
Initial Goal: Fuse the Hybrid Differential Inclusions (HDI) framework [GST09] with the categorical formulations of Culbertson et. al. [CGKS19] and Lerman–Schmidt [LS20].

The Need: While comprehensive enough to provide general guarantees, HDI does not provide a systematic toolkit for leveraging the topology/combinatorics of large structured sets of continuous modes.

⇝ A categorical viewpoint to further empower HDI in applications
**Initial Goal:** Fuse the Hybrid Differential Inclusions (HDI) framework [GST09] with the categorical formulations of Culbertson et. al. [CGKS19] and Lerman–Schmidt [LS20].

**The Need:** While comprehensive enough to provide general guarantees, HDI does not provide a systematic toolkit for leveraging the topology/combinatorics of large structured sets of continuous modes.

\[\implies\text{A categorical viewpoint to further empower HDI in applications}\]

**Next Steps:** Develop an instance dedicated specifically to distributed control of mobile agent networks, using categorical descriptions for:

- Kruskal graph structure (edge exchanges) on the space of spanning trees
- Other computationally accessible classes of graphs, e.g., chordal graphs [CGS17]
- Parallel composition operations representing interactions among agent coalitions.
Existing Categorical Frameworks

**General mantra:** “Hybrid System = Graph of Dynamical Systems”

- **Ames [Ame06]:**
  - general “hybridization” construction for any category;
  - applies to smooth dynamical systems (no composition).

- **Haghverdi–Tabuada–Pappas [HTP05]:**
  - an open system version (both discrete and continuous control).
  - weakened notion of equivalence: bisimulation.

- **Lerman, Lerman–Schmidt [Ler18, LS20]:**
  - open systems as hybrid submersions;
  - interconnections via hybrid submersions between products.

- **Culbertson–Gustafson–Koditschek–Stiller [CGKS19]:**
  - hybrid semiconjugacies to construct template-anchor pairs;
  - Sequential composition using weakened notion of trajectory.
**Informal Tidbits: Compositions**

**Sequential composition** may be thought of as a *concatenation operator* on the trajectories of a pair of systems:

**CGKS [CGKS19]:** discuss difficulties with sequential composition of piecewise smooth (hybrid) trajectories, establishing the need for coarse notions of (1) hybrid trajectory and/or (2) hybrid time domains.
Parallel compositions.

- The simplest example is a decoupled Cartesian product of systems.
- In mobile agent networks, interconnection may be intermittent.

[While far apart, the two agent coalitions do not interact]

Ames [Ame06] and Lerman–Schmidt [LS20]: enable interconnections, but need to be reconciled with HDI and sequential compositions.
Parallel compositions.

- The simplest example is a decoupled Cartesian product of systems.
- In mobile agent networks, interconnection may be intermittent.

[While far apart, the two agent coalitions do not interact]

Ames [Ame06] and Lerman–Schmidt [LS20]: enable interconnections, but need to be reconciled with HDI and sequential compositions.
Informal Tidbits: Compositions

Parallel compositions.

- The simplest example is a decoupled Cartesian product of systems.
- In mobile agent networks, interconnection may be intermittent.

While far apart, the two agent coalitions do not interact.

Ames [Ame06] and Lerman–Schmidt [LS20]: enable interconnections, but need to be reconciled with HDI and sequential compositions.
Parallel compositions.

- The simplest example is a decoupled Cartesian product of systems.
- In mobile agent networks, interconnection may be intermittent.

[While far apart, the two agent coalitions do not interact]

Ames [Ame06] and Lerman–Schmidt [LS20]: enable interconnections, but need to be reconciled with HDI and sequential compositions.
Parallel compositions.

- The simplest example is a decoupled Cartesian product of systems.
- In mobile agent networks, interconnection may be intermittent.

[While far apart, the two agent coalitions do not interact]

Ames [Ame06] and Lerman–Schmidt [LS20]: enable interconnections, but need to be reconciled with HDI and sequential compositions.
Parallel compositions.

- The simplest example is a decoupled Cartesian product of systems.
- In mobile agent networks, interconnection may be intermittent.

[Once communication is established, their motion may be coupled...]

Ames [Ame06] and Lerman–Schmidt [LS20]: enable interconnections, but need to be reconciled with HDI and sequential compositions.
Informal Tidbits: Compositions

Parallel compositions.

- The simplest example is a decoupled Cartesian product of systems.
- In mobile agent networks, interconnection may be intermittent.

[Once communication is established, their motion may be coupled...]

Ames [Ame06] and Lerman–Schmidt [LS20]: enable interconnections, but need to be reconciled with HDI and sequential compositions.
Parallel compositions.

- The simplest example is a decoupled Cartesian product of systems.
- In mobile agent networks, interconnection may be intermittent.

[... then decoupled, if the need arises]

Ames [Ame06] and Lerman–Schmidt [LS20]: enable interconnections, but need to be reconciled with HDI and sequential compositions.
Parallel compositions.

- The simplest example is a decoupled Cartesian product of systems.
- In mobile agent networks, interconnection may be intermittent.

[... then decoupled, if the need arises]

Ames [Ame06] and Lerman–Schmidt [LS20]: enable interconnections, but need to be reconciled with HDI and sequential compositions.
Informal Tidbits: Compositions

Parallel compositions.

▶ The simplest example is a decoupled Cartesian product of systems.

▶ In mobile agent networks, interconnection may be intermittent.

[..then decoupled, if the need arises]

Ames [Ame06] and Lerman–Schmidt [LS20]: enable interconnections, but need to be reconciled with HDI and sequential compositions.
Informal Tidbits: Refinement/Coarsening

**Refinement:** Splitting and recombining continuous modes is useful:

- Time as a hybrid system, trajectories as maps of time into a state space.
  
  \[ \rightsquigarrow \text{A central principle in all approaches} \]

- Need *generalized* trajectories to support ill-behaved time subdivisions
  
  \[ \rightsquigarrow \text{b/c mode-transitions are only allowed at jump times!} \]
Coarsening: When is “projection” of a HS to the underlying discontinuous structure more informative?

Methods for bringing topology and hybrid structure into sync?

~ =~ This is precisely what happened to us in [AGK16]!
Moving away from graphs as discrete models of hybrid structure? (a “Conley decomposition”?)

Fixed points are two-dimensional simplices?

Probabilistic aspects of hybrid structure? (Entropy??)

Probability of arrival in $B$ given $A$ or given $C$?
Informal Tidbits: It’s About Time

▶ Generalized Hybrid Time Domains (HDT)?

A smooth “Cantor-themed” curve between two domains...

⇝ MORE admissible solutions!

▶ Reformulate HDTs to facilitate trajectories of this form?

⇝ Another vote in favor of replacing graphs with complexes?

⇝ An opening into measure-theoretic machinery?

▶ Then we need to replace graphs-of-modes with covers-by-modes!
Thank You!


