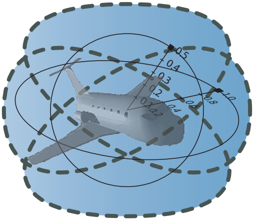
Teaching CPS Foundations With Contracts

André Platzer

aplatzer@cs.cmu.edu
Computer Science Department
Carnegie Mellon University, Pittsburgh, PA

http://symbolaris.com/course/fcps13.html

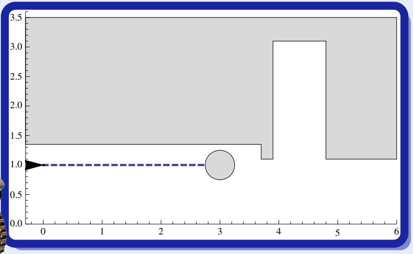


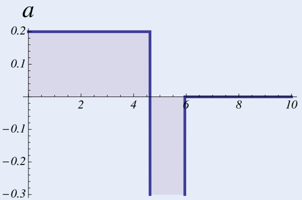
Can you trust a computer to control physics?

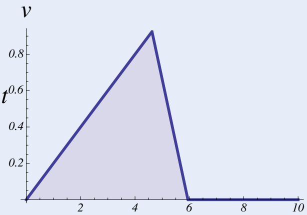
Challenge (Hybrid Systems)

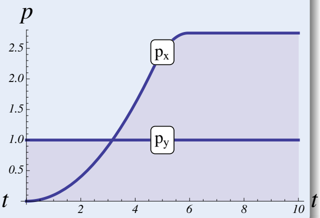
Design & verify controller for a robot avoiding obstacles

- Accelerate / brake (discrete dynamics)
- 1D motion (continuous dynamics)





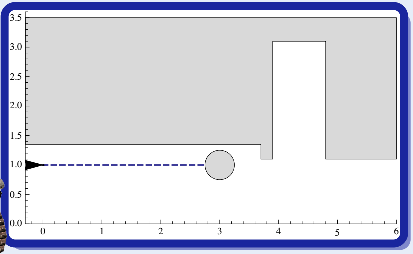


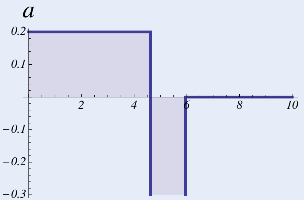


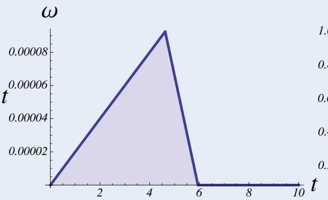
Challenge (Hybrid Systems)

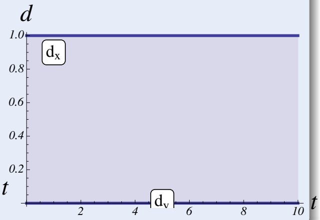
Design & verify controller for a robot avoiding obstacles

- Accelerate / brake (discrete dynamics)
- 1D motion (continuous dynamics)





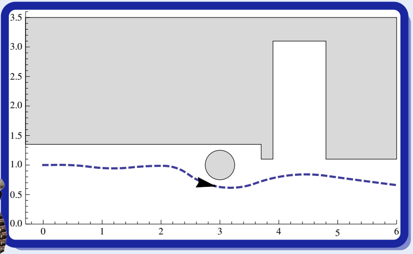


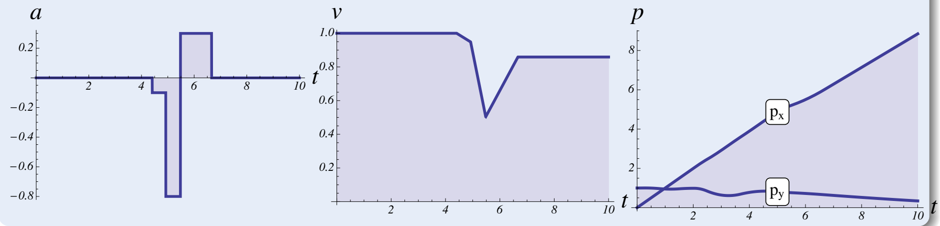


Challenge (Hybrid Systems)

