Safety-Aware Flexible Schedule Synthesis for Cyber-Physical Systems using Weakly-Hard Constraints

Shengjie Xu, Bineet Ghosh, Clara Hobbs, P. S. Thiagarajan, and Samarjit Chakraborty

Motivation

- Modern autonomous systems – multiple controllers on a shared computation resource
- Two stage process:
  - Control engineers design controllers and set deadlines
  - Embedded systems engineers schedule tasks to meet deadlines
- Meeting all the deadlines of the control tasks comes at the expense of pessimistic and inefficient implementations

Motivation

Can “system-level” property such as control safety be preserved despite that some deadlines are missed?

Safety

- How do we define safety?
- One notion of safety: the plant deviates from an ideal behavior no more than a predetermined threshold

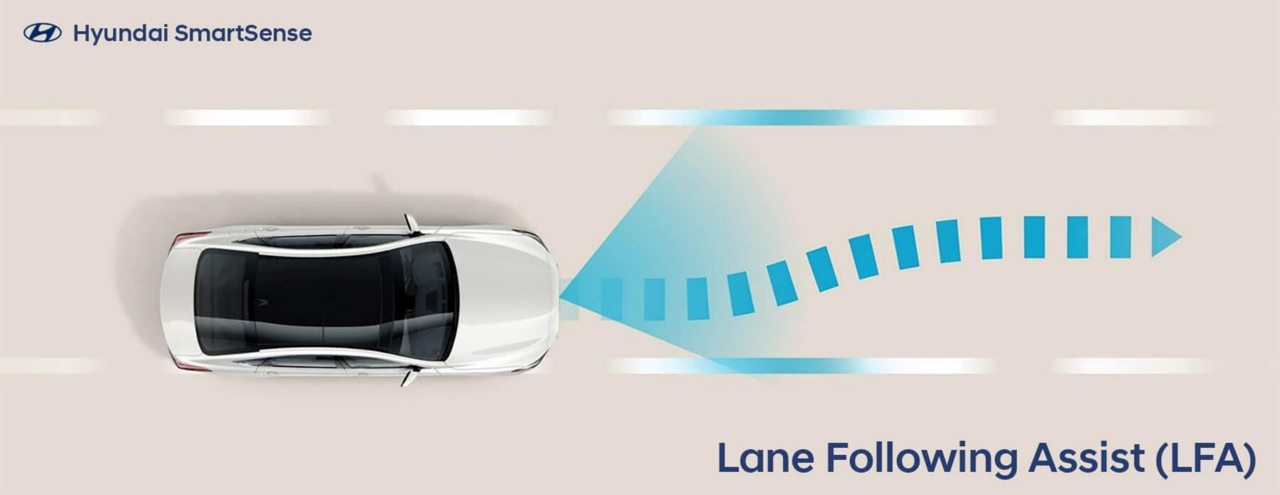


Image credit: https://www.verneidehyundaisiouxcity.com/hyundai-lane-following-assist-lfa/

Problem Statement

- Given a set of tasks, can we schedule all of them and satisfy their respective safety properties, without necessarily meeting all their deadlines?
- Advantages:
  - Reduce the pessimism within an implementation
  - Focus directly on the property of consequence, *viz.*, safety property
  - Does not require redesign of control algorithms

