Conflict-Tolerant Specifications for Hybrid Systems

Deepak D'Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

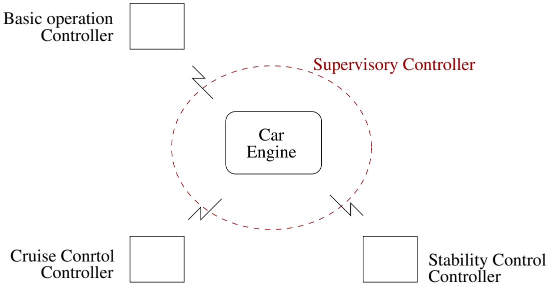
Joint work with Madhu Gopinathan, S. Ramesh, and Prahlad Sampath.

January 29, 2011

Outline

Overview

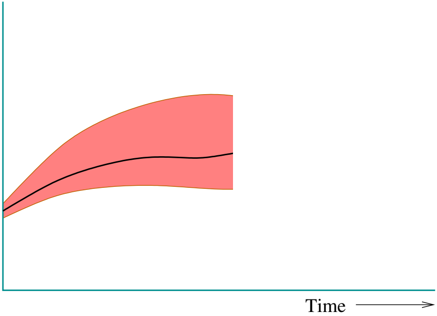
- We consider systems which are composed of a base system + multiple controllers + a supervisor.
- Supervisor chooses when to provide the control input of a controller to the base system (e.g. based on priority).



We propose a way of specifying the behaviour of individual controllers.

Why a classical safety specification is inadaquate

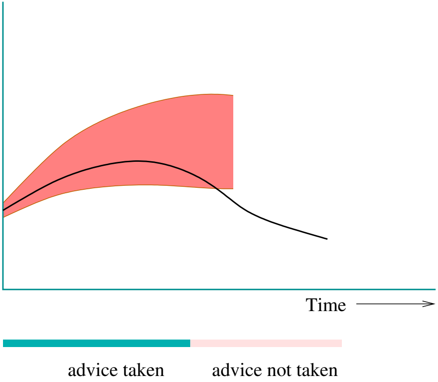
Specifies a safety "cone" (prefix-closed set of behaviours) within which the behaviour of the controlled system must lie.



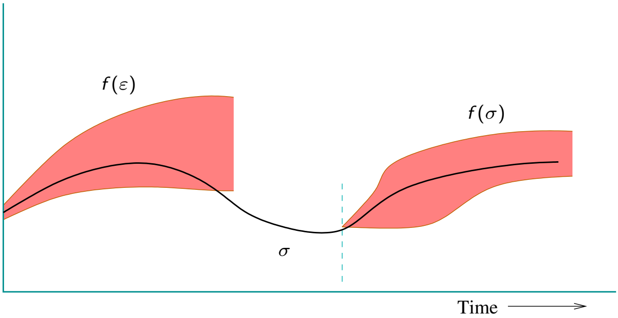
Why a classical safety specification is inadaquate

What happens if the controller's input is disregarded by the supervisor (due to a conflict)?

- The resumed controller has no specification to adhere to.



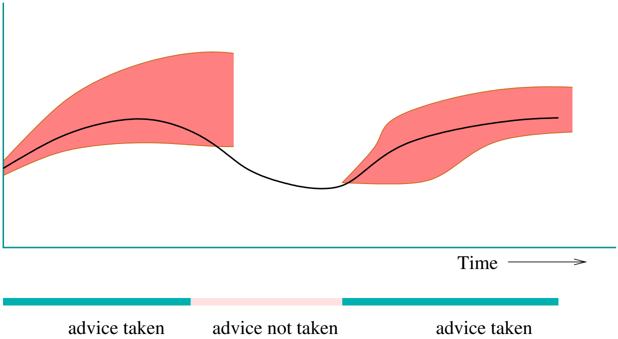
Conflict-tolerant specification



An advice function f which specifies a safety cone $f(\sigma)$ after each behaviour σ of the base system.

