Hierarchical Synthesis of Hybrid Controllers from Temporal Logic Specifications

Georgios E. Fainekos, Antoine Girard and George J. Pappas
Search & Rescue Scenario

UAVs search for accidents. If accident is found, then they summon the ambulance.

\[ \dot{z}(t) = v(t), \quad \|v(t)\| \leq v \]
\[ y'(t) = z(t) \]
\[ z(t) \in \mathbb{R}^2, \quad z(0) = x_0, \quad y'(t) \in \mathbb{R}^2 \]

Search & Rescue Scenario

Two issues:

1) How can you construct robust trajectories wrt the temporal logic specifications?

2) How can we handle complex environments with complicated specifications beyond kinematics models?
Hierarchical Modeling of Control Systems

\[
\begin{align*}
S_N & \quad \text{Rough model} \\
S_k & \quad \text{Intermediate model} \\
S_0 & \quad \text{Fine model}
\end{align*}
\]

\[
\begin{align*}
\dot{z}(t) &= g(z(t),v(t)), \quad v(t) \in V \\
y'(t) &= k(z(t)) \\
z(t) &\in \mathbb{R}^m, \quad z(0) = z_0, \quad y'(t) \in \mathbb{R}^p
\end{align*}
\]

\[
\begin{align*}
\dot{x}(t) &= f(x(t),u(t)), \quad u(t) \in U \\
y(t) &= h(x(t)) \\
x(t) &\in \mathbb{R}^n, \quad x(0) = x_0, \quad y(t) \in \mathbb{R}^p
\end{align*}
\]
\[ \ddot{x}(t) = u(t), \quad \|u(t)\| \leq \mu \]
\[ y(t) = x(t) \]

\( x(t) \in \mathbb{R}^2, \ x(0) \in X_0, \dot{x}(0) = 0, \ y(t) \in \mathbb{R}^2. \)
Task: “Stay always in $\pi_0$ and visit area $\pi_2$, then area $\pi_3$, then area $\pi_4$ and, finally, return to and stay in region $\pi_1$, while avoiding areas $\pi_2$ and $\pi_3$”

$$\text{RTL: } \square \pi_0 \land (\pi_2 \land (\pi_3 \land (\pi_4 \land (\neg \pi_2 \land \neg \pi_3) \cup \square \pi_1)))$$
Overview of solution

Agile robot

\[ \dot{x}(t) = u(t), \quad \left\| u(t) \right\| \leq \mu \]
\[ y(t) = x(t) \]
\[ x(t) \in \mathbb{R}^2, \quad x(0) = x_0, \quad \dot{x}(0) = 0, \quad y(t) \in \mathbb{R}^2 \]
\[ \phi \in RTL \]


Sluggish robot

\[ \dot{z}(t) = v(t), \quad \left\| v(t) \right\| \leq v \]
\[ y'(t) = z(t) \]
\[ z(t) \in \mathbb{R}^2, \quad z(0) = x_0, \quad y'(t) \in \mathbb{R}^2 \]
\[ \phi' \in RTL \]

Abstraction

\[ \phi' \]

Refinement

Girard, Pappas

CDC 2005, CDC 2006, Automatica

Fainekos, Kress-Gazit, Pappas

ICRA 2005, CDC 2005, CDC 2006

Kress-Gazit, Fainekos, Pappas

ICRA 2007
\[
\dot{z}(t) = g(z(t), v(t)), \quad v(t) \in V \\
y'(t) = k(z(t)) \\
\]

\[
\dot{x}(t) = f(x(t), u(t)), \quad u(t) \in U \\
y(t) = h(x(t)) \\
\]

Approximate Simulation Relation

\[
\dot{x}(t) = f(x(t), u(t)), \quad u(t) \in U \\
y(t) = h(x(t)) \\
x(t) \in \mathbb{R}^n, \quad x(0) = x_0, \quad y(t) \in \mathbb{R}^p
\]

\[
\dot{z}(t) = g(z(t), v(t)), \quad v(t) \in V \\
y'(t) = k(z(t)) \\
z(t) \in \mathbb{R}^m, \quad z(0) = z_0, \quad y'(t) \in \mathbb{R}^p
\]

\[
\|y'(t) - y(t)\| \leq \delta
\]