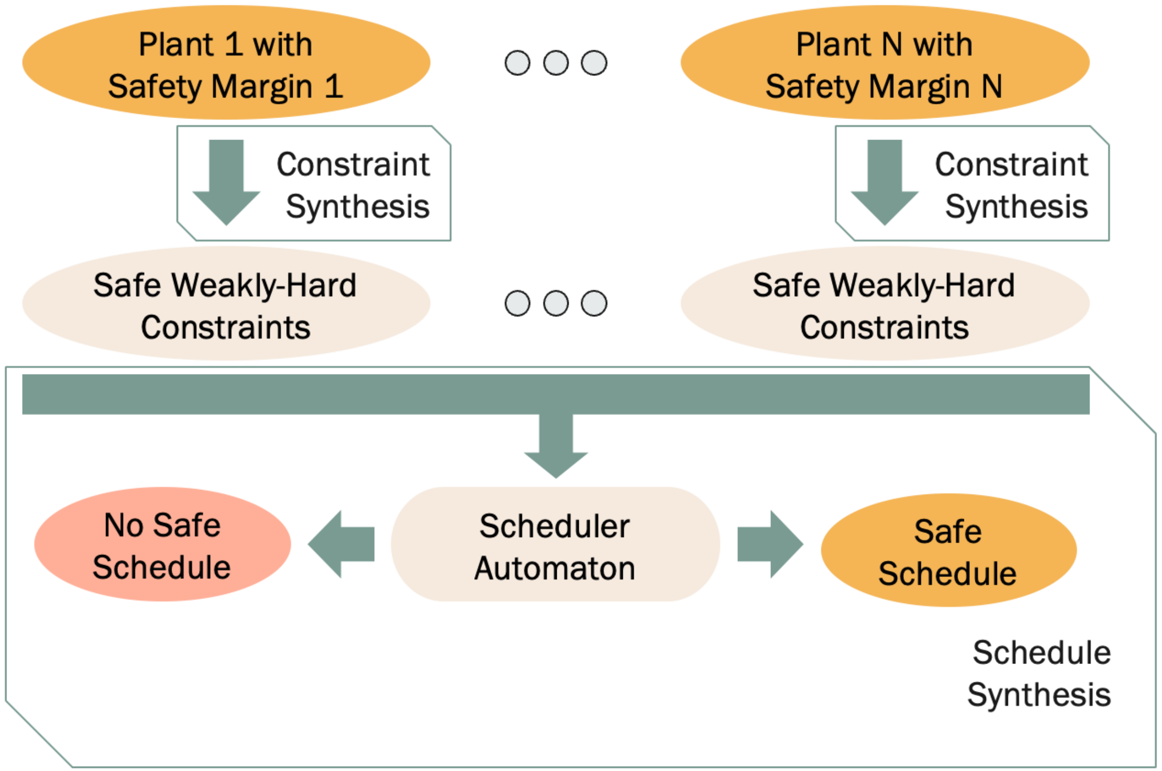
Example

- Five tasks with same period
- Only two of them can be scheduled in each 20 ms slots
- Cannot be scheduled if all deadlines are to be met
- But we can schedule them if safety properties are what need to be satisfied

Dynamical System	Period
RC Network (RC)	20 ms
F1Tenth Car (F1)	20 ms
DC Motor (DC)	20 ms
Car Suspension (CS)	20 ms
Cruise Control (CC)	20 ms