Conflict-tolerant Specification: Guarantee

Suppose a controller $\mathcal{C}$ satisfies its tolerant specification $\mathcal{S}$ wrt a base system $\mathcal{B}$.



- Then in every period in which it is control, $\mathcal{C}$ does "the right thing" (according to $\mathcal{S}$).
- This guarantee is regardless of other controllers/supervisors it is composed with.

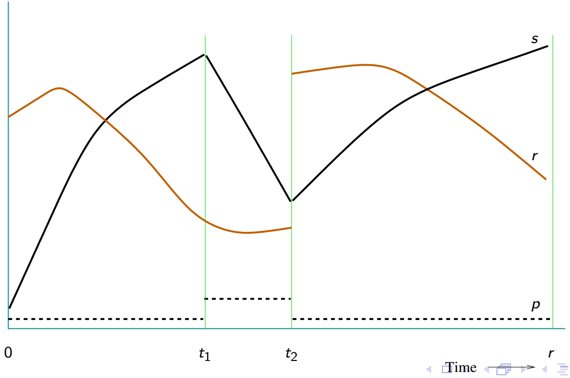
In this talk

- We give a mechanism to describe conflict-tolerant specifications for hybrid systems.
- Solve the verification problem when components are given as initialized rectangular hybrid automata.

Behaviour of a hybrid system

A signal over a set of variables W is a function $\sigma : [0, r) \longrightarrow \mathbf{W}$ which has only finitely many points of discontinuity.

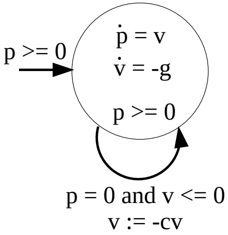
- There is a strictly increasing sequence of time points $t_0 = 0 < t_1 < t_2 < \cdots < t_n = r$ such that σ is continuous in the interval $[t_k, t_{k+1})$.



Example hybrid system: Bouncing ball

System variables:

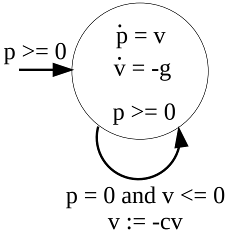
- p: vertical position (height) of the ball.
- v: velocity of ball.

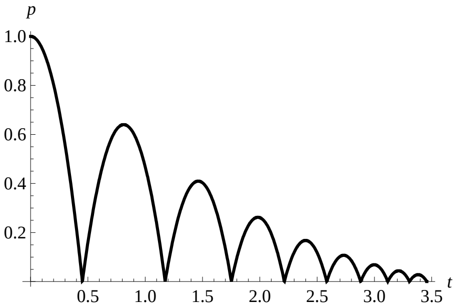


Example hybrid system: Bouncing ball

System variables:

- p: vertical position (height) of the ball.
- v: velocity of ball.

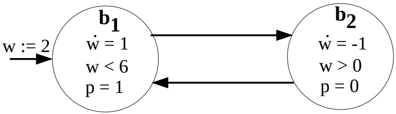




Example hybrid system: Water tank with pump

System variables:

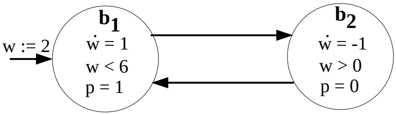
- w: level of water in tank.
- p: On/Off status of pump (1="on").

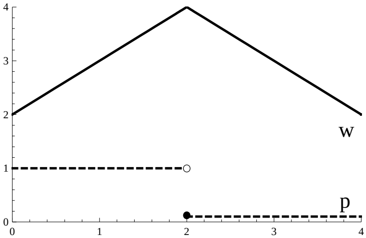


Example hybrid system: Water tank with pump

System variables:

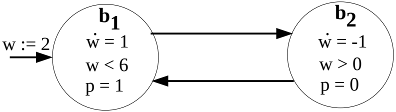
- w: level of water in tank.
- p: On/Off status of pump (1="on").



