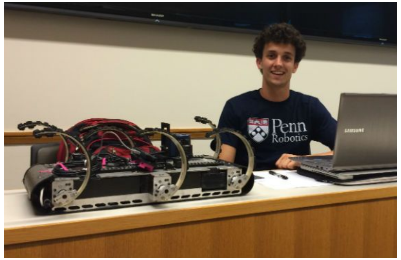
Compositional Robotics Workshop
ICRA 2021, May 31, Xi’An, China

Toward Practicable Hybrid Dynamical Type Theories for Programming Physical Robot Behaviors

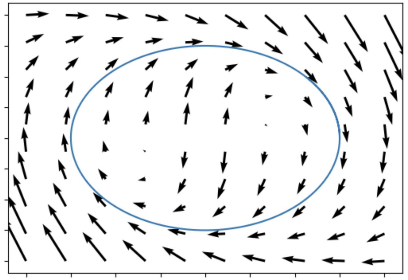
Paul Gustafson (Wright State University)
jww Jared Culbertson (AFRL), Dan Koditschek (Penn), Peter Stiller (TAMU)

Can we make behaviors modular?

Current approach:
Grad student descent



The future:
Physically-grounded programming languages



+

English	Type Theory
True	$\mathbf{1}$
False	$\mathbf{0}$
A and B	$A \times B$
A or B	$A + B$
If A then B	$A \to B$
A if and only if B	$(A \to B) \times (B \to A)$
Not A	$A \to \mathbf{0}$

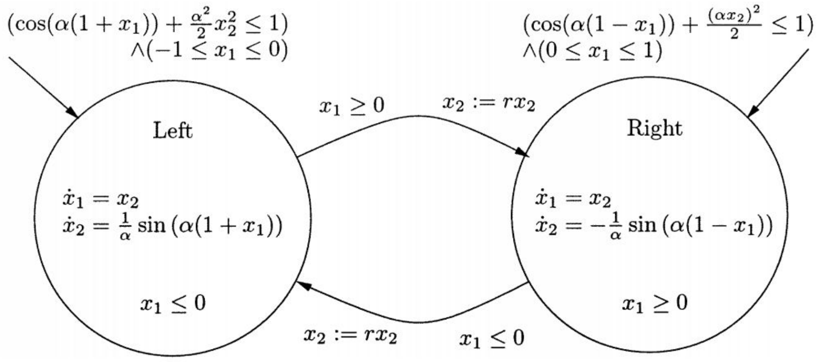
Big Picture

- Composition invariably leads to **categories** (either explicit or implicit)
  - Interfaces <-> objects <-> types
  - Controllers <-> morphisms <-> terms
- How can we encode **parallel**, **hierarchical**, and **sequential compositions** of hybrid systems?
- How can we incorporate **liveness** and **safety** constraints?
- How can we develop **interoperability** with the state-of-the-art linear-time temporal logic (LTL)-based synthesis approaches?

Hybrid systems and semiconjugacies

A **hybrid system** H consists of

- a directed graph $G = (V, E, \mathfrak{s}, \mathfrak{t})$;
- for each **mode** $v \in V$,
  - an **ambient smooth system** (M_v, X_v)
  - an **active set** $I_v \subset M_v$
  - a **flow set** $F_v \subset I_v$
- for each **reset** $e \in E$, a **guard set** $Z_e \subset I_{\mathfrak{s}(e)}$ and an associated **reset map** $r_e \colon Z_e \to I_{\mathfrak{t}(e)}$.



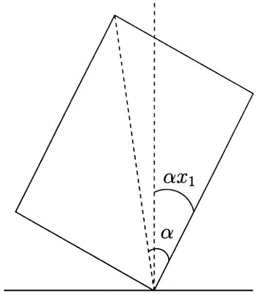
Morphisms: hybrid semiconjugacies

- "execution-preserving maps"