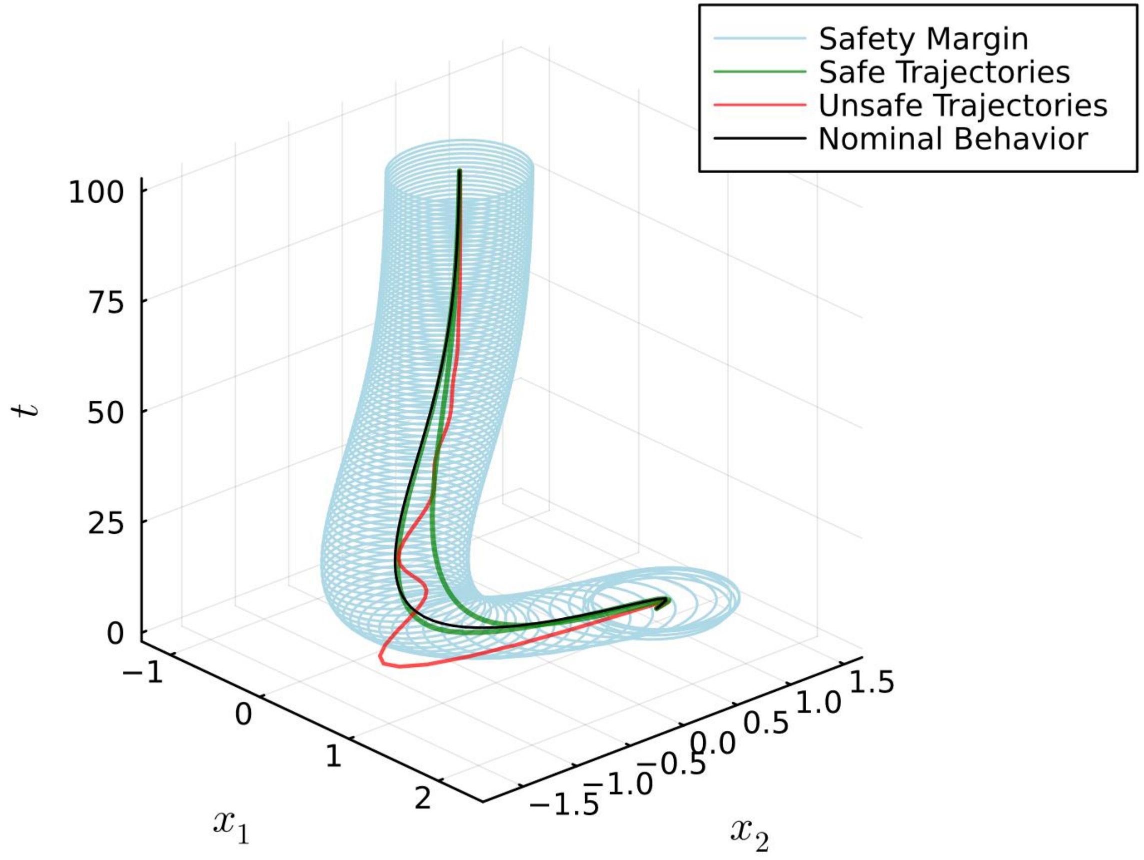
Proposed Method

- Constraint synthesis: safety requirement → how to schedule one system safely
- Schedule synthesis: knowledge about all systems → safe schedule of all tasks



Constraint Synthesis

- Given one dynamical system
- We want to find out safe patterns of deadline hits and misses



Safety: Revisit

- Trajectory
- Nominal trajectory
- Safety margin
- *How do we relate deadline misses with control safety?*

