Some motivations for building an approximation theory of hybrid signals and systems

Paul Caspi

Verimag-Cnrs
www-verimag.imag.fr

TCC, Santa Barbara
April 2006

Introduction

- Model-based development widely recognised as a method of choice for efficient and safely design
- More effectively used in embedded control
  - e.g., automatic code generation in Airbus fly-by-wire (1984)
  - Simulink-CAN, Simulink-TTA,...
- But rather empirical, lack of foundations

Some issues:

1. Model-based development in control and in computer science
2. Sampling theory for discrete event and hybrid systems
3. Fault-tolerance in GALS

Model-based design in computer science and control

Model-based design in computer science

- Starts from a non deterministic specification
- Based on successive property-preserving refinements
- Until an implementation is reached

Remarks:

- This is an idealised scheme, seldom fulfilled
- Yet has a paradigmatic value
- Some real-world impressive achievements in control!!
  - B method (Abrial): Paris, Barcelona, New York subways

Model-based design in computer science

$MACHINE$	$initial$
$SETS$	$persons, buildings$
$ABSTRACT\_CONSTANTS$	$state, authorisation$
$PROPERTIES$	$persons \neq \emptyset \ \wedge \ buildings \neq \emptyset$
	$\wedge \quad state \in persons \rightarrow buildings$
	$\wedge \quad authorisation \in persons \leftrightarrow buildings$
$INVARIANT$	$state \subseteq authorisation$
$OPERATION$	$move \mathrel{\hat{=}} \ ANY \quad (p, b)$
	$WHERE \quad (p, b) \in authorisation$
	$\wedge \ state(p) \neq b$
	$THEN \quad state(p) := b$
	END
END	

Model-based design in computer science

Further steps:

- Add implementation details:
  - paths, doors, badge controls,...
- Separate controllers from environment !!!
- Generate control programs

Model-based design in computer science and control

Model-based design in control science

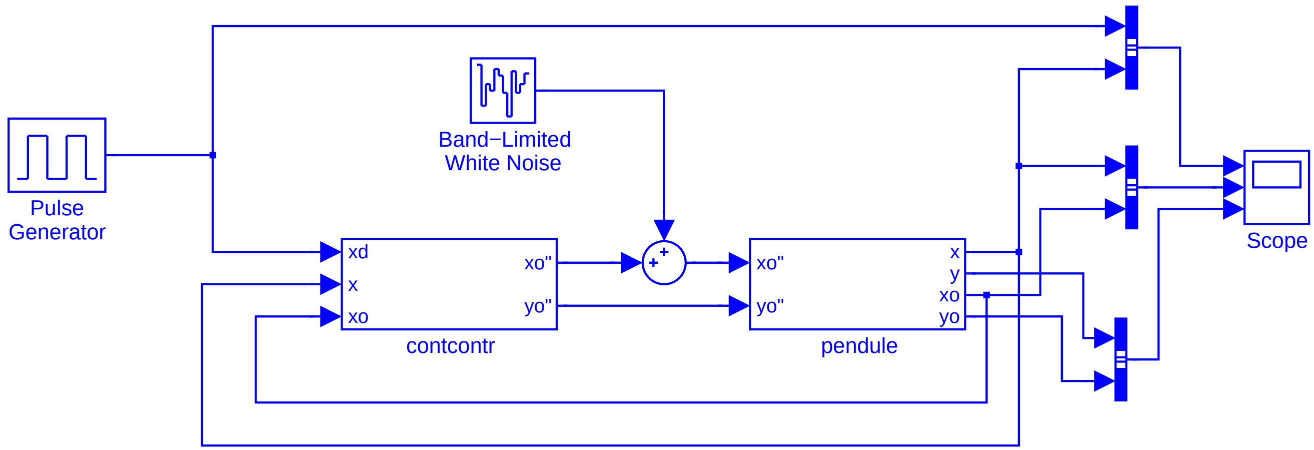
- Start from a perfect model
- Design a robust controller
- Add perturbations and implementation details and checks for robustness

