

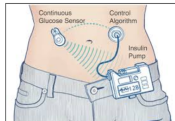
Dependable Hybrid Systems Design: Prelude

Zheng Cheng Dominique Méry

Nov, 2020

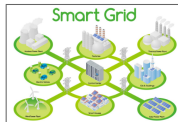
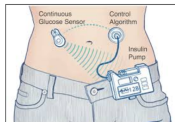
Outlines

Hybrid Systems



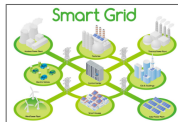
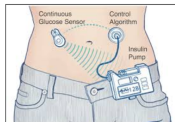
- **Hybrid** = Continuous + Discrete behaviors

Hybrid Systems



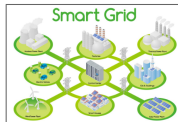
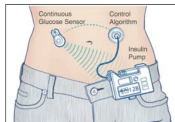
- ▶ **Hybrid** = Continuous + Discrete behaviors
- ▶ Open-loop vs. **Close-loop**

Hybrid Systems



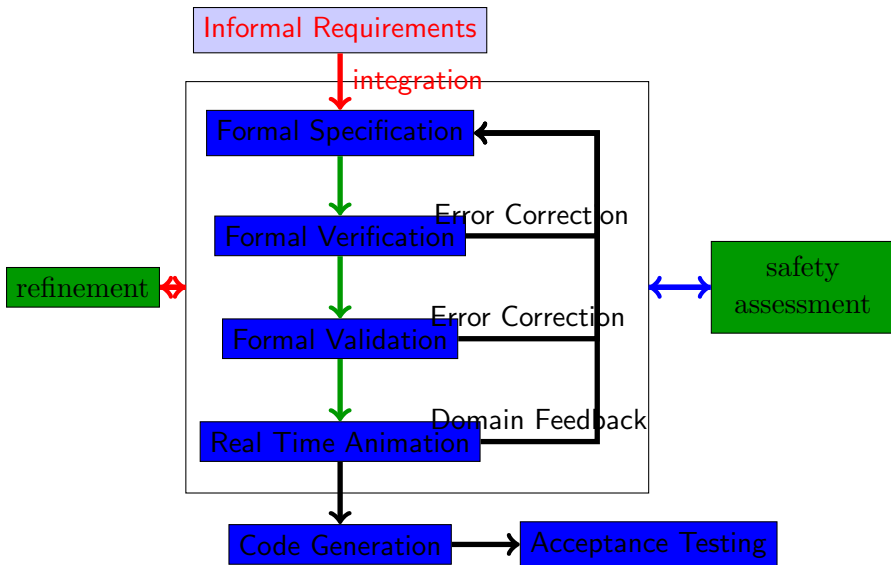
- ▶ **Hybrid** = Continuous + Discrete behaviors
- ▶ Open-loop vs. **Close-loop**
- ▶ Ubiquitous

Hybrid Systems



- ▶ **Hybrid** = Continuous + Discrete behaviors
- ▶ Open-loop vs. **Close-loop**
- ▶ Ubiquitous
- ▶ **Safety**

Tool Support for Design Safe Hybrid Systems



Tool Support for Design Safe Hybrid Systems

- ▶ Requirement engineering(formal specification)
- ▶ Theorem Proving(formal verification)
- ▶ Model Checking(formal validation)
- ▶ Simulation(real-time animation)

Requirement engineering

- ▶ Workflow
 - ▶ Rewriting informal requirement into formal specification(preconditions and postconditions)
- ▶ e.g. Event-B, Z, TLA+

Theroem proving

- ▶ Workflow
 - ▶ Hybrid system safety as a theorem:
 $\{\text{Precondition}\} \text{ System } \{\text{Postcondition}\}$
 - ▶ Interacting with a mechanized theorem prover to generate proofs
- ▶ e.g. KeYmaera, HHL, Z3
- ▶ Expensive, time consuming

Model Checking

- ▶ Workflow
 - ▶ Building model, specify its initial states of interest, and safety property
 - ▶ Checking every execution of model starting in an initial state always stays within the set of safe states
- ▶ e.g. HyTech, SpaceEx, Flow*
- ▶ Undecidable in general

Simulation

- ▶ Workflow
 - ▶ Building model(behaviors, environment, interactions)
 - ▶ Computational evaluation of a model instance over time
 - ▶ Checking evaluation result
- ▶ e.g. Ptolemy, Matlab(Simulink,Stateflow), SAMSON
- ▶ Inconclusive result (valid only on the chosen input)

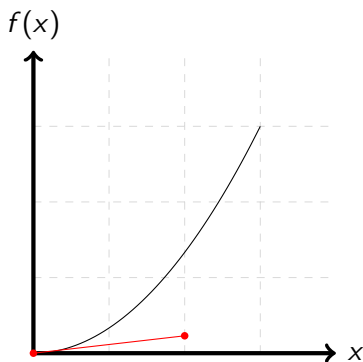
Outlines

Derivatives

- ▶ The rate of change of function $f(x)$, w.r.t. x

Derivatives

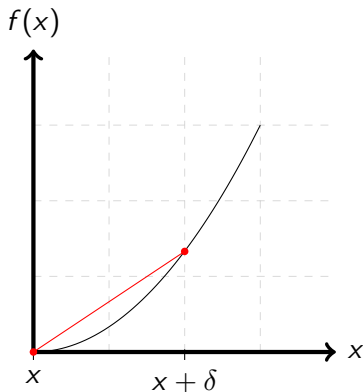
- ▶ The rate of change of function $f(x)$, w.r.t. x
- ▶ Slope of the tangent line!