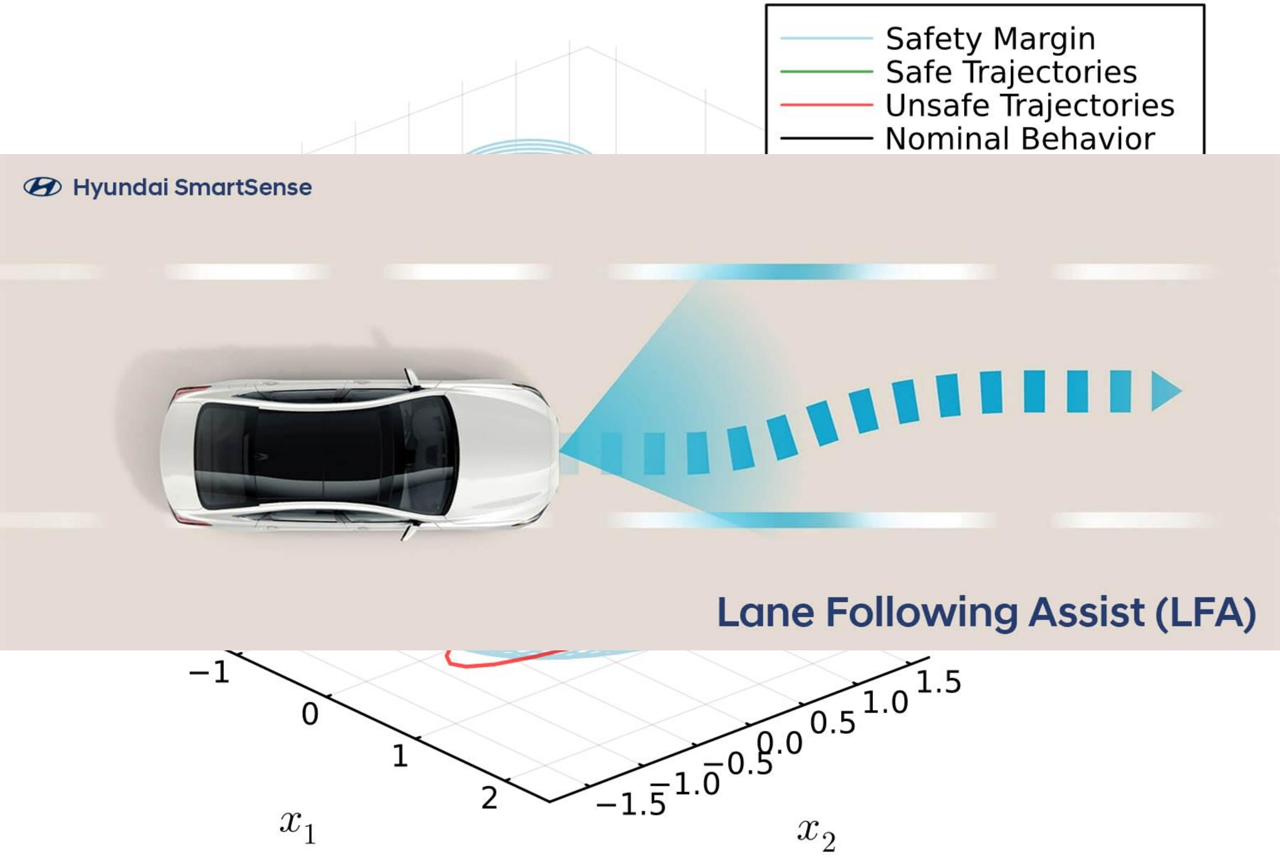
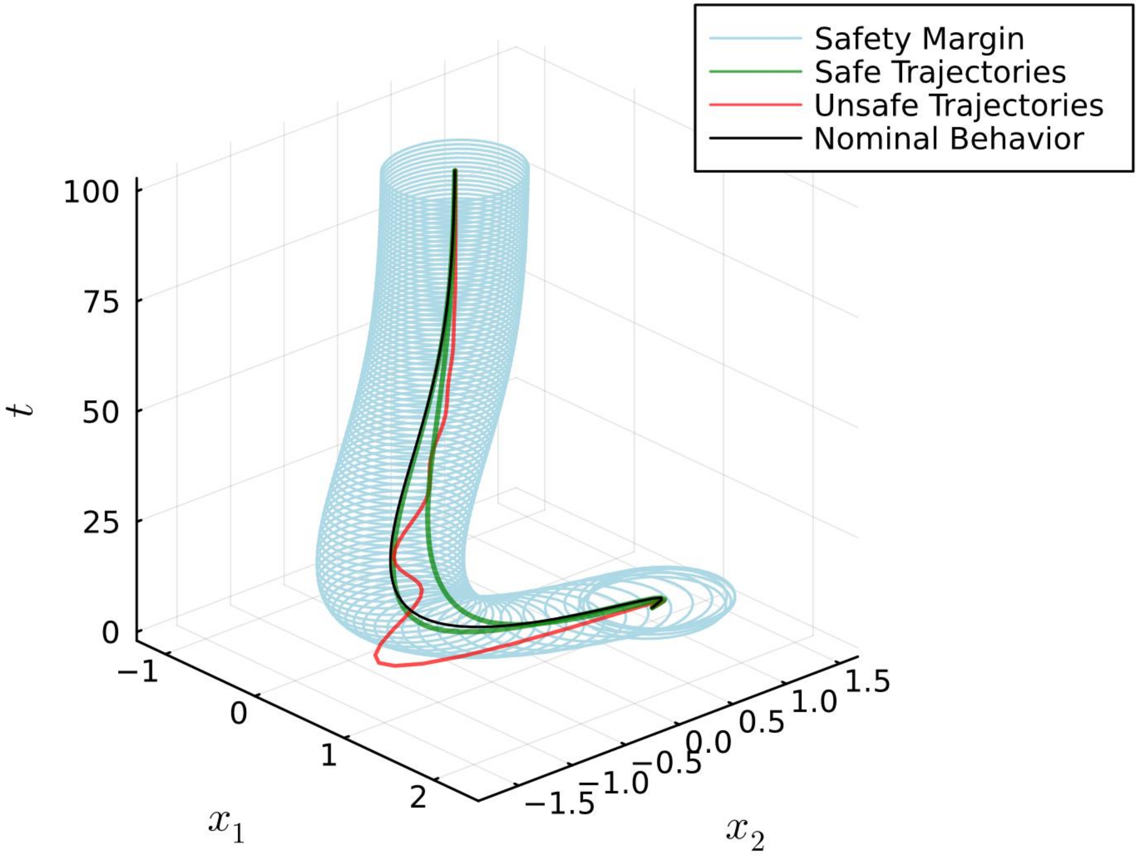


