Membership-based Synthesis of Linear Hybrid Automata

Christian Schilling

July 16, 2019, CAV

joint work with
Miriam García Soto, Thomas A. Henzinger, Luka Zeleznik
Model synthesis from data
Model synthesis from data

Data → Model
Adaptive synthesis algorithm

Data

Model

Improved model
Overview

Introduction

Preliminaries

Synthesis

Evaluation

Summary
We consider non deterministic linear hybrid automata

The LHA features piecewise-linear executions

- Nondeterministic mode changes
- Restriction in this work: continuous executions

Linear hybrid automaton (LHA) model
We consider non-deterministic linear hybrid automata. The LHA features piecewise-linear executions.

\[
\begin{align*}
q_1 & : \dot{x} = 1, \quad x \in [2, 3] \\
q_2 & : \dot{x} = 0, \quad x \in [0, 3] \\
q_3 & : \dot{x} = -1, \quad x \in [0, 3]
\end{align*}
\]
Linear hybrid automaton (LHA) model

- Nondeterministic mode changes
- Restriction in this work: continuous executions
Piecewise-linear (PWL) function

\[ f(t) \]

\[ X \]

\[ t \]
Definition. $f : [0, T] \rightarrow \mathbb{R}^n$ is an $m$-piecewise-linear (m-PWL) function if $f \equiv p_1, p_2, \ldots, p_m$ sequence of $m$ affine pieces of the form $p_i(t) = a_i t + b_i$, where:

- $a_i$ is the slope ($p_i$)
- $b_i$ is the initial value
- $f(t) = p_i(t)$ for $t \in \text{dom}(p_i)$
- $f$ is continuous

$x$ $\uparrow$ $\downarrow$ $t$

$p_1$ $p_2$ $p_3$ $p_4$ $f$

$0$ $t_1$ $t_2$ $t_3$ $T$
Related work

- SARX models (discrete time, deterministic switching) and PWARX models (SARX with state-space partition) can be synthesized algebraically\(^1\); some adaptive algorithms exist\(^2,^3\)

- Existing approaches for hybrid automata are not adaptive and come with limitations (e.g., periodic\(^4\), acyclic\(^5\), stateless\(^6\), deterministic\(^7\))

---

\(^1\) S. Paoletti et al. *Eur. J. Control* (2007).

\(^2\) A. Skeppstedt et al. *Int. J. Control* (1992).

\(^3\) Y. Hashambhoy and R. Vidal. *CDC*. 2005.

\(^4\) R. Grosu et al. *HSCC*. 2007.

\(^5\) O. Niggemann et al. *AAAI*. 2012.

\(^6\) D. L. Ly and H. Lipson. *JMLR* (2012).

\(^7\) I. Lamrani et al. *ICPS*. 2018.
| Introduction | Preliminaries | Synthesis | Evaluation | Summary |
|--------------|---------------|-----------|------------|---------|
|              |               |           |            |         |

**Overview**

Introduction

Preliminaries

Synthesis

Evaluation

Summary
Synthesis problem

Given a finite set of PWL functions $F$ and a value $\varepsilon \in \mathbb{R} \geq 0$, construct an LHA $H$ that $\varepsilon$-captures all $f \in F$.

$\varepsilon$-capturing

An LHA $\mathcal{H}$ $\varepsilon$-captures a PWL functions $f$ if there exists an execution $\sigma$ with $d(f, \sigma) \leq \varepsilon$.

Synthesis problem

Given a finite set of PWL functions $\mathcal{F}$ and $\varepsilon \in \mathbb{R}_{\geq 0}$, construct an LHA $\mathcal{H}$ that $\varepsilon$-captures all $f \in \mathcal{F}$. 

$d(f, \sigma) = \max_{t \in [0,T]} |f(t) - \sigma(t)|$
Synchronous switching

- Execution $\sigma$ must switch synchronously with PWL function $f$
Synchronous switching

- Execution $\sigma$ must switch synchronously with PWL function $f$
Synchronous switching

- Execution $\sigma$ must switch synchronously with PWL function $f$

- Reduction to satisfiability of linear-arithmetic formula
  - Parametric in number of modes (i.e., can be minimized)
Asynchronous switching

- Execution $\sigma$ must switch in intervals close to PWL function $f$
Asynchronous switching

- Execution $\sigma$ must switch in intervals close to PWL function $f$
Asynchronous switching

- Execution $\sigma$ must switch in intervals close to PWL function $f$
Asynchronous switching

- Execution $\sigma$ must switch in intervals close to PWL function $f$

- Counterexample-guided, based on membership test
Recall: Adaptive synthesis algorithm

Data → Improved model

Model
**Membership algorithm**

- **Flow**($q_1$) and **Flow**($q_2$)

- Example: one step along path with prefix $q_1q_2$
Membership algorithm

Flow($q_1$)  Flow($q_2$)

$A_{aux} = \text{POST}(P, G_1, \text{Flow}(q_1))$

$A = \text{PRE}(X_1, A_{aux}, \text{Flow}(q_2))$

$X_1 \subseteq G_1 \cap G_2 \subseteq X_2$
Membership algorithm

Flow(q₁)  Flow(q₂)

Aaux = POST (P, G₁, Flow(q₁))
A = PRE (X₁, Aaux, Flow(q₂))

G

P

X

0

t
Membership algorithm

Flow(q₁)  Flow(q₂)

G

P

Baux

X

0

F₁

F₂

X₁

X₂

0

t
Membership algorithm

Flow\( (q_1) \)  \hspace{1cm}  \text{Flow\( (q_2) \)}

\[ \text{Flow}(q_1) \]

\[ \text{Flow}(q_2) \]
Membership algorithm

\[ P = A \cup B \]

- PWL function is \( \varepsilon \)-captured along path iff last set is nonempty
Overview

Introduction

Preliminaries

Synthesis

Evaluation

Summary
**Synthetic model replication**

\[
\begin{align*}
\dot{x} &= 2, & x &\in [0, 10] \\
x &= \in [-0.10, 9.87] \\
\dot{x} &= -1, & x &\in [0, 10] \\
\dot{x} &= -2, & x &\in [0, 10] \\
x &= \in [0, 10] \\
\dot{x} &= 1, & x &\in [0, 10] \\
\end{align*}
\]

**ε = 0.2**

**Evaluation**
Voltage traces of excitable cell

\[ \dot{x} = 0.00 \quad x \in [-76.04, -73.92] \]

\[ \dot{x} = 130.02 \quad x \in [-76.04, 46.02] \]

\[ \dot{x} = -2.13 \quad x \in [-76.04, -4.00] \]

\[ \dot{x} = -0.76 \quad x \in [-6.05, 36.02] \]

\[ \dot{x} = -1.52 \quad x \in [33.79, 46.02] \]

Sample input traces

Sample executions
Summary

- Automatic synthesis of linear hybrid automaton from piecewise-linear functions
- Trade-off parameter $\varepsilon$ (model size vs. model precision)
- Model with synchronous switching
  - Reduction to linear arithmetic
  - Minimal number of modes
- Model with asynchronous switching
  - Adaptive algorithm
  - Based on membership/reachability queries
  - Sound and complete for a general class of LHA