Exact simulation

Approximate simulation
Approximate Simulation Relation

A relation $R \subseteq \mathbb{R}^m \times \mathbb{R}^n$ is a $\delta$-approximate simulation if for all $(z,x) \in R$,

1. $\|k(z) - h(x)\| \leq \delta$

2. for all $T \geq 0$, for all trajectories $z(t)$ of $S'$ such that $z(0) = z_0$, there exists a trajectory $x(t)$ of $S$ such that $x(0) = x_0$ satisfying:

   $\forall t \in [0, T], (z(t), x(t)) \in R.$

If initially the states of $S$ and $S'$ are in $R$ (i.e. $(z_0, x_0) \in R$), then

Any observed trajectory of $S'$ has an observed trajectory of $S$ in its $\delta$-neighborhood.
Simulation Functions

\[ \dot{x}(t) = f(x(t), u(t)), \quad u(t) \in U \]
\[ y(t) = h(x(t)) \]
\[ x(t) \in \mathbb{R}^n, \quad x(0) = x_0, \quad y(t) \in \mathbb{R}^p \]

\[ \dot{z}(t) = g(z(t), v(t)), \quad v(t) \in V \]
\[ y'(t) = k(z(t)) \]
\[ z(t) \in \mathbb{R}^m, \quad z(0) = z_0, \quad y'(t) \in \mathbb{R}^p \]

- We can construct approximate simulation relations using simulation functions.
- A simulation function \( W: \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^+ \) satisfies:

\[
W(z, x) \geq \left\| k(z) - h(x) \right\|^2 \\
\sup_{v \in V} \inf_{u \in U} \left( \frac{\partial W(z, x)}{\partial z} g(z, v) + \frac{\partial W(z, x)}{\partial x} f(x, u) \right) \leq 0
\]

- Effective characterization of approximate simulation relations

\[ R = \left\{ (z, x) \mid W(z, x) \leq \delta^2 \right\} \] is an approximate simulation relation of precision \( \delta \).
Abstraction

• Let $S'$ be an abstraction of $S$ such that there is simulation function $W$.
• Let $\delta^2 = W(z_0, x_0)$, then

Any observed trajectory of $S'$ has an observed trajectory of $S$ in its $\delta$-neighborhood.
Hybrid Controller

\[ \dot{z}(t) = g(z(t), v(t)), \quad v(t) \in V \]

\[ y'(t) = k(z(t)) \]

\[ S' \]

Interface

\[ \dot{u}(t) = f(x(t), u(t)), \quad u(t) \in U \]

\[ y(t) = h(x(t)) \]

\[ S \]

\[ y'(t) = \varphi' \]

\[ y(t) = \varphi \]
Linear Temporal Logic: Continuous Time

Like propositional logic, but also reasoning wrt time ...

**Basic operators:**
Propositional: $\land, \lor, \neg$
Temporal: $\Diamond, \Box, U, R$

- $\Diamond$(red)
  - Eventually red

- $\Box$(gray)
  - Always gray

- (gray) U (red)
  - gray Until red
LTL with continuous time semantics (RTL)

Let $\Pi$ be a set of atomic propositions.

Define the denotation $[[.]] : \Pi \to P(X)$.

Syntax in NNF: $\phi ::= \pi | \phi_1 \land \phi_2 | \phi_1 \lor \phi_2 | \phi_1 U \phi_2 | \phi_1 R \phi_2$

Semantics:

$\langle y, [[.]] \rangle \models \pi$ iff $y(0) \in [[\pi]]$
$\langle y, [[.]] \rangle \models \phi_1 \land \phi_2$ iff $\langle y, [[.]] \rangle \models \phi_1$ and $\langle y, [[.]] \rangle \models \phi_2$
$\langle y, [[.]] \rangle \models \phi_1 \lor \phi_2$ iff $\langle y, [[.]] \rangle \models \phi_1$ or $\langle y, [[.]] \rangle \models \phi_2$
$\langle y, [[.]] \rangle \models \phi_1 U \phi_2$ iff $\exists t \geq 0$ s.t. $\langle y|_t, [[.]] \rangle \models \phi_2$ and $\forall s \in [0, t]$ $\langle y|_s, [[.]] \rangle \models \phi_1$
$\langle y, [[.]] \rangle \models \phi_1 R \phi_2$ iff $\forall t \geq 0$ $\langle y|_t, [[.]] \rangle \models \phi_2$ or $\exists s \in [0, t]$ s.t. $\langle y|_s, [[.]] \rangle \models \phi_1$