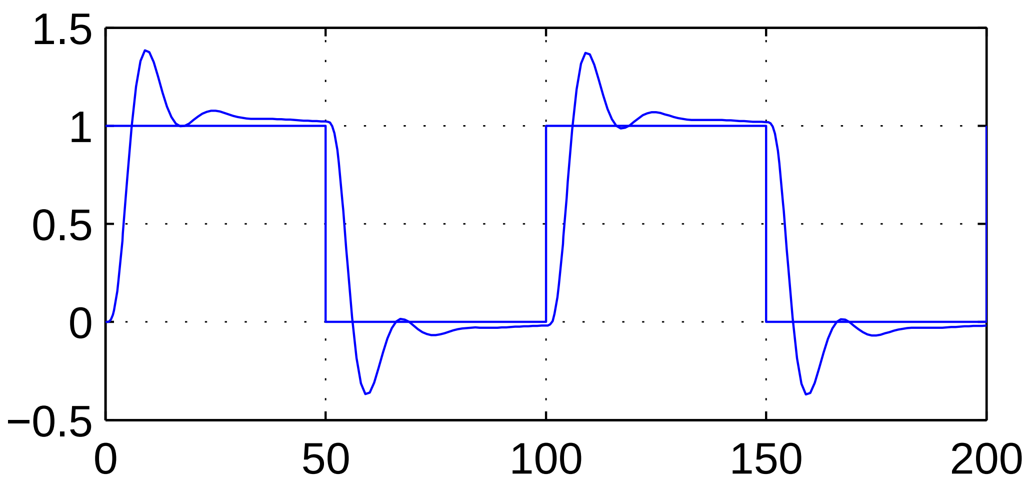
Remarks:

- This is also an idealised scheme

Model-based design in control

Perfect model

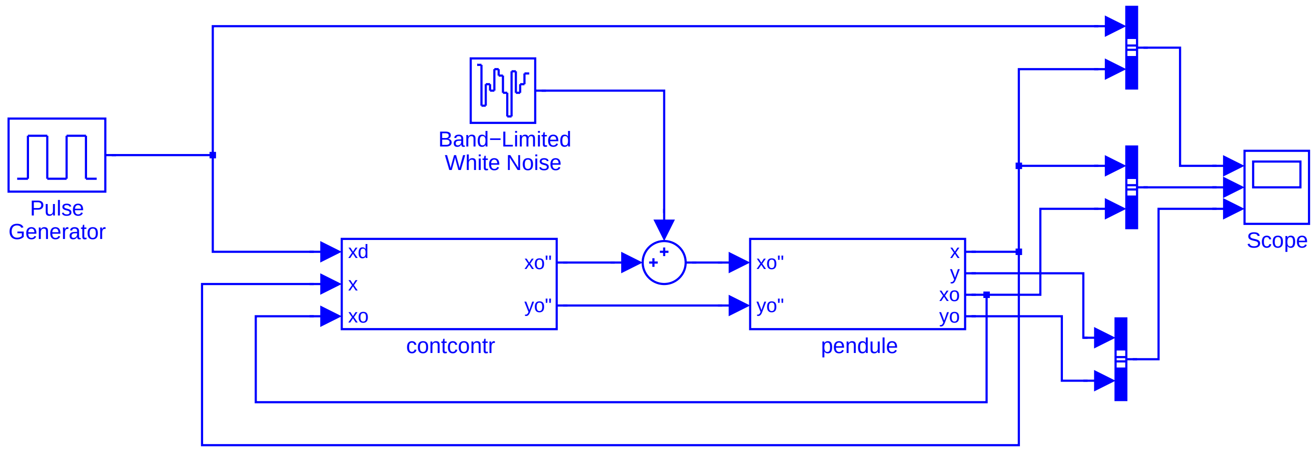


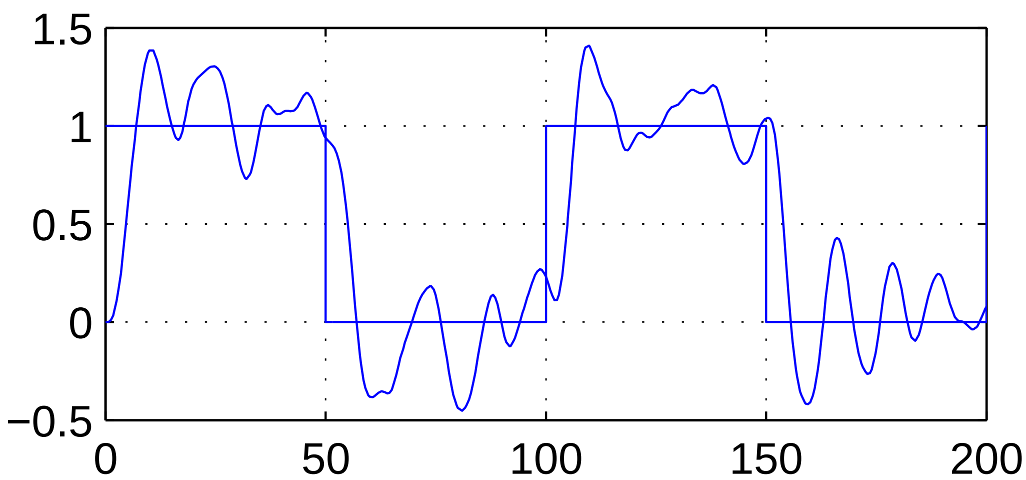


Time offset: 0

Model-based design in control

Perfect model with noise

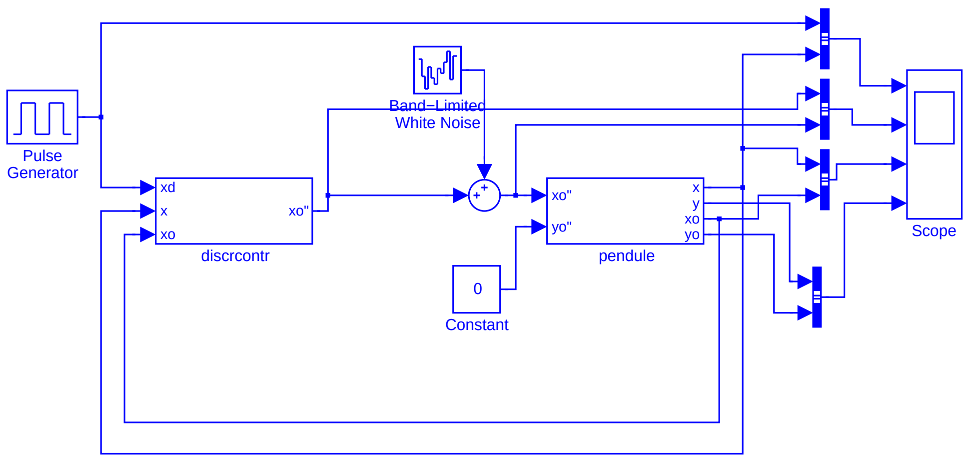


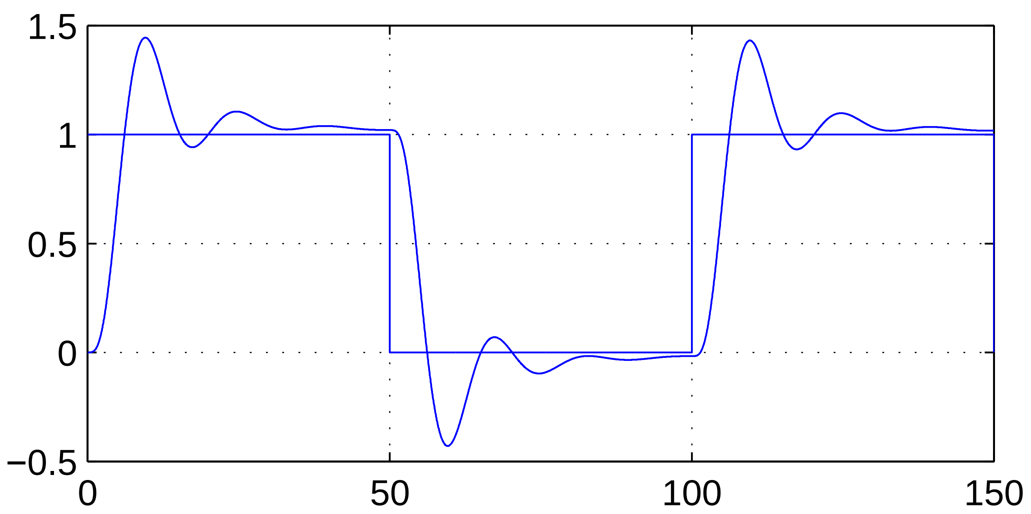