Independent parallel composition

⇕

Categorical product

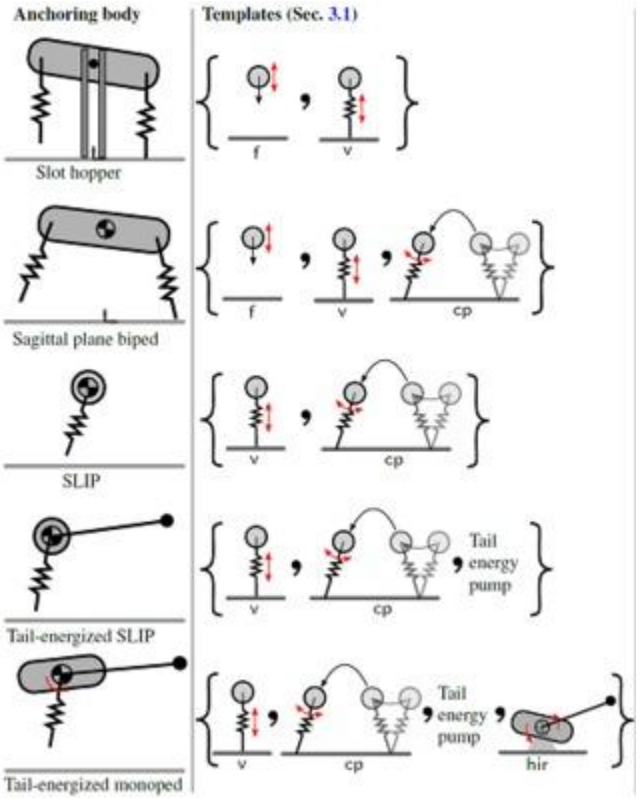


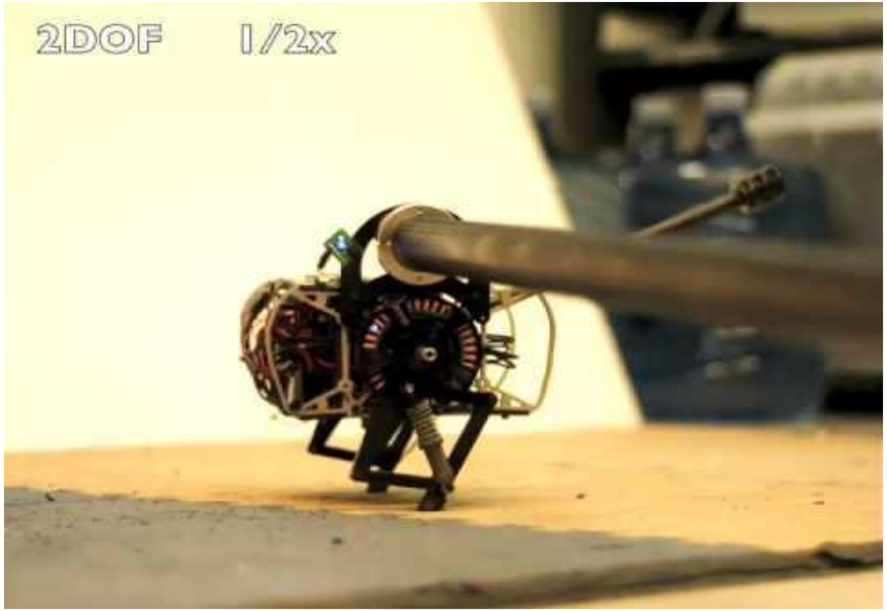
Related work:

- **Lerman**. "A category of hybrid systems." arXiv:1612.01950, 2016.
- **Ames**. "A Categorical Theory of Hybrid Systems." PhD dissertation, Electrical Engineering and Computer Sciences, University of California, Berkeley, 2006.

Image source: Lygeros et al., "Dynamical properties of hybrid automata." IEEE Transactions on automatic control, 2003.