Image credit: https://www.verneidehyundaisiouxcity.com/hyundai-lane-following-assist-lfa/

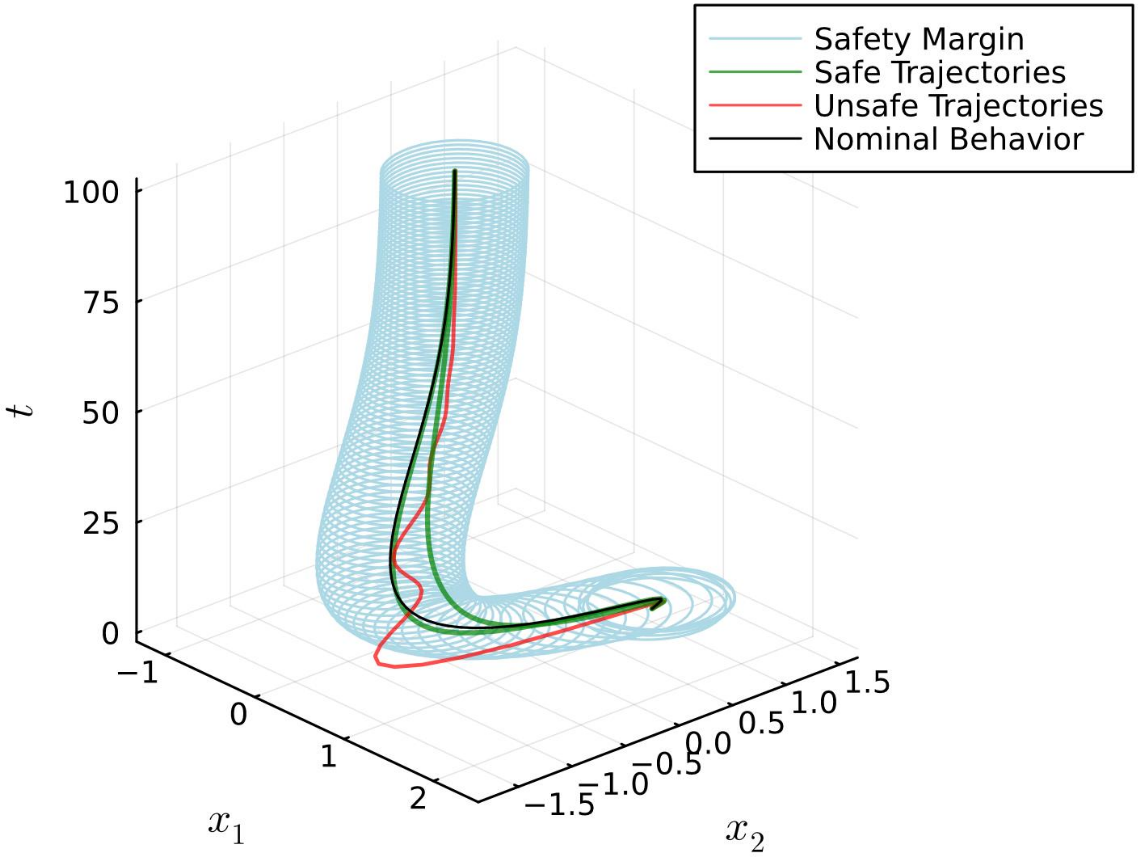
Weakly-Hard Constraints

- $\binom{m}{k}$: in every window of k periods, at least m deadlines must be met
- Example: $\binom{3}{5}$
  - 1 0 1 1 0 1 0 0 1…
- A constraint corresponds to multiple such sequences