Time offset: 0

Model-based design in control

Discrete-time controller

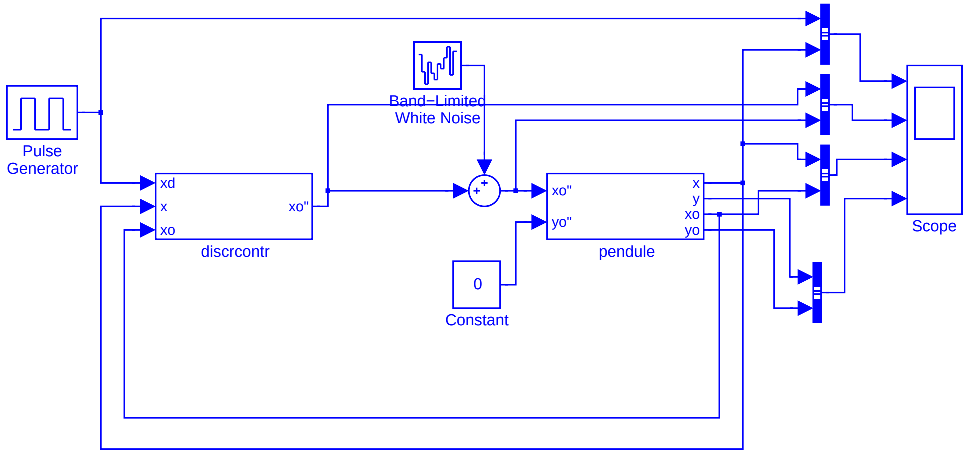


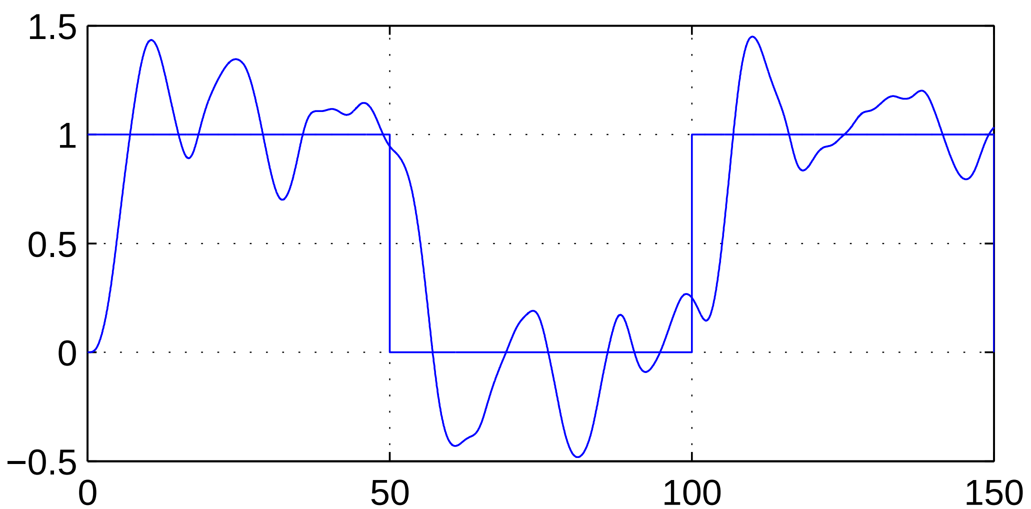


Time offset: 0

Model-based design in control

Discrete-time controller with noise

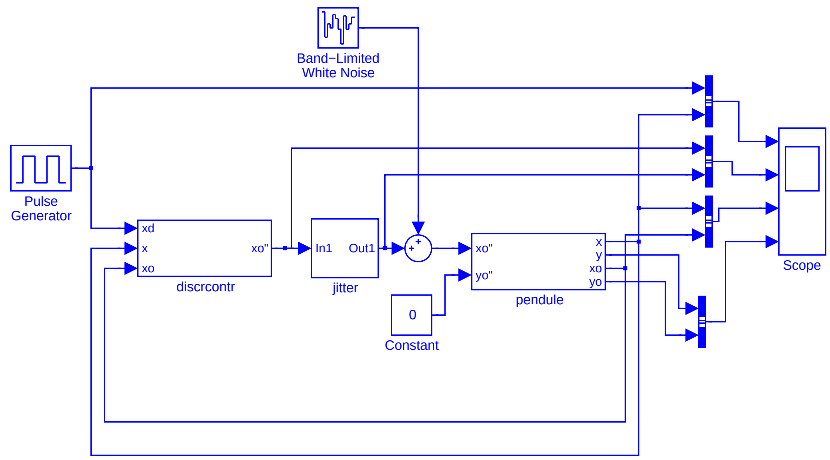