Design & verify controller for a robot avoiding obstacles

- Accel / brake / steer (discrete dynamics)
- 2D motion (continuous dynamics)

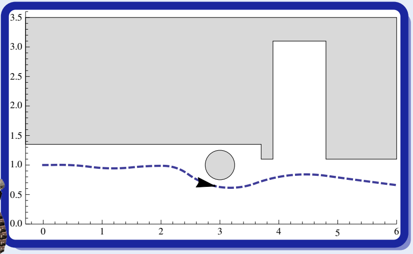


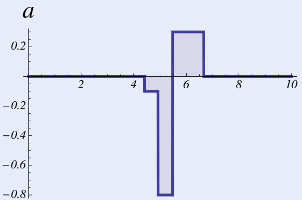


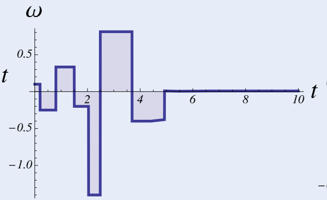
Challenge (Hybrid Systems)

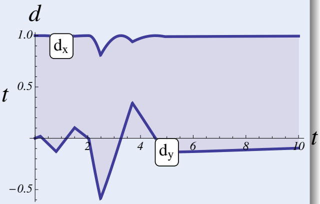
Design & verify controller for a robot avoiding obstacles

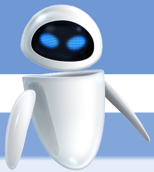
- Accel / brake / steer (discrete dynamics)
- 2D motion (continuous dynamics)







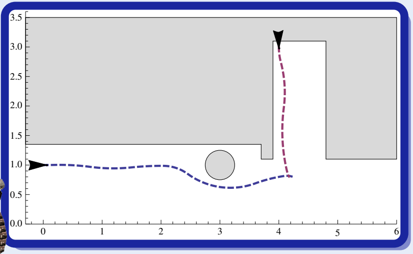


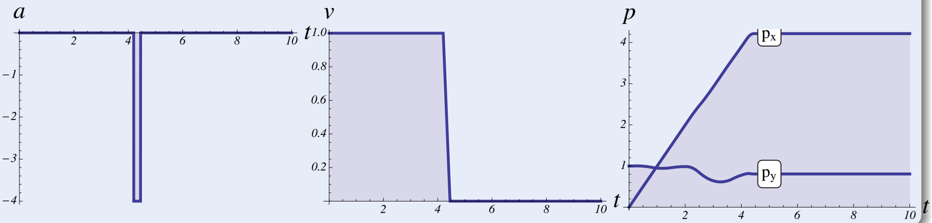


Challenge (Hybrid Systems)

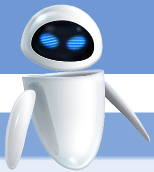
Design & verify controller for a robot avoiding obstacles

- Dynamic obstacles (other agents)
- Avoid collisions (define safety)





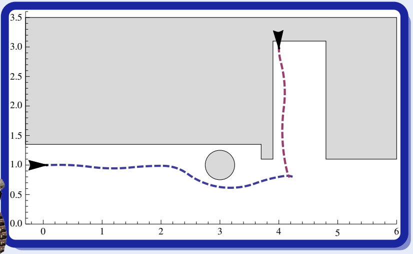
CPS Analysis & Design: Robot Lab

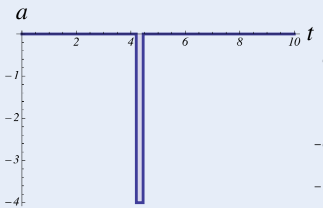


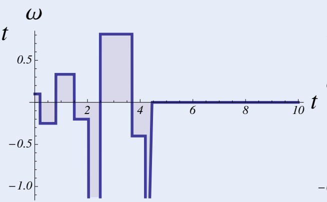
Challenge (Hybrid Systems)