Some notation: $y|_t(s) = y(t+s)$ $\Diamond \phi = T U \phi$ $\Box \phi = F R \phi$
Contraction - Expansion

For $\delta > 0$, the $\delta$-ball centered at $\alpha \in A$ is defined as:

$$B_\delta(\alpha) = \{ \beta \in A : \| \alpha - \beta \| \leq \delta \}$$

For $\Gamma \subseteq A$, the contraction and expansion are defined as:

$$C_\delta(\Gamma) = \{ \alpha \in A : B_\delta(\alpha) \subseteq \Gamma \}$$
$$B_\delta(\Gamma) = \{ \alpha \in A : B_\delta(\alpha) \cap \Gamma \neq \emptyset \}$$
“Robustifying” an RTL formula

Define a translation function

\[ \text{rob} : \Pi \rightarrow \Xi_{\Pi} \]

where

\[ \Xi_{\Pi} = \{ \xi_{\alpha} | \alpha = \pi \text{ or } \alpha = \neg \pi \text{ for } \pi \in \Pi \} \]

Define new map \([.]_\delta : \Xi_{\Pi} \rightarrow P(X)\)

\[
[\xi]_\delta = \begin{cases} 
C_\delta([\pi])^c & \text{if } \xi = \xi_{-\pi} \\
C_\delta([\pi]) & \text{if } \xi = \xi_{\pi}
\end{cases}
\]

RTL:

\[
\square \xi_{\pi_0} \land \diamond (\xi_{\pi_2} \land (\xi_{\pi_3} \land (\xi_{\pi_4} \land (\xi_{-\pi_2} \land \xi_{-\pi_3}) \cup \square \xi_{\pi_1})))
\]
Main Result

**Theorem:** Consider an RTL formula \( \varphi \), a map \([.]: \Pi \rightarrow P(X) \) and a number \( \delta > 0 \), then for all functions \( y(t), y'(t) \) from \( R^+ \) to \( R^p \) such that for all \( t \geq 0 \),

\[
||y(t) - y'(t)|| \leq \delta
\]

it is

\[
(y', [.]) \in \text{rob} (\varphi) \quad \Rightarrow \quad (y, [.]) \in \varphi
\]
Back to our simple example

[Back to our simple example]

[Back to our simple example]
Hybrid Controller

\[ \dot{z}(t) = g(z(t), v(t)), \quad v(t) \in V \]

\[ y'(t) = k(z(t)) \]

\[ y'(t) = \varphi' \]

A hybrid controller is a tuple

\[ H' = (Z, V, L, E, \text{Inv}, \text{Out}, \text{Init}, \text{Guard}) \]

where

- \( Z \) is the state space of the system \( S' \)
- \( L \) is the set of control locations,
- \( E \subseteq L \times L \) is the set of control switches
- \( \text{Inv} : L \to P(Z) \) assigns an invariant set to each location
- \( \text{Out} : L \times Z \to V \) is the control input for \( S' \)
- \( \text{Init} : L \to P(Z) \) assigns to each control location a set of initial conditions
- \( \text{Guard} : E \to P(Z) \) is the guard condition that enables a control switch \( e \) in \( E \)

\[
\dot{z}(t) = g(z(t), v(t)), \quad v(t) \in V
\]

\[ y'(t) = k(z(t)) \]
Each control location
How can you design such controllers?