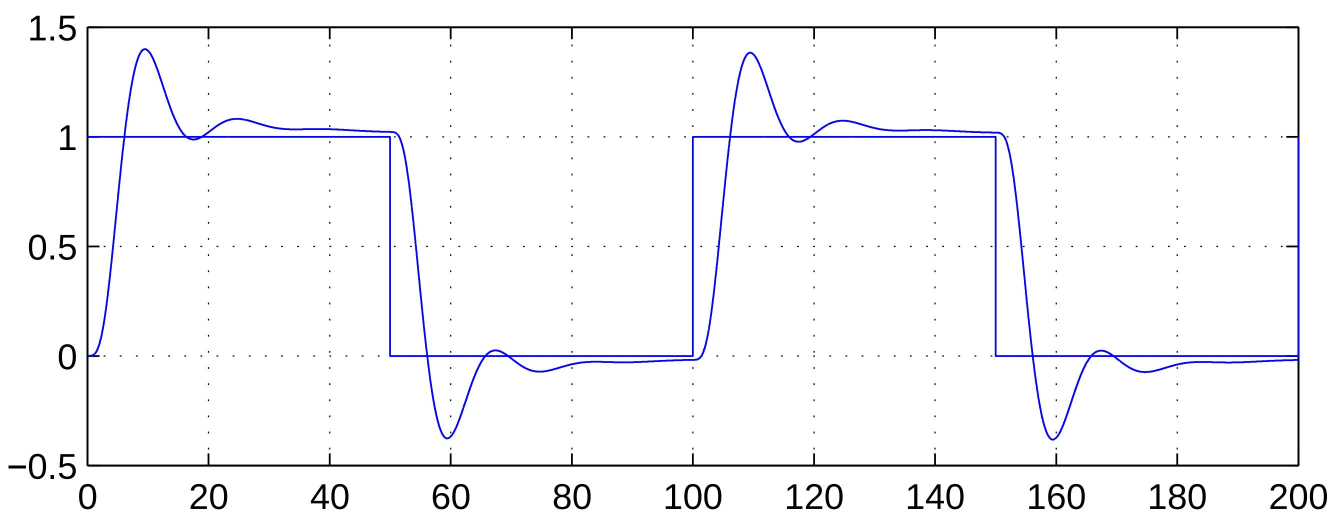


Time offset: 0

Model-based design in control

Discrete-time controller with jitter

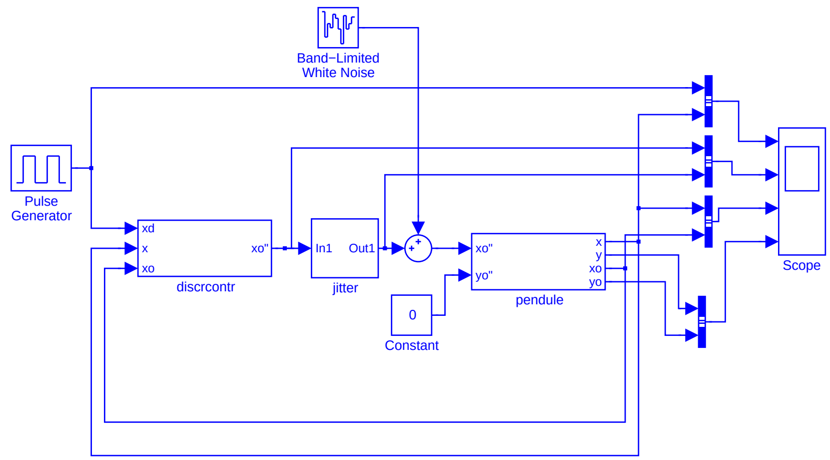


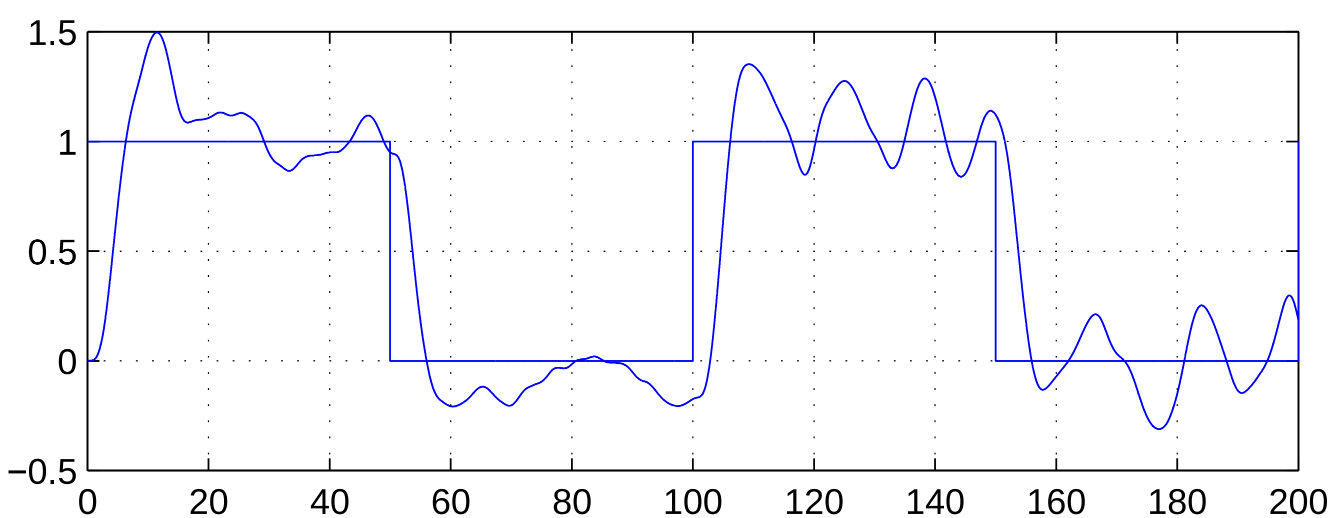


Time offset: 0

Model-based design in control

Discrete-time controller with jitter and noise