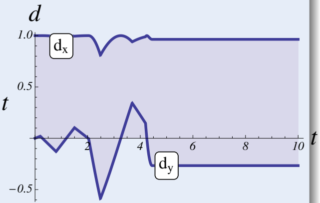
Design & verify controller for a robot avoiding obstacles

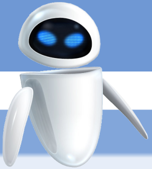
- Dynamic obstacles (other agents)
- Avoid collisions (define safety)







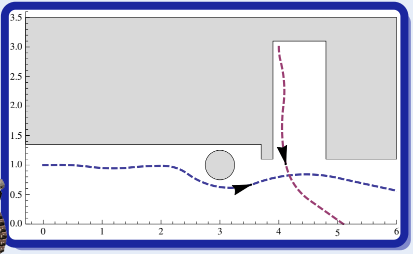


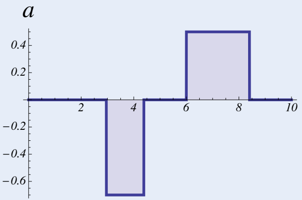


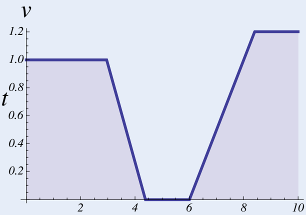
Challenge (Hybrid Systems)

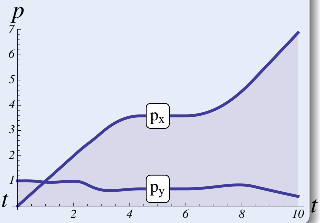
Design & verify controller for a robot avoiding obstacles

- Control robot (respect delays)
- Environment interaction (obstacles, agents, uncertainty)

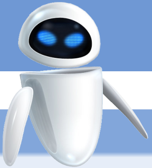








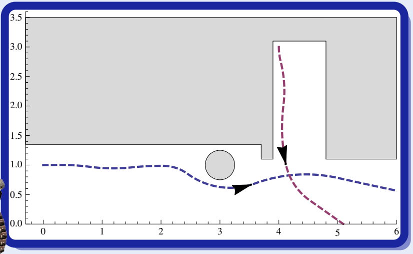
CPS Analysis & Design: Robot Lab

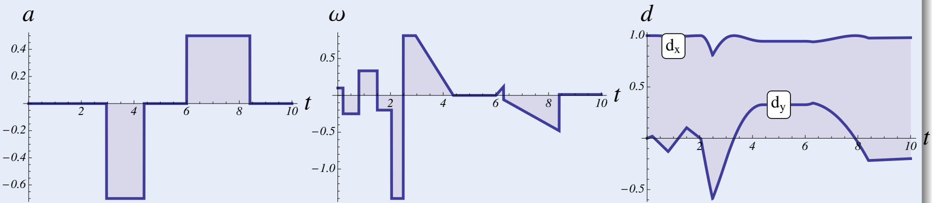


Challenge (Hybrid Systems)

Design & verify controller for a robot avoiding obstacles

- Control robot (respect delays)
- Environment interaction (obstacles, agents, uncertainty)





CPS Design & CPS Contracts in Programs

HP Reveal in layers

Contracts Reason about CPS

```
@requires(v^2 <= 2*b*(m-x))
@requires(v>=0 & A>=0 & b>0)
@ensures(x<=m)
  {
    if (v^2 <= 2*b*(m-x) - (A+b)*(A+2*v)) {
        a := A;
    } else {
        a := -b;
    }
    t := 0;
    {x'=v, v'=a, t'=1, v>=0 & t<=1}
  }*@invariant(v^2 <= 2*b*(m-x))
```

