Monitoring Cyber Physical Systems in a Timely Manner

A. Prasad Sistla,

joint work with M. Zefran, Y Feng, and Y Ben

University of Illinois at Chicago

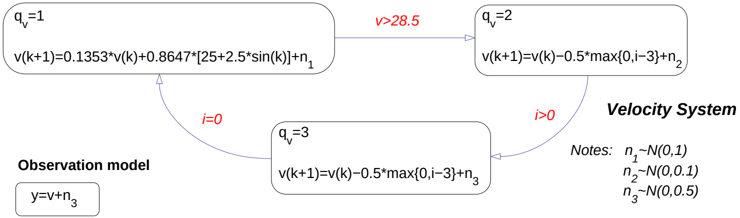
Ensuring Correctness

- Cyber Physical Systems (CPS): hybrid states
  - e.g. automotive systems, robotic systems etc.
- Correctness: hard to achieve
- Testing: not exhaustive
- Thorough Verification:
  - Not always feasible due to complexity
  - Source code may not be available
  - Assumptions made may not hold at run time
- Monitoring: a Complementary Approach
  - Provides additional level of safety;
  - Monitor takes outputs of the systems,
    Checks if the system computation is correct.

Probabilistic Systems

- System behavior probabilistic due to
  - Noise in the sensors etc.
  - Other uncertainties (e.g., failures)
- System state is only partially observable

Example: A Train Velocity and Braking System modeled with Prob. Hybrid Automata



System Modeling

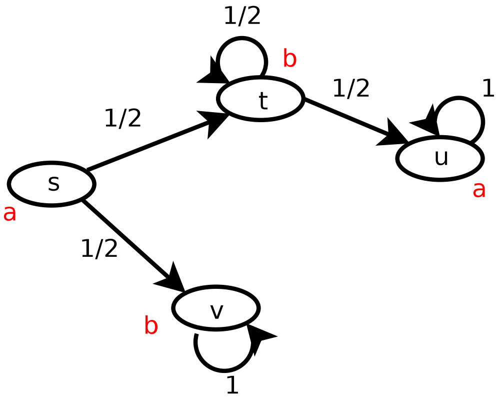
Two Approaches:

- Discretize the state space and model it as a Hidden Markov Chain (HMC)
- Use Extended HMC with hybrid states

Hidden Markov Chains (HMC)

- A HMC $H = (G, O, r_0)$ where
  - $G = (S, R, \phi)$ is a Markov chain;
    - S - countable set of states
    - $R \subseteq S \times S$ - transition relation
    - $\phi : R \rightarrow (0, 1]$ assigns probabilities to transitions
  - $O : S \rightarrow \Sigma$ where Σ is a countable set of outputs;
  - $r_0 \in S$ is the start state
- Define Prob. $\mathcal{F}_{G,s}$ on measurable sets of *state* sequences,
- Prob. $\mathcal{F}_{H,s}$ on measurable sets of *output* sequences.

A HMC Example



"$\Diamond\, v$" denotes paths in which v appears eventually.

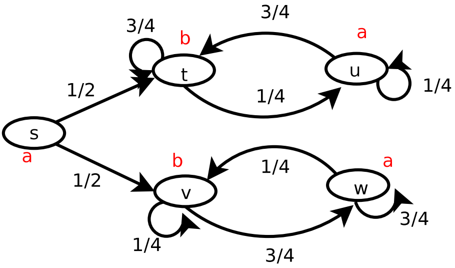
$$\mathcal{F}_{G,s}(\Diamond\, v) = \frac{1}{2}$$
$$\mathcal{F}_{H,s}(\Box\Diamond\, b) = \frac{1}{2}$$

Given a HMC H, a property automaton $\mathcal{A}$ and a monitor M, which observes outputs at runtime and raises alarms,

- Acceptance Accuracy (AA) is Prob. a good computation is accepted by M. 1-AA: false alarms.
- Rejection Accuracy (RA) is Prob. a bad computation is rejected by M. 1-RA: missed alarms

H is Monitorable w.r.t. $\mathcal{A}$, if $AA \to 1$ and $RA \to 1$ are achievable.

E.g. H is monitorable, w.r.t. $\Diamond v$



Conservative Threshold Monitors

- Given *H* and $\mathcal{A}$, a Threshold Monitor *M* at runtime acts as follows:

  1. After the system outputs sequence α, *M* estimates the cond. prob. $AccPr(\alpha)$ that the computation generating α is correct;

  2. If $AccPr(\alpha) < atr$, raises an alarm.

- Every "bad" computation is rejected, i.e. $RA = 1$.

- While $atr \to 0$, we have $AA \to 1$.

Timely Monitoring

Assume $\mathcal{A}$ specifies a safety property,

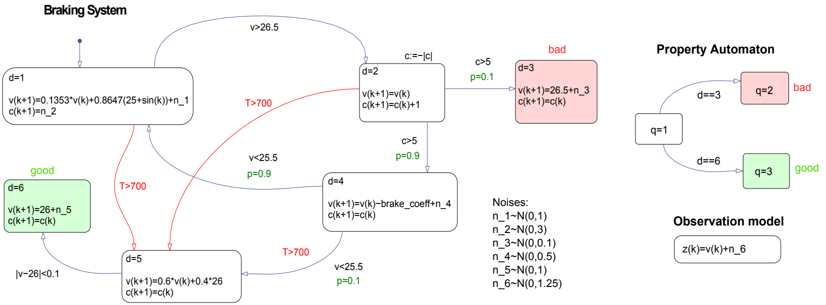
- Define random variable *MTIME*(*atr*) to represent the time taken by a monitor to raise an alarm after failure.
- *H* is exponentially converging monitorable (ECM) w.r.t. $\mathcal{A}$, if *AccPr*(α) converges to 0 exponentially w.r.t. *length*(α) (in a probabilistic sense), for α generated by a bad computation.

Theorem

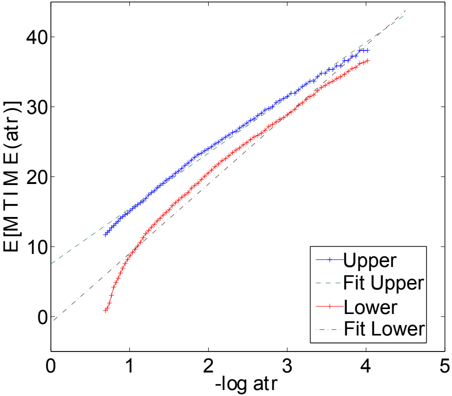
If H is exponentially converging monitorable w.r.t. $\mathcal{A}$,
$E(MTIME(atr)) = O(log(\frac{1}{atr})) \sim O(log(\frac{1}{1-AA}))$.

Implementation of Threshold Monitors:
- Perform state estimation on $H \times \mathcal{A}$ using particle filters.

Example: A Car Braking System



Plot of $E[MTIME(atr)]$ vs. $\log \frac{1}{atr}$.



- Upper bound on $AccPr()$: $AccProb^U(\alpha) = 1 - P[d_{=3}]$;
- Lower bound (no timeout transitions): $AccProb^L(\alpha) = 1 - P[d_{=3}] - \frac{0.1}{1-0.9^2}(P[d_{=1}] + P[d_{=2}] + 0.9P[d_{=4}])$

Current and Future Work

- Implement the monitors on a real system, such as a robotic system
- Optimize particle filter algorithms
- Developing modular monitors
- Generating system model automatically

Thank you!