Model-bounded monitoring of hybrid systems

Masaki Waga¹, Étienne André², Ichiro Hasuo³

Kyoto University¹, Université de Lorraine², National Institute of Informatics³

18 May 2021, MT-CPS 2021

This work is partially supported by JST ACT-X Grant No. JPMJAX200U, by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), by JSPS Grant-in-Aid No. 18J22498, and by ANR-NRF ProMiS (ANR-19-CE25-0015).
Safety Critical CPSs

Self-driving car crash in Arizona: Red light runner hits Waymo van

Tesla Model 3: Autopilot engaged during fatal crash

Monitoring

**Specification:** No \((v > 120)\)
Monitoring

Specification: No \((v > 120)\)
Monitoring with Sampling

**Specification:** No \((v > 120)\)
Monitoring with Sampling

**Specification:** No \((v > 120)\)
Monitoring with Sampling

**Specification:** No \((v > 120)\)
Signal Interpolation

Specification: No \((\nu > 120)\)
Signal Interpolation

Specification: No \( (v > 120) \)
Signal Interpolation

**Specification:** No \((v > 120)\)
Signal Interpolation

**Specification:** No \((\nu > 120)\)
Signal Interpolation

Specification: No \((v > 120)\)
Specification: No  \( (v > 120) \)

Impossible because
\[ \left| \frac{dv}{dt} \right| < K \]
Model-Bounded Monitoring

**Specification:** No \( (v > 120) \)

**Knowledge (bounding model)**

\[ \left| \frac{dv}{dt} \right| < K \]

[Graph showing velocity over time with a bounding line at 120]
Model-Bounded Monitoring

**Specification:** No \((v > 120)\)

**Knowledge**
(bounding model)
\[ \left| \frac{dv}{dt} \right| < K \]

**Feasible execution with**
\[ \left| \frac{dv}{dt} \right| < K \]
Model-Bounded Monitoring

**Specification:** No \( (v > 120) \)

**Knowledge** (bounding model)
\[
\left| \frac{dv}{dt} \right| < K
\]
Model-Bounded Monitoring

**Specification:** No \( (v > 120) \)

**Knowledge** (bounding model)

\[
\left| \frac{dv}{dt} \right| < K
\]

Feasible execution with

\[
\left| \frac{dv}{dt} \right| < K
\]
Q. How to Represent Bounding Model?

A. Linear Hybrid Automata
Contributions

• Proposed model-bounded monitoring
  Bounding model (knowledge): linear HAs $\mathcal{M}$

• Formalized with monitored language $L_{\text{mon}}(\mathcal{M})$
  $L_{\text{mon}}(\mathcal{M})$: possible discrete observations of $\mathcal{M}$

• Algorithms + implementations
  Idea: bounded-time reachability
  Experiment $\rightarrow$ effectively monitorable
Model-Bounded Monitoring

Given

- Bounding model in LHA $M$
- Safety Specification $\varphi$
- Discrete Log $w$

Decide if the actual behavior might violate the spec.

Our Contribution

| $\frac{dv}{dt}$ | $< K$ |

No $(v > 120)$
Model-Bounded Monitoring

Given

- Bounding model in LHA $M$
- Safety Specification $\varphi$
- Discrete Log $w$

Decide if the actual behavior might violate the spec.
Monitored Language \( L_{\text{mon}} \)

Our Contribution

Combine cont. exec. of \( \mathcal{M} \) and disc. obs. of \( w \)

\[
L_{\text{mon}}(\mathcal{M}) = \{ \text{Discr. Obs } w \mid \}
\]
Monitored Language $L_{mon}$

Our Contribution

Combine cont. exec. of $M$ and disc. obs. of $w$

$$L_{mon}(M) = \{ \text{Discr. Obs } w \mid \exists \text{ exec. } \sigma \text{ of } M \text{ s.t.}$$

\[ t \]

\[ y \]

\[ 120 \]
Monitored Language $L_{mon}$

Our Contribution

Combine cont. exec. of $M$ and disc. obs. of $w$

\[ L_{mon}(M) = \{ \text{Discr. Obs } w \mid \exists \text{ exec. } \sigma \text{ of } M \text{ s.t. } \]
\[ w \text{ is a sample of } \sigma \} \]
Workflow of Model-bounded Monitoring

1. Construct an LHA $M_{\neg \varphi}$ from bounding model $M$ and spec. $\varphi$
   
   **Idea:** Product of LHAs

2. Check if $w \in L_{\text{mon}}(M_{\neg \varphi})$
   
   **Idea:** Bounded-time reachability analysis
Algorithm: Bounded-time Reachability
Algorithm: Bounded-time Reachability
Algorithm: Bounded-time Reachability
Algorithm: Bounded-time Reachability
Algorithm: Bounded-time Reachability
Algorithm: Bounded-time Reachability
Algorithm: Bounded-time Reachability

Feasible execution with \[
\left| \frac{dv}{dt} \right| < K
\]
Algorithm: Bounded-time Reachability

Feasible execution with \[ \left| \frac{dv}{dt} \right| < K \]

bounded
Implementations

**Approach 1**: Utilize existing model-checker (PHAVerLite)

Pros: Highly-optimized reachability analysis impl.

**Approach 2**: Implement dedicated monitor (HAMoni)

Pros: Best performance in theory
Environment of Experiments

- Used 3 benchmarks on adaptive cruise controller (ACC) + 1 robot navigation (NAV) benchmark

  - **ACC**: Cars should not be too close (or no physical contact)
  - **NAV**: Do not enter an unsafe region

- Amazon EC2 c4.large instance / Ubuntu 18.04 LTS (64 bit)
  - 2.9 GHz Intel Xeon E5-2666 v3, 2 vCPUs, 3.75 GiB RAM
Experiment Results
Changing Observation Length

Dedicated impl. ≈ 10x faster

> 5000 samples / sec.
Experiment Results
Changing Model Dimension

![Graph showing execution time vs dimension for PHAVerLite and HAMoni with different values of ε.]

- PHAVerLite, len. 100, ε = 2.0
- PHAVerLite, len. 100, ε = 0.9
- HAMoni, len. 100, ε = 2.0
- HAMoni, len. 100, ε = 0.9

Existing tool was faster for dim. > 6

Future work: further optimization
Experiment Results

False Alarms

False alarm for “very safe” exec. → sampling is coarse
Conclusions

• Proposed model-bounded monitoring
  Bounding model (knowledge): linear HAs \( \mathcal{M} \)

• Formalized with monitored language \( L_{mon}(\mathcal{M}) \)
  \( L_{mon}(\mathcal{M}) \): possible discrete observations of \( \mathcal{M} \)

• Algorithms + implementations
  Idea: bounded-time reachability
  Experiment \( \rightarrow \) effectively monitorable