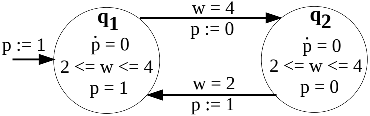


Controller for water tank

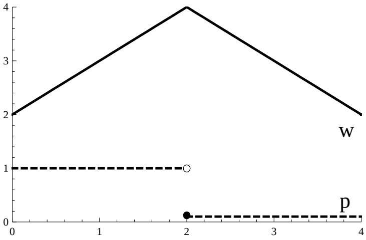
- System variables: $X = \{w\}$
- Control variables: $U = \{p\}$.



Water Tank

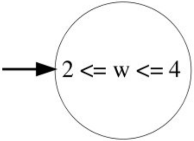


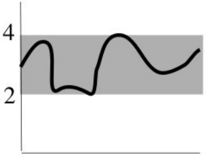
Controller



A classical specification for water level controller

Classical safety specification = prefix-closed set of signals.



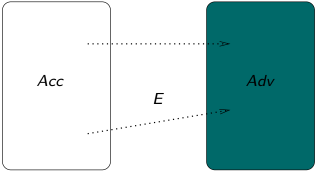


Conflict-tolerant specification

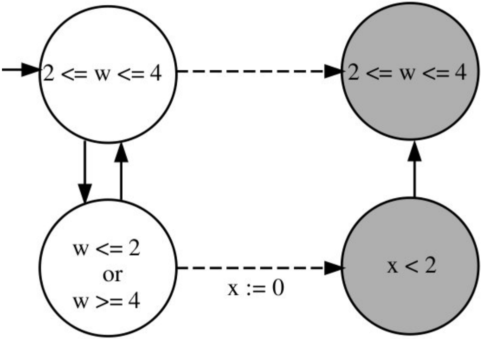
An advice function over a set of variables W is function $f : Signals \rightarrow 2^{Signals}$ such that each $f(\sigma)$ is prefix-closed.

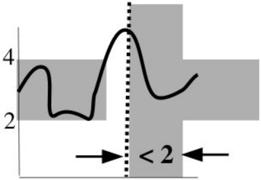
Mechanism to specify such advice functions: $\mathcal{S} = (Acc, Adv, E)$ where

- Acc and Adv are hybrid automata over W
- E is a set of edges between Acc and Adv called an advice relation.

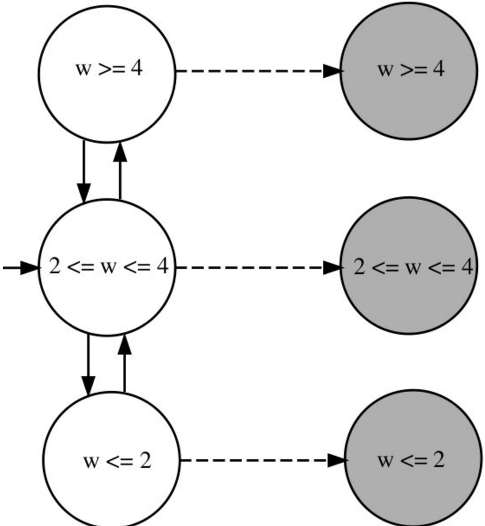


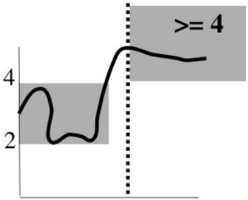
Example tolerant specifications for water level controller I



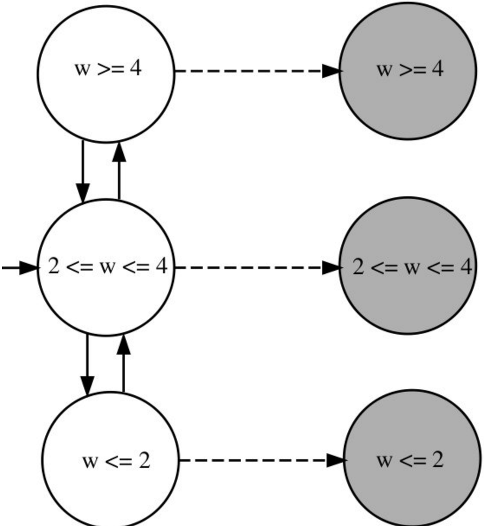


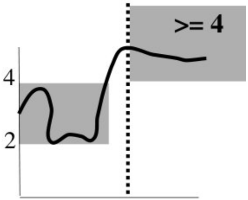
Example tolerant specifications for water level controller II