Abstraction via templates and anchors



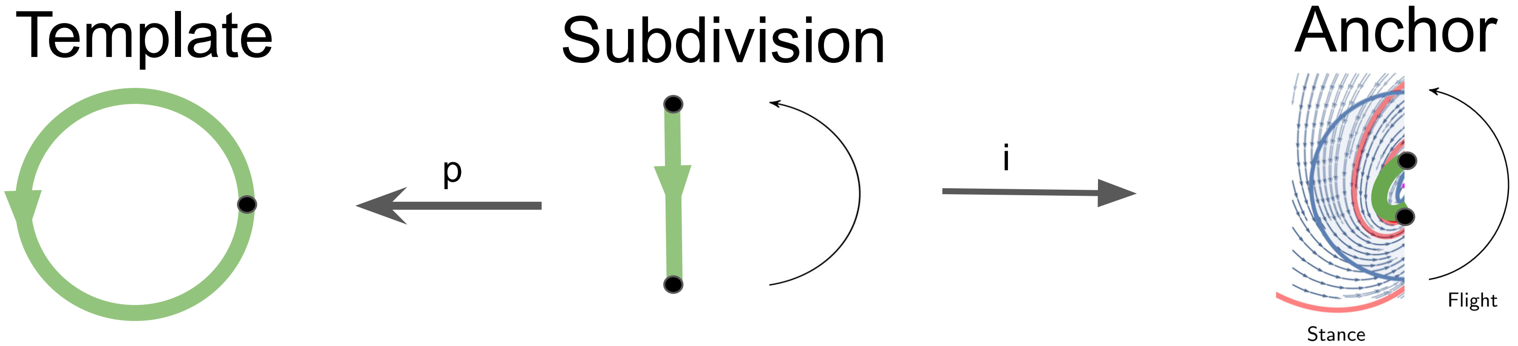


- Full and Koditschek. "Templates and anchors: neuromechanical hypotheses of legged locomotion on land." ***Journal of experimental biology***, 1999.
- De and Koditschek. "Parallel composition of templates for tail-energized planar hopping." **ICRA**, 2015.

Anchoring a limit cycle in a vertical hopper

A **template-anchor pair** is a span $T \xleftarrow{p} S \xrightarrow{i} A$ such that

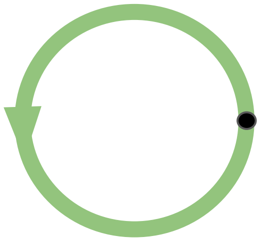
- p is a hybrid subdivision;
- i is a hybrid embedding;
- $i(S)$ is attracting in A.

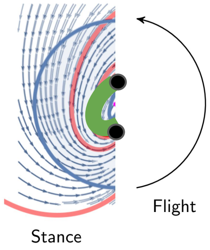


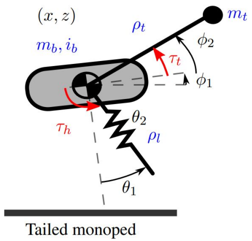
Hierarchical composition

Theorem (CGKS). Template-anchor pairs are weakly associatively composable.

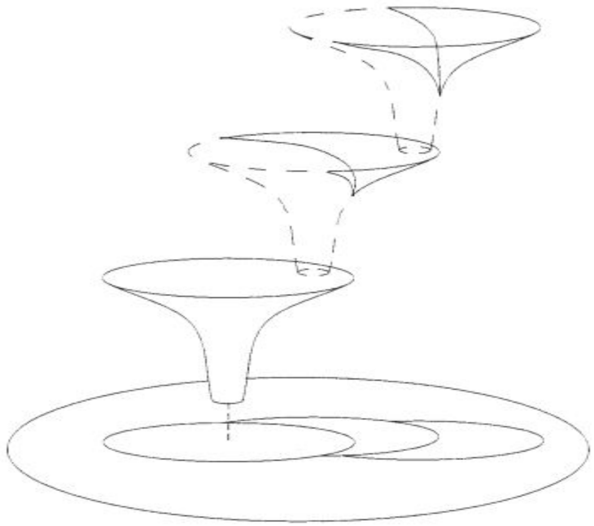
$$
\begin{tikzcd}
& & S = S_1 \times_{A_1} S_2 \arrow[dl, two heads] \arrow[dr, hook] \arrow[ddll, bend right, two heads, "p"'] \arrow[ddrr, bend left, hook, "i"] \arrow[dd, phantom, "\vee", near start] & & \\
& S_1 \arrow[dl, two heads] \arrow[dr, hook] & & S_2 \arrow[dl, two heads] \arrow[dr, hook] & \\
T_1 & & A_1 & & A_2
\end{tikzcd}
$$