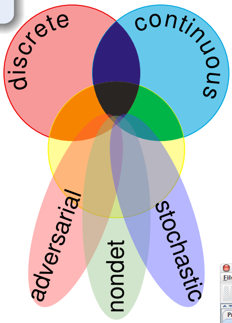
CPS Simulate for intuition

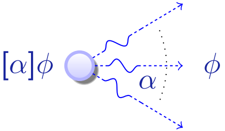
CT Design-by-invariant

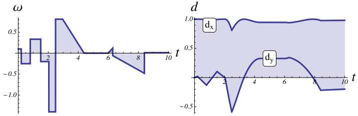
Teaching CPS Foundations With Contracts

differential dynamic logic

$d\mathcal{L} = DL + HP$

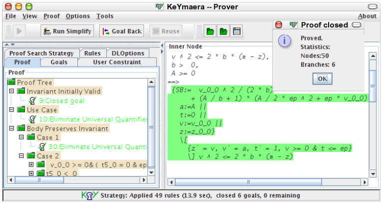


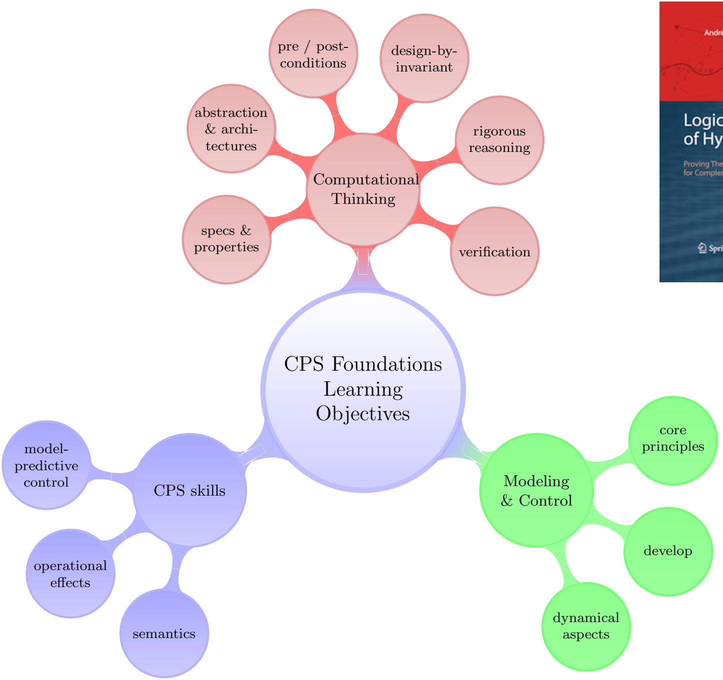


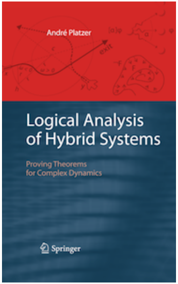


- Develop CPS models
- Express CPS contracts
- Intuition for operation
- Reason rigorously about CPS
- Focus on core principles
- CPS programs + contracts

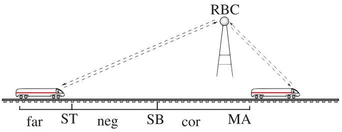
KeYmaera

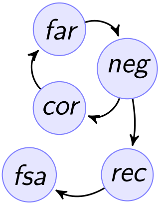


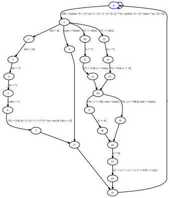

CPS Foundations Learning Objectives
Computational Thinking
pre / post-conditions
design-by-invariant
abstraction & architectures
rigorous reasoning
specs & properties
verification
CPS skills
model-predictive control
operational effects
semantics
Modeling & Control
core principles
develop
dynamical aspects

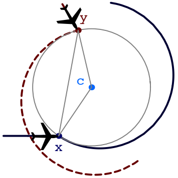

André Platzer
Logical Analysis of Hybrid Systems
Proving Theorems for Complex Dynamics
Springer