Example tolerant specifications for water level controller II

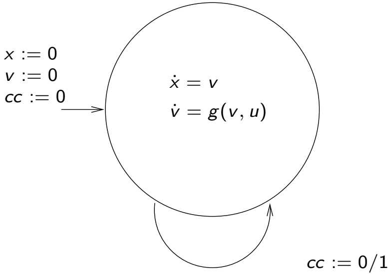




Note: Both tolerant specs induce the same classical spec but are quite different as tolerant specs.

Car motor base system

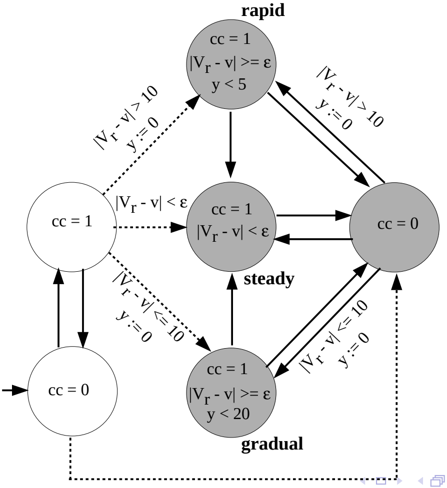
System variables: $\{x, v, cc\}$, Input variables: $\{u\}$.



Consider two controllers: for "cruise-control" and "stability-control".

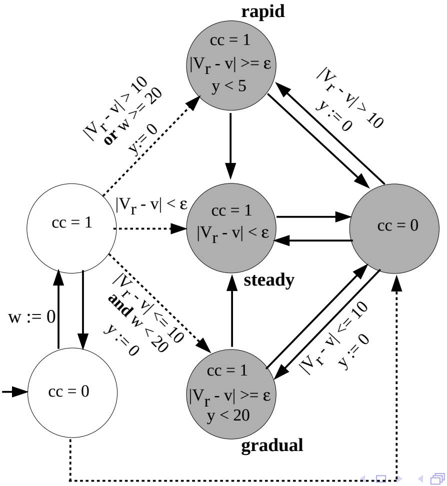
Cruise control tolerant specification I

"Reach set point within 20 sec if already close to it, else reach within 5 sec."



Cruise control tolerant specification II

"Reach set point within 5 sec if far from it or if *cc* has been on for more than 20 sec; else reach within 20 sec."



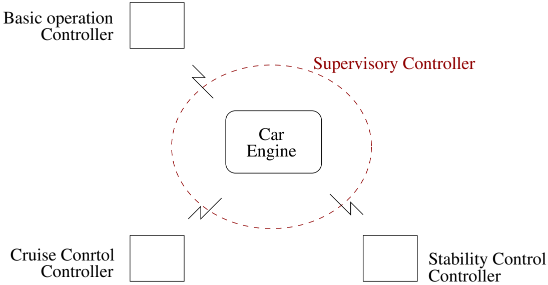
Verification for rectangular hybrid automata

We can solve the verification problem for such specs: Given

- a base system $\mathcal{B}$ modelled as an initialized rectangular automaton over (X, Y),
- a controller $\mathcal{C}$ modelled as an initialized rectangular automaton over (X, Y),
- a tolerant spec $\mathcal{S} = (Acc, Adv, E)$ whose components are IRHA:

we can check whether $\mathcal{C}$ satisfies $\mathcal{S}$ wrt $\mathcal{B}$.

Conclusion



- Conflict-tolerant specifications more richly capture a controller's specification.
- A modular or "compositional" way of developing and reasoning about systems with multiple controllers.
- A mechanism to specify them via hybrid automata.
- Decision procedure for the verification problem when components are given as initialised rectangular hybrid