Time offset: 0

Model-based design in computer science and control

How can we make them converge ??

A suggestion :

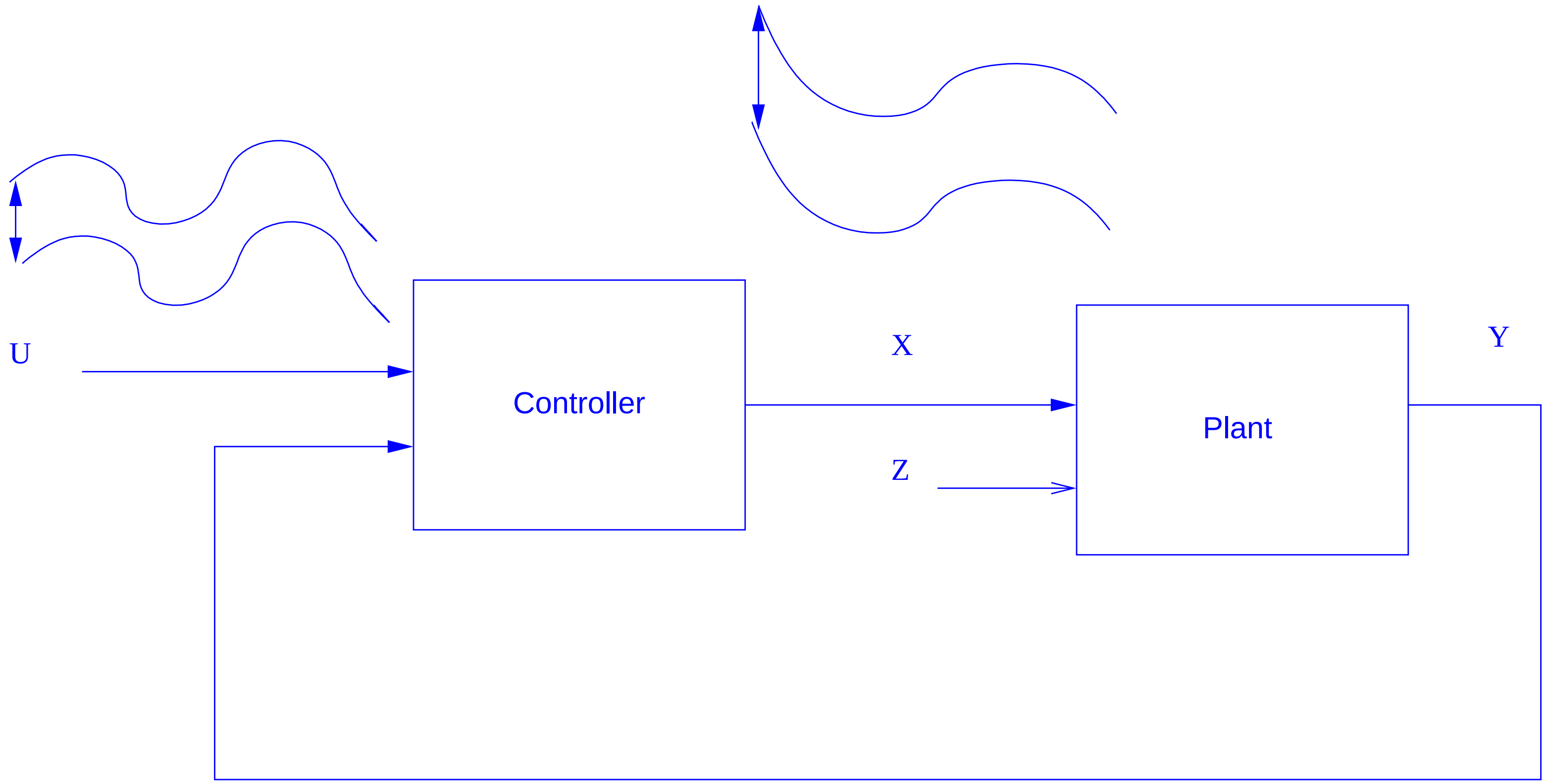
Consider the perfect control model as specifying a set of behaviours, those behaviours which are within some “distance” of the perfect model behaviour.