Derivatives

- ▶ The rate of change of function $f(x)$, w.r.t. x
- ▶ Slope of the tangent line!
- ▶

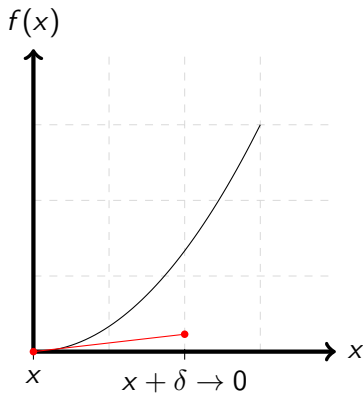
$$\frac{df}{dx} \approx \frac{f(x + \delta) - f(x)}{\delta}$$



Derivatives

- ▶ The rate of change of function $f(x)$, w.r.t. x
- ▶ Slope of the tangent line!
- ▶

$$\frac{df}{dx} = \lim_{\delta \rightarrow 0} \frac{f(x + \delta) - f(x)}{\delta}$$



Power Rule

Try $f(x) = x^n$, (Power Rule): $\frac{df}{dx} = nx^{n-1}$

$$\begin{aligned}\frac{df}{dx} &\approx \frac{f(x + \delta) - f(x)}{\delta} && \text{(approximation)} \\&= \frac{1}{\delta} [(x + \delta)^n - x^n] && \text{(rewriting)} \\&= \frac{1}{\delta} (x^n + nx^{n-1}\delta + \frac{n(n-1)}{2}x^{n-2}\delta^2 \dots - x^n) && \text{(Pascal-triangle)} \\&= \frac{1}{\delta} (nx^{n-1}\delta + O(\delta^2)) \\&= nx^{n-1} + O(\delta)\end{aligned}$$

Chain Rule

$$\frac{d}{dx}(f(g(x))) = \dot{f}(g(x)) \cdot \dot{g}(x)$$

Ex:

- ▶ $f(x) = \cos(x^3)$
- ▶ $f(x) = 2^x$
- ▶ $f(x) = e^{x^2} \sin(x)$

Differential Equations

Ex: x is the size of a population of procreate bunnies...

- ▶ Population grows at a rate λ proportional to its population size: $\frac{dx}{dt} = \lambda x$
- ▶ What is population as a function over time?

$$\frac{dx}{dt} = \lambda x$$

$$\rightarrow \frac{dx}{x} = \lambda dt$$

$$\rightarrow \int \frac{dx}{x} = \int \lambda dt$$

$$\rightarrow \ln(x) = \lambda t + C$$

$$\rightarrow x = e^{\lambda t + C}$$

- ▶ Determine C by initial conditions, e.g. (x_0 at t_0)

Outlines

Review of Event-B

- ▶ Context: static properties of Event-B models
 - ▶ Sets: user-defined types
 - ▶ Constants: static object in development
 - ▶ Axioms: presumed properties about sets and constants
 - ▶ Theorems: derived properties about sets and constants

Review of Event-B

- ▶ Machine: behavioral properties of Event-B models
 - ▶ Variables: states
 - ▶ Invariants: properties of variables that always need to hold
 - ▶ Theorems: derived properties about variables
 - ▶ Events: possible state changes

Review of Event-B

- ▶ Proof obligations: must be proved to show that Event-B models fulfill their specified properties.
 - ▶ INV: invariant preservation
 - ▶ FIS: action feasibility
 - ▶ ...

Develop Theories in Event-B

- ▶ **Theory plugin**: more modularize and reusable polymorphic “Context”
- ▶ Developed at University of Southampton, still under development
- ▶ Installation:
`http://rodin-b-sharp.sourceforge.net/updates`
 - ▶ Modelling Extensions → Theory Feature
- ▶ Let us develop a theory for real numbers
- ▶ Fork: `https://github.com/veriat1/LORIA_WEEK1`
 - ▶ Open model “theory-axiom-reals”

Exercise One (*)

► Prove: $a + b + c = c + b + a$ on real numbers

? How to write this theorem

? What is the key to prove this theorem

? How to use theory plugin to prove this

Exercise Two (**)

- ▶ Develop the power operator a^b
- ? What are its arguments and results
- ? What is its semantics

Exercise Three (***)

- ▶ Open model “ex-pattern-const-DE”
- ? What this model does
- ? What is its invariant
- ? What operators are needed to express this invariant, and what are their semantics
- ? How to prove your invariant

Caveats

- ▶ Axioms inconsistency \rightarrow Introduce when necessary, Prove when you can
- ▶ Big fat theories \rightarrow Modular theories