Constraint Synthesis

- A weakly-hard constraint $\binom{m}{k}$ corresponds to a set of trajectories
- $\mathrm{d}(\mathrm{m},\mathrm{k})$: maximum deviation of trajectories that satisfy $\binom{m}{k}$
- We mark constrains with $\mathrm{d}(\mathrm{m},\mathrm{k}) \leq$ safety margin as safe

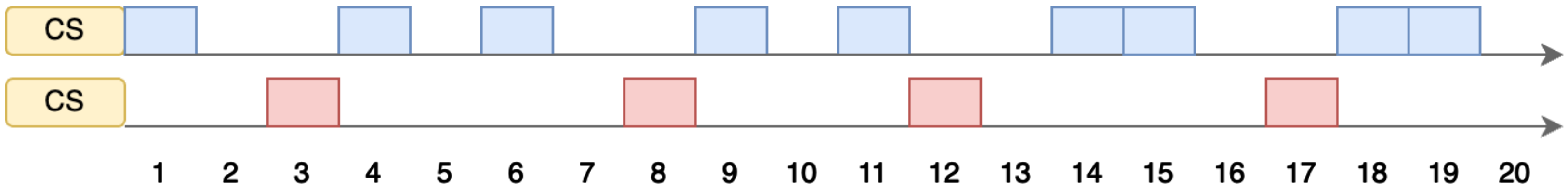


Constraint Synthesis

- We obtain a list of safe weakly-hard constraint
- A schedule that satisfies any one of the safe weakly-hard constraints is guaranteed safe
- No safety guarantee otherwise

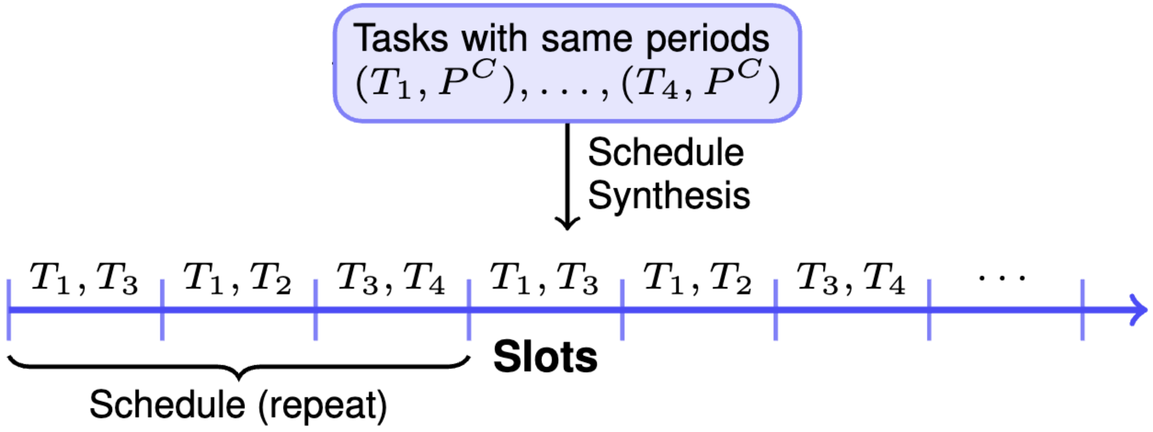
Window Size (k)	Minimum Hits (m) 1	2	3	4	5
2	✓	—	—	—	—
3	×	✓	—	—	—
4	×	✓	✓	—	—
5	×	✓	✓	✓	—
6	×	×	✓	✓	✓

Safe weakly-hard constraints for Car Suspension (CS)