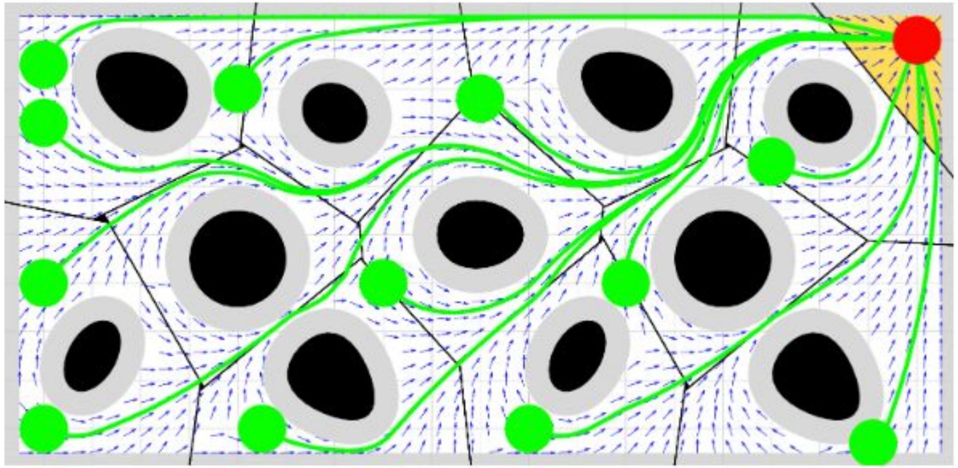
Sequential composition



Burridge, Robert R., Alfred A. Rizzi, and Daniel E. Koditschek. "Sequential composition of dynamically dexterous robot behaviors." *The International Journal of Robotics Research* 18.6 (1999): 534-555.

Goal: define a class of “funnel-like” hybrid systems closed under sequentially composition

Liveness: eventually reach a goal location

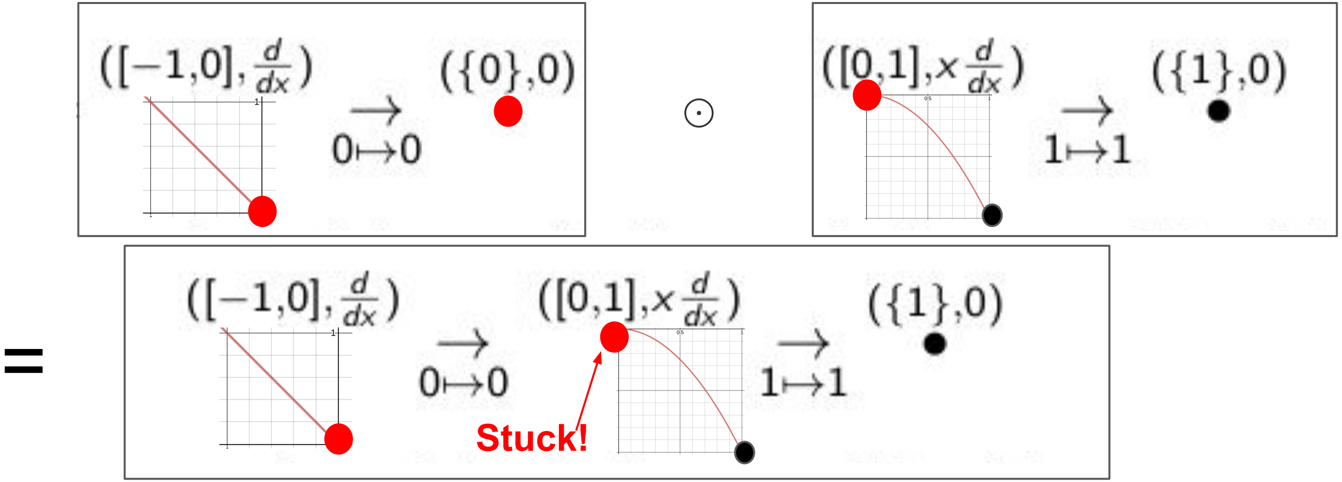


Theorem 3. *The piecewise continuously differentiable "move-to-projected-goal" law in (11) leaves the robot's free space $\mathcal{F}$ (1) positively invariant; and if Assumption 2 holds, then its unique continuously differentiable flow, starting at almost[1] any configuration $\mathrm{x} \in \mathcal{F}$, asymptotically reaches the goal location x^*, while strictly decreasing the squared Euclidean distance to the goal, $\|\mathrm{x} - \mathrm{x}^*\|^2$, along the way.*

