Verifying Average Dwell Time
by solving optimization problems

Sayan Mitra, Nancy Lynch and Daniel Liberzon

HSCC 2006, Santa Barbara
Stability properties arise naturally, some examples:

- Switching supervisory controllers
- Mobile robots starting from *arbitrary* positions must eventually converge, say on a circle
- Real-time distributed computing with failures; *once failures stop*, the processes must perform some useful computation.
Stability Under Slow Switchings

\[ \frac{\partial V_i}{\partial x} f_i \leq -\lambda V_i \]

\[ V_j \leq \mu V_i \]

- **ADT characterizes switching signal \( \sigma \)**

- **Definition:** Hybrid automaton \( A \) has **average dwell time (ADT)** \( T \) if there exists a constant \( N_0 \) such that for every execution \( \alpha \) of \( A \),

\[ N(\alpha) \leq N_0 + \text{dur}(\alpha)/T. \]

\( N(\alpha) \): # mode switches in \( \alpha \), \( \text{dur}(\alpha) \): duration of \( \alpha \)

**Extra switches:** \( S_T(\alpha) \equiv N(\alpha) - \text{dur}(\alpha)/T \)
Problem statement

- **Theorem** (Morse & Hespanha): For stability it suffices to show that the modes of $\mathcal{A}$ have a set of Lyapunov functions $(\lambda, \mu)$ and that the ADT of $\mathcal{A} > \log \mu / \lambda$.

- Given hybrid automaton $\mathcal{A}$ and $T > 0$, we want to check if $T$ is an ADT for $\mathcal{A}$?

  What is the ADT of $\mathcal{A}$?

  - Invariant-based method [M-Liberzon: CDC04]
  - Optimization-based method for verifying ADT
  - ADT preserving abstraction: switching simulations
Hybrid I/O Automata
[Lynch, Segala, Vaandrager]

- $X$: set of state variables, $A$: set of actions
- Actions bring about state \textit{transitions (mode switches)}, $x \xrightarrow{a} x'$
- A \textit{trajectory} of $X$ is a function $\tau: [0, t] \rightarrow \text{val}(X)$
- An \textit{execution} is a sequence $\alpha = \tau_0 a_1 \tau_1 a_2 \tau_2 \ldots$ $\text{dur}(\alpha) = \sum_i \text{dur}(\tau_i)$

- Notion of external behavior, \textit{input/output actions and variables}
- Abstraction/implementation relations
- Compositionality
Switching simulation

- Consider two hybrid automata $A$ and $B$. A relation $R \subseteq X_A \times X_B$ is a switching simulation relation from $A$ to $B$ if:
  - For every start state of $A$ there is a related start state of $B$
  - If $x \in X_A$, $y \in X_B$, $x R y$ and
    - $x \xrightarrow{a} x'$, exists an execution fragment $\beta$ of $B$, s.t. $y \xrightarrow{\beta}\ y'$ & $x' R y'$ & $N(\beta) \geq 1$, $\text{dur}(\beta)=0$
    - $x \xrightarrow{\tau} x'$, exists an execution fragment $\beta$ of $B$, s.t. $y \xrightarrow{\beta}\ y'$ & $x' R y'$ & $\text{dur}(\tau) \geq \text{dur}(\beta)$

\[ \beta = \tau_0 a_1 \tau_1 a_2 \tau_2 \]
Switching simulations

\[ \mathbf{R} \subseteq X_A \times X_B \]
1. For every start state of \( A \) there is a related start state of \( B \)
2. If \( x \mathrel{\mathbf{R}} y \) and \( x \mathrel{\rightarrow_a} x' \), \( \exists \beta \) s.t. \( y \mathrel{\rightarrow_\beta} y' \) & \( x' \mathrel{\mathbf{R}} y' \) & \( \text{N}(\beta) \geq 1 \), \( \text{dur}(\beta) = 0 \)
3. If \( x \mathrel{\mathbf{R}} y \) and \( x \mathrel{\rightarrow_\tau} x' \), \( \exists \beta \) s.t. \( y \mathrel{\rightarrow_\beta} y' \) & \( x' \mathrel{\mathbf{R}} y' \) & \( \text{dur}(\tau) \geq \text{dur}(\beta) \)

- Suppose \( \mathbf{R} \) is a switching simulation relation from \( A \) to \( B \) and \( T \) be ADT of \( A \)
- Consider any execution \( \alpha = \tau_0 a_1 \tau_1 a_2 \tau_2 \ldots \) of \( A \)
- Inductively construct a corresponding execution \( \eta \) of \( B \)
- \( S_T(\eta) \geq S_T(\alpha) \)

\[ \square \text{Theorem: } \mathbf{R} \text{ is a switching simulation from } A \text{ to } B \implies \text{ADT of } A \geq \text{ADT of } B. \]
Linear Hysteresis switch

- Not initialized hybrid automaton
- Abstraction using switching simulation
Simple Abstraction

- This is in fact a one-clock initialized
- Verifying ADT reduces to finding minimum mean cost cycle.