This requires some notion of “distance”, able to account for

- perturbations
- modelling errors
- discretisation
- jitter and communication delays
- ...

Model-based design in computer science and control

How can we make them converge ??



Sampling discrete event and hybrid systems

Continuous control is implemented by periodic sampling (time-triggered)

- sampled-data control theory
- numerical analysis

Discrete event control is implemented by event triggered systems

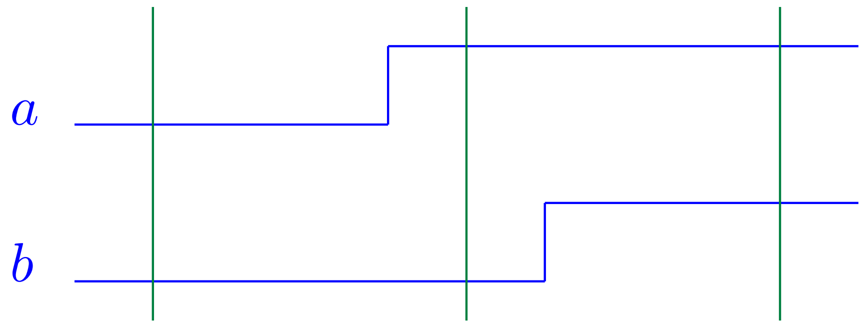
What about mixed (hybrid) systems ??