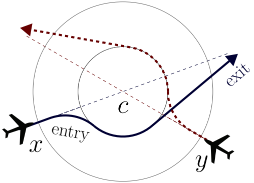
Successful Hybrid Systems Proofs

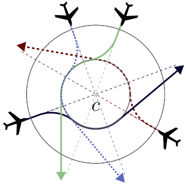


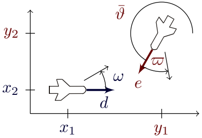


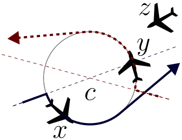


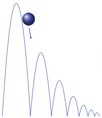


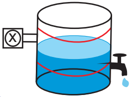




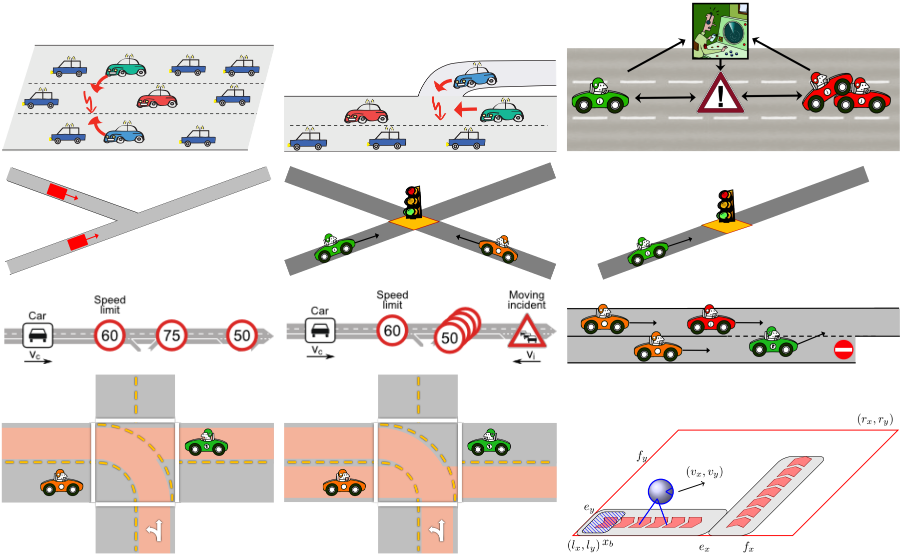




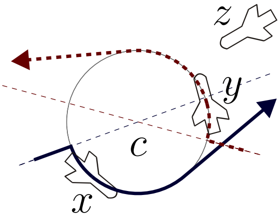


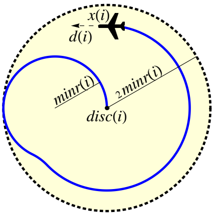


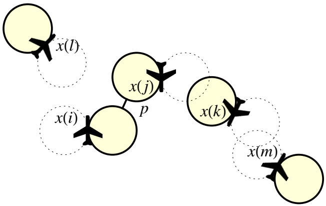
Successful Hybrid Systems Proofs

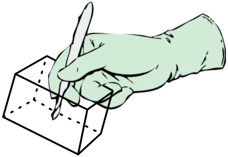


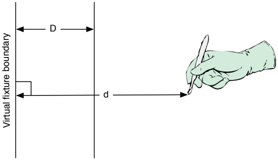
Successful Hybrid Systems Proofs

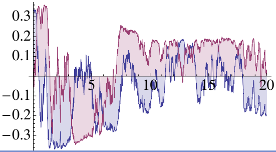


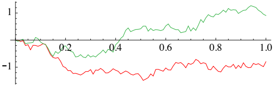


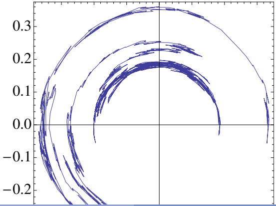






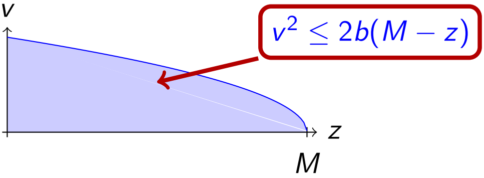






differential dynamic logic

$\mathrm{d}\mathcal{L} = \mathrm{FOL}_{\mathbb{R}}$