[Conner et. al.] “Composition of local potential functions for global robot control and navigation”, ICRA 2003
Main idea: Use a $C^2$ diffeomorphism from the simplex to the unit disk and then construct a potential function free of local minima.

$$u = - D_q r^T$$

[Belta, Habets], “Constructing decidable hybrid systems with velocity bounds”, CDC 2004
Main idea: Affine functions on simplexes are uniquely defined on vertices. The set of all controllers can be parameterized by the values on the vertices.

$$u = Ax + b$$

[Lindemann et. al.], “Real time feedback control for nonholonomic mobile robots with obstacles”, CDC 2006
Main idea: Perform GVD decomposition in each convex cell and design smooth vector fields.
Back to the toy example ...

\[ \text{RTL: } \Box \xi_{\pi 0} \land \Box (\xi_{\pi 3} \land \Box (\xi_{\pi 4} \land (\xi_{\pi 2} \land \xi_{\pi 3}) \cup \Box \xi_{\pi 1}))) \]
Back to the toy example ...

\[ \delta = 2 \]
Refinement

\[ \dot{x}(t) = f(x(t), u(t)), \quad u(t) \in U \]
\[ y(t) = h(x(t)) \]

\[ x(t) \in \mathbb{R}^n, \quad x(0) = x_0, \quad y(t) \in \mathbb{R}^p \]

\[ \dot{z}(t) = g(z(t), v(t)), \quad v(t) \in V \]
\[ y'(t) = k(z(t)) \]

\[ z(t) \in \mathbb{R}^m, \quad z(0) = z_0, \quad y'(t) \in \mathbb{R}^p \]

• Can we synthesize \( u(.) \) from the input \( v(.) \) ?

• Interface \( u_W : \mathbb{R}^m \times \mathbb{R}^n \times V \rightarrow U \) associated with simulation function \( W \):

\[
W(z, x) \geq \|k(z) - h(x)\|^2
\]

\[
\sup_{v \in V} \left( \frac{\partial W(z, x)}{\partial z} g(z, v) + \frac{\partial W(z, x)}{\partial x} f(x, u_W(z, x, v)) \right) \leq 0
\]
Back to our simple example

\[ \dot{x}(t) = u(t), \|u(t)\| \leq \mu \]
\[ y(t) = x(t) \]

\[ x(t) \in \mathbb{R}^2, \ x(0) = x_0, \ \dot{x}(0) = 0, \ y(t) \in \mathbb{R}^2 \]

\[ \dot{z}(t) = v(t), \|v(t)\| \leq \nu \]
\[ y'(t) = z(t) \]

\[ z(t) \in \mathbb{R}^2, \ z(0) = x_0, \ y'(t) \in \mathbb{R}^2 \]

- Simulation function and associated interface:

\[ W(z, x, \dot{x}) = \max \left( \|z - x\|^2 + \alpha \|z - x - 2 \dot{x}\|^2, 4\nu^2 \right) \]

\[ u_w(z, x, \dot{x}, v) = \frac{v}{2} + \frac{1 + \alpha}{4\alpha} (z - x) - \dot{x} \]

- Condition:

\[ \frac{v}{2} \left( 1 + \left| 1 - \frac{1}{\alpha} \right| + \frac{2}{\sqrt{\alpha}} \right) \leq \mu \]

Then \( R \) is an approximate simulation relation of precision \( 2\nu \) of \( S' \) by \( S \).
\[ \dot{z}(t) = g(z(t), v(t)), \quad v(t) \in V \]
\[ y'(t) = k(z(t)) \]

\[ y'(t) = \varphi' \]

\[ y(t) = \varphi \]

\[ \dot{x}(t) = f(x(t), u(t)), \quad u(t) \in U \]
\[ y(t) = h(x(t)) \]
Final Motion for $\nu=1$ (or $\delta=2$)
Example for \( \nu = 2.5 \) (or \( \delta = 5 \))
If the abstract system is too fast ...
Conclusions / Future work

- Hierarchical framework for the synthesis of HA from temporal logic
  - Hierarchical controller
  - Approximate simulation relations
  - Robustness of temporal logic formulas
  - Takes advantage of existing methods for the design of HA from TL
  - Can provide solutions for complicated specifications, for systems with complicated continuous dynamics that operate in complicated environments

- Compute interfaces for other systems
- Design for distributed specifications and for open systems
- Make a public distribution of the prototype toolbox
Thank You!

Any Question(s) ?

Where do you want to go today???