Experience shows that periodic sampling is a popular solution

but an empirical one

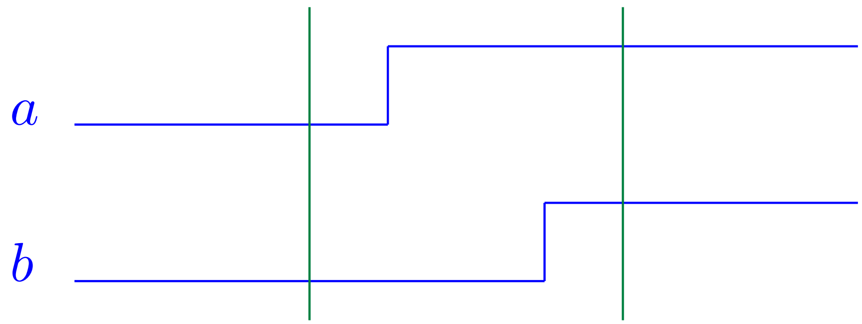
Sampling discrete event systems

A possible sampling



Sampling discrete event systems

Another one

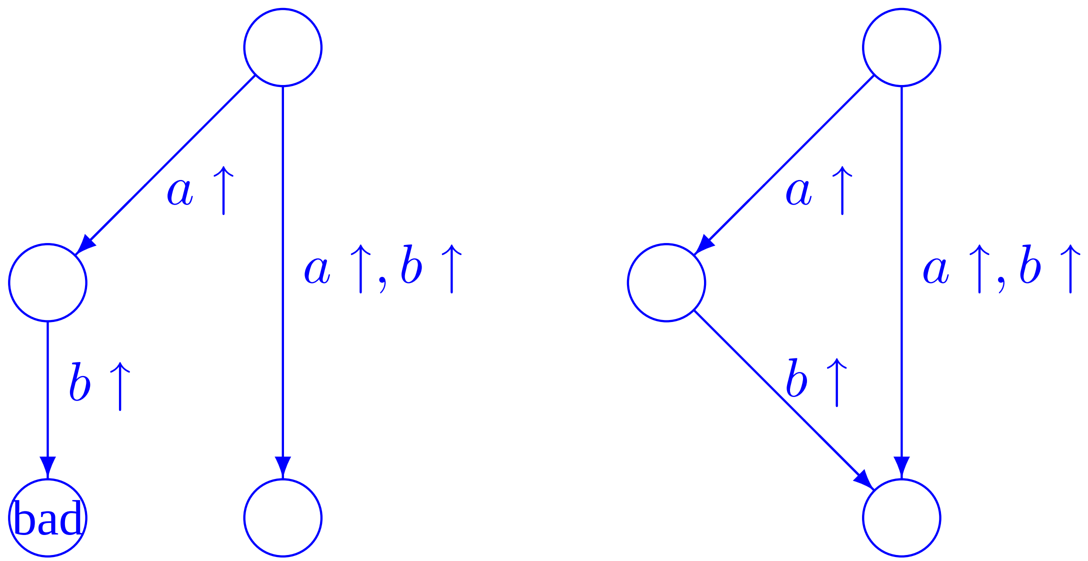


Races

A race takes place when two variables can change in distinct orders

A race is critical if different states can be reached according to which variable changes first

A critical race　　　　A non-critical race



Sampling discrete event and hybrid systems

Which kind of “distance” can account for

- perturbations
- modelling errors
- discretisation
- jitter and communication delays
- sampling discrete events
- races
- ...

Fault tolerance in GALS

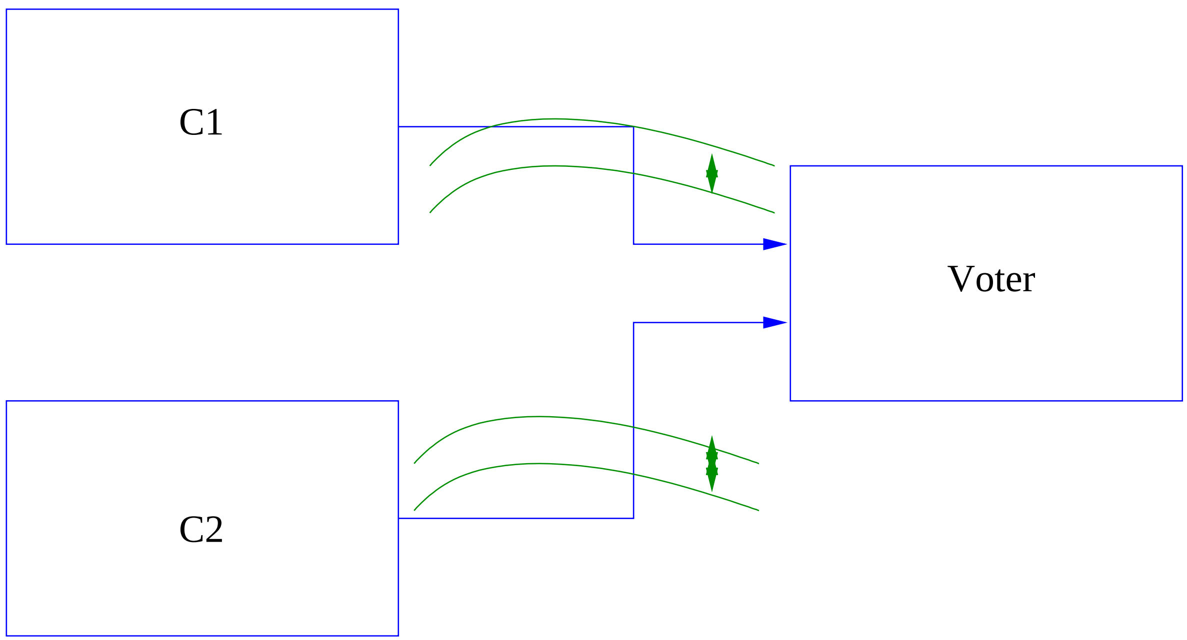
Communication "by periodic sampling" is a popular technique in distributed control

- autonomy, non blocking
- akin to shared memory

Example: Airbus "fly-by-wire" architecture

Entails a need for distance voting in fault-tolerant systems based on this communication principle

Distance voting



When do two signals differ for more than their “normal deviation” ?

Conclusion

This seems to show some need for

An approximation, sampling and voting theory for hybrid signals and systems

Conclusion

This seems to show some need for

An approximation, sampling and voting theory for hybrid signals and systems

Thank you for your attention