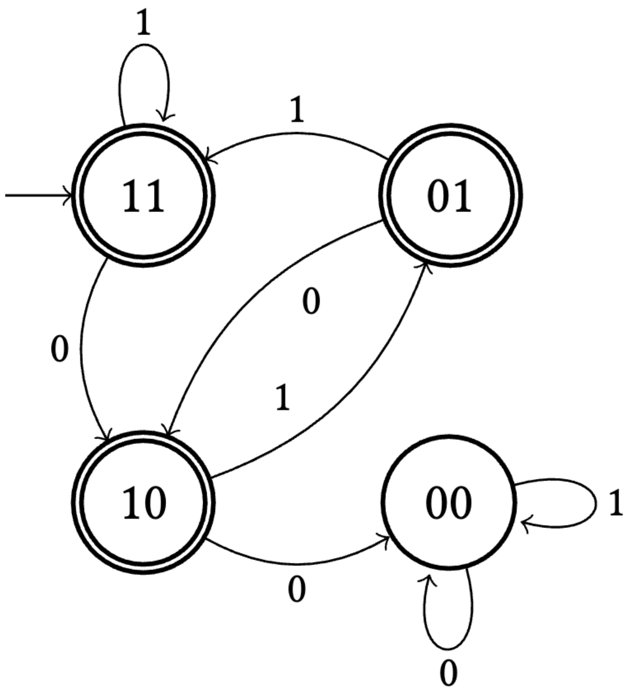
Schedule Synthesis

- Given N systems each with a list of safe constraints
- At most $J\ (< N)$ controller tasks can be scheduled in each slot
- Find a safe schedule

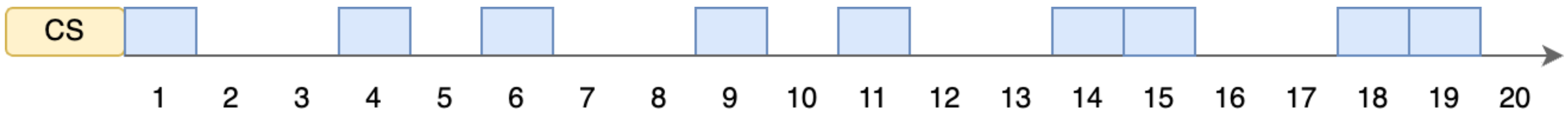


Schedule Synthesis

- $\binom{m}{k}$ represents a regular language
- The union of all constraints for a controller is also regular; we call this a controller automaton
- Accepted strings represent safe schedules for one controller