Use e.g., Karp’s algorithm.

\[ \pi(y.\text{mode}) = x.\text{mode} \land \]
\[ R = \begin{cases} 
\pi(y.\text{mode}) = j \Rightarrow \frac{x.\mu_j}{x.\mu_{\text{min}}} = e^{c_{j,y,t}} \land \\
\pi(y.\text{mode}) \neq j \Rightarrow \frac{x.\mu_j}{x.\mu_{\text{min}}} = y.\text{mode}[i][j], i \in \{1,2\} 
\end{cases} \]
Optimization problem

- \( N(\alpha) \leq N_0 + \frac{\text{dur}(\alpha)}{T} \)
- \( S_T(\alpha) \equiv N(\alpha) - \frac{\text{dur}(\alpha)}{T} \)

\( \text{OPT}(T) : \alpha^* = \arg \max_{\alpha \in \text{execs}} S_T(\alpha) \)

If \( S_T(\alpha^*) \) is bounded then \( T \) is ADT for \( A \),
Otherwise, \( \alpha^* \) an execution violating ADT \( T \)
Optimization based approach

**Theorem (part 1):** If $\max_{\alpha \in \text{cycles}} S_T(\alpha) > 0$ then $OPT(T)$ is unbounded.

- If $\exists$ a cycle with $S_T(\alpha) > 0$ then $\alpha . \alpha . \alpha ...$ is an execution with unbounded extra switches.
Theorem (part 2): For initialized and rectangular $A$, $\text{OPT}(T)$ is unbounded only if $\max_{\alpha \in \text{cycles}} S_T(\alpha) > 0$.

- If $\text{OPT}(T)$ is unbounded, $\exists$ execution $\alpha$, $S_T(\alpha) > m^3$.
- $N(\alpha) > m^3 + \text{dur}(\alpha)/T$.
- $\exists$ sequence of 3 modes that repeat in $\alpha$, say a-b-c.
- Since $A$ is rectangular and initialized, $\exists$ cyclic $\alpha^*$ such that $S_T(\alpha^*) \geq S_T(\alpha)$.
MILP formulation

Mixed Integer Linear Program to find cycles with extra switches for initialized rectangular automata.

Objective function: \[ S_{\tau_a} : \frac{K}{2} - \frac{1}{r_{\tau_a}} \sum_{i=0,2,...}^{K} t_i \]

Mode: \( \forall i \in \{0, 2, \ldots, K\}, \sum_{j=1}^{N} m_{ij} = 1 \) and \( \forall i \in \{1, 3, \ldots, K - 1\}, \sum_{j=1}^{N} \sum_{k=1}^{N} p_{ijk} = 1 \)

Cycle: \( x_0 = x_K \) and \( \forall j \in \{1, \ldots, N\}, m_{0j} = m_{Kj} \)

Preconds: \( \forall i \in \{1, 3, \ldots, K - 1\}, \sum_{j=1}^{N} \sum_{k=1}^{N} G[j, k] \cdot p_{ijk} \cdot x_i \leq \sum_{j=1}^{N} \sum_{k=1}^{N} p_{ijk} \cdot g[j, k] \)

Initialize: \( \forall i \in \{1, 3, \ldots, K - 1\}, \sum_{j=1}^{N} \sum_{k=1}^{N} R[j, k] \cdot p_{ijk} \cdot x_{i+1} \leq \sum_{j=1}^{N} \sum_{k=1}^{N} p_{ijk} \cdot r[j, k] \)

Invariants: \( \forall i \in \{0, 2, \ldots, K\}, \sum_{j=1}^{N} A[j] \cdot m_{ij} \cdot x_i \leq \sum_{j=1}^{N} m_{ij} \cdot a[j] \)

Evolve: \( \forall i \in \{0, 2, \ldots, K\}, x_{i+1} = x_i + \sum_{j=1}^{N} c[j] \cdot m_{ij} \cdot t_i \)
Conclusions

- Using powerful existing tools (MILP) for verifying ADT, i.e., proving stability.
- Switching simulations for abstractions

Future work

- Probabilistic hybrid systems and stability in the stochastic setting, using Lyapunov function like techniques
- Explore other properties that are quantified over executions; liveness properties
- Finding switching simulations automatically?