HyTech: A Model Checker for Hybrid Systems

Ming Xiong
James Hill
Motivation

- In mission-critical applications, formal guarantees about the absence of logical and timing errors are desirable
- Time Automata – focus on real-time systems
- Hybrid Automaton – focus on more general hybrid systems
Model-Checking Technology

- Used for system verification
- A formal model of a system is checked, fully automatically, for correctness with respect to a requirement expressed in temporal logic
- Symbolic model checking has been widely used to verify complex systems
Overview of HyTech

- Provides a yes or no to correctness requirement
- Provides diagnostic information that aids in design and debugging, e.g. computes necessary constraints that help finding correct design parameters
- Approximate system using linear hybrid automata
Hybrid Dynamic System

- A dynamic system mixing Boolean-valued variables and real-valued variables, an variant of hybrid system
- Described by $\mathbb{B}^m \times \mathbb{R}^n$
- Example: thermostat

![Thermostat automaton](image)
A **hybrid automaton** is defined as $H = (X, V, \text{flow}, \text{inv}, \text{init}, E, \text{jump}, e, \Sigma, \text{syn})$ where

- $V$ is a set of control modes
- $X$ is a set of continuous variables
- $\text{Init}$ is a labeling function that assigns an initial condition to each control mode in $V$
- $\text{flow}$ is a labeling function that assigns a flow condition to each control mode in $V$
- $\text{Inv}$ is a labeling function that assigns an invariant condition to each control mode in $V$
- $E$ is a collection of control switches
- $\text{Jump}$ is a labeling function that assigns a jump condition to each control switch in $E$
- $\Sigma$ is a finite set of events
- $\text{Syn}$ is a labeling function that assigns an event in $\Sigma$ to each control switch in $E$
Safety Requirement

- Asserts that nothing bad will happen
- Safety verification amounts to computing the set of reachable states (to see if it’s unsafe)
- State assertion
  - a function that assigns to each control in $\forall a$ predicate $\varphi$ over the variables in $X$
  - the states for which $\varphi$ is true are called $\varphi$-states
    - e.g. $\text{inv}$-states are precisely admissible states
- A hybrid automata $H$ satisfies the safety requirement specified by $\text{unsafe}$ if the state assertion $\text{unsafe}$ is false for all reachable states of $H$
Linear Hybrid Automata

- Requirements
  - Linearity
  - Flow independence

- Theorem:
  If $A$ is a linear hybrid automaton, and $\phi$ is a linear state assertion for $A$, then $\text{Post}(\phi)$ can be computed and the result is again a linear state assertion for $A$.

- The above theorem enables safety verification as well as temporal-logic model checking:
  - i.e. in HyTech, the model to be checked has to be a linear model.
What about non-linear model?

- No direct means of automatically verifying non-linear model
- Has to convert a non-linear model to a linear model
  - Clock translation
  - Linear phase-portrait approximation
The idea is sometimes the value of a variable can be determined from a past value (a constant) and the time that has elapsed since the variable had that value

- Solvability
- Initialization
The idea is to relax nonlinear flow, invariant, initial and jump condition using weaker linear condition: each nonlinear predicate \( p \) is replaced by a linear predicate.

Fig. 1. Thermostat automaton

Fig. 4. Linear phase-portrait approximation of the thermostat automaton

Fig. 5. Tighter linear phase-portrait approximation of the thermostat automaton

Need to be careful about the approximation
Safety Verification for Thermostat systems

- Add extra variables or control modes to specify our safety requirement
- Use both *reach* and *unsafe* assertion
  - if there is any state for which reach and unsafe are true, the safety requirement is violated

Now we can specify $y = 60$ and $z \geq \frac{2y}{3}$

Linear phase-portrait approximation

**Fig. 1. Thermostat automaton**

**Fig. 2. Thermostat automaton augmented for safety verification**
HyTech performs these computations for us, until neither new jump successors nor new flow successors can be found.
Sometimes it is convenient to build a separate automaton, called a monitor, whose role is to enter an unsafe state precisely when the original system violates a requirement.

Monitor must observe the original system without changing its behavior.