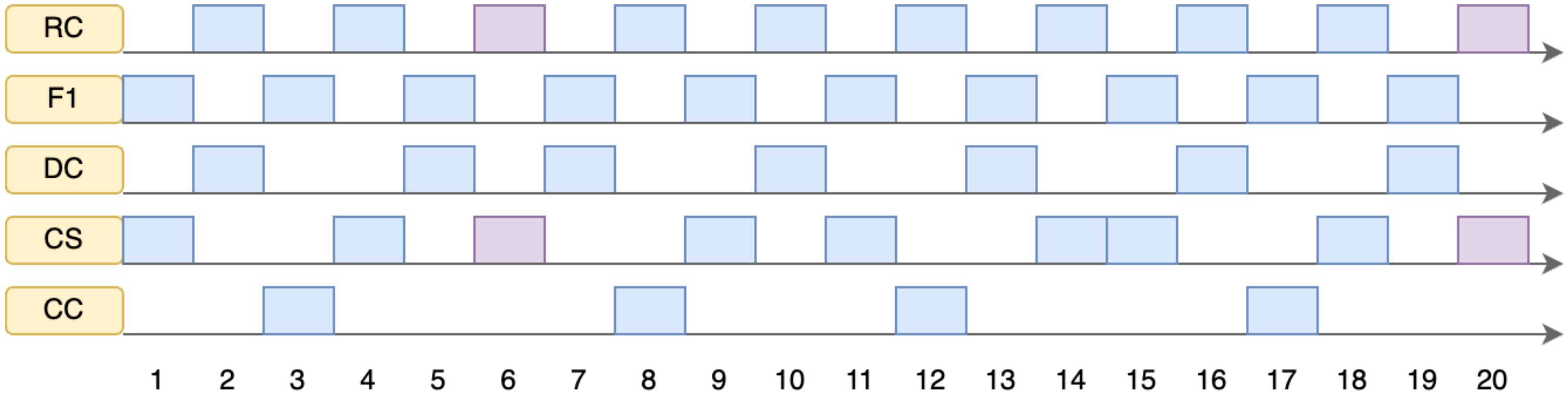


The automaton modelling the weakly-hard constraint $\binom{1}{2}$



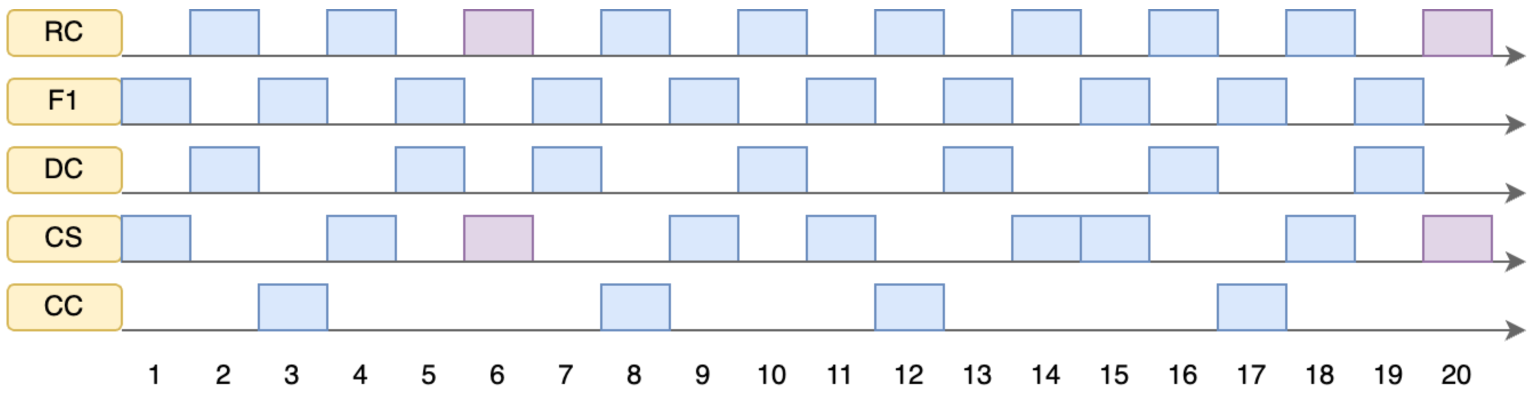
Schedule Synthesis

- Controller automata → scheduler automaton
- Accepting string represent safe schedules for all controllers
- Interpreting the schedule:
  - Scheduled tasks meet their deadline for that period
  - Non-scheduled tasks miss their deadline for that period



Case Study

- We use five control systems to evaluate our methods
- Findings regarding weakly-hard constraints:
  - Utilization based test does not apply
  - Least laxity first (LLF) is no longer an optimal scheduler



Model	Window Size (k)	Minimum Hits (m) 1	2	3	4	5
RC network	2	✓	–	–	–	–
	3	✓	✓	–	–	–
	4	×	✓	✓	–	–
	5	×	×	✓	✓	–
	6	×	×	×	✓	✓
F1 Tenth	2	✓	–	–	–	–
	3	×	✓	–	–	–
	4	×	×	✓	–	–
	5	×	×	×	✓	–
	6	×	×	×	×	✓
DC Motor	2	✓	–	–	–	–
	3	✓	✓	–	–	–
	4	✓	✓	✓	–	–
	5	×	✓	✓	✓	–
	6	×	×	✓	✓	✓
Car Suspension	2	✓	–	–	–	–
	3	×	✓	–	–	–
	4	×	✓	✓	–	–
	5	×	✓	✓	✓	–
	6	×	×	✓	✓	✓
Cruise Control	2	✓	–	–	–	–
	3	✓	✓	–	–	–
	4	✓	✓	✓	–	–
	5	✓	✓	✓	✓	–
	6	×	✓	✓	✓	✓

Conclusions

- Safety properties of control systems are linked to the schedules of their controller tasks implemented on a shared resource
- Safe weakly-hard constraints are identified
- Safe schedule is generated from safe constraints (if exists)
- We note that requiring the periods of all control tasks to be the same is limiting

Thank you!