Arslan, Omur, and Daniel E. Koditschek. "Sensor-based reactive navigation in unknown convex sphere worlds." *The International Journal of Robotics Research* (2019).

How to define ``funnel-like'' systems?

- **Problem:** the naive measure-theoretic and topologically notions of "almost all" are incompatible with fully general sequential composition
- Example:

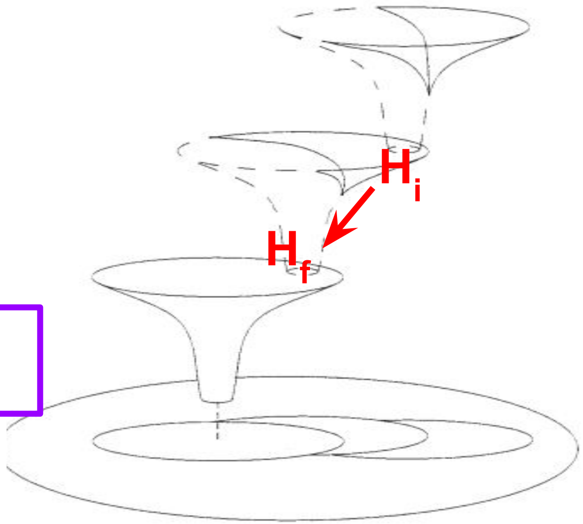


Directed systems

A **directed hybrid system** $H \colon H_i \rightsquigarrow H_f$ is a tuple (H, η_i, η_f) consisting of

- a metric hybrid system H,
- embeddings $\eta_i \colon H_i \to H$ and
- a hybrid embedding $\eta_f \colon H_f \to H$ such that each component $(\eta_f)_v$ is a diffeomorphism, and $G(H_f)$ is a sink in $G(H)$

such that for all $\varepsilon, T > 0$ and $x \in H$, there exists an (ε, T)-**chain** from x to some $y \in H_f$.



Composable!

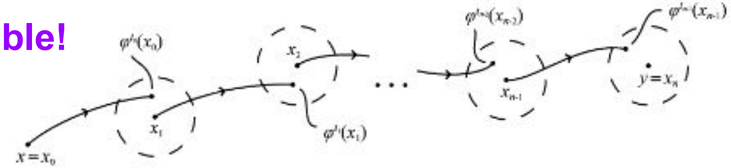
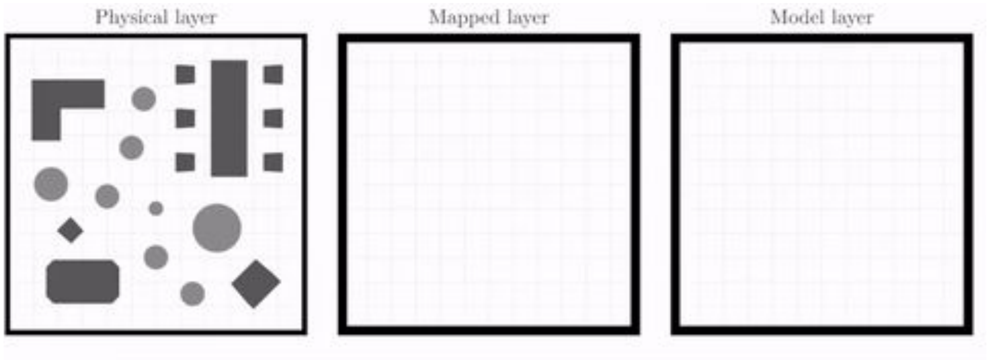


Image source: Alongi and Nelson, *Recurrence and Topology*. AMS, 2007.

A double category of hybrid systems



V. Vasilopoulos, D.E. Koditschek (2018). Reactive Navigation in Partially Known Non-Convex Environments. In WAFR 2018.

Semiconjugacies

$$
\begin{tikzcd}
H_i \arrow[r, "\alpha_i"] \arrow[d, rightsquigarrow, "H"'] \arrow[dr, phantom, "\overset{\alpha}{\Longrightarrow}"] & K_i \arrow[d, rightsquigarrow, "K"] \\
H_f \arrow[r, "\alpha_f"'] & K_f
\end{tikzcd}
$$

Directed systems

Linear dependent type theory

- Dynamic input and output conditions + safety specs
- Linear fragment
  - Manages **states**, **resources**, and **liveness**
  - From symmetric monoidal category of **directed systems** under sequential composition
- Nonlinear fragment
  - Manages sensor-dependent **parameters** and **proofs of safety**
  - Internal language of **presheaves** over the sensorium
    - Example: in this open set of sensor readings, $d(robot, O_i) > \varepsilon$
- **Starting point:** Fu, Kishida, and Selinger. "Linear dependent type theory for quantum programming languages." Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science. 2020.

Navigation example types

$$
\begin{aligned}
go &: (g : X, n : \mathbb{N}) \to Free \otimes (s : See(n)) \multimap (At(g) \otimes See(n)) \oplus Interrupt(s) \\
Interrupt &: See(n) \multimap Free \otimes (NewObs(See(n+1)) \oplus LoseObs(See(n-1)) \oplus TimeStep)) \\
detect &: See(n) \multimap See(n-1) \oplus See(n) \oplus See(n+1) \\
nearestObs &: See(n) \to List(X) \\
projGoal &: ConvHull(n) \to X \to X \\
voronoi &: See(n) \to ConvHull \\
ConvHull &= List(X) \\
Safe &= (s : See(n)) \to d(x, nearestObs(s)) > R \\
controller &: (g : X) \to (c : Free \otimes See(n) \multimap At(g) \otimes (m : \mathbb{N}, See(m)), Safe(c))
\end{aligned}
$$

Semantics of simple types

Type	Template	Presheaf (evaluated at $U \subset B$)
$See(n)$	$(X^n \times \mathbb{R}^n, 0)$	$\lvert\pi_0(f^{-1}([0, M]))\rvert = n$ for all $f \in \pi_{C(S^1, \overline{\mathbb{R}}_{\geq 0})}(U)$
$Free$	$(*, *)$	$\top$
$At(g)$	$(X, \nabla\lVert x - g\rVert^2)$	$\sup_{x \in U} d(x, g) < \epsilon$
$Safe$	$(X, -\sum_i \nabla\lVert x - o_i\rVert^2)$	$\sup_{x \in U, o \in \bigcup_i O_i} d(x, o) > r$

Integration with LTL-based controller synthesis

1. What LTL buys you
    a. **Automatic synthesis**
        i. Kress-Gazit, Fainekos, and Pappas. "Temporal-logic-based reactive mission and motion planning." IEEE transactions on robotics, 2009
    b. Provable safety/finite-time task completion for **particular control systems** using (control) Lyapunov/barrier functions
2. What dependent LL buys you
    a. **Correct-by-construction composition** of subcontrollers
    b. Physical grounding
        i. Extend safe/unsafe sets with **dynamic interfaces** between behaviors
3. Complementary -- embed LTL specs into dependent linear types
    a. Example: “Eventually(Always(g))” becomes “($A \multimap B$) and g(supp(B)) ”
    b. Use synthesized controllers in correct-by-construction composite controllers

Operational semantics

1. No simple notion of abstract machine/lambda calculus for operational semantics
2. Can we define a “gradual” version of operational semantics based on template-anchor hierarchies?
   a. Examples
      i. Anchor At(g) point attractor template in a differential drive robot
      ii. Anchor See(n) template inside navigation + sensing product corresponds to stabilizing sensor readings
   b. Related work: New and Licata. “Call-by-name gradual type theory.” Logical Methods in Computer Science, 2020.

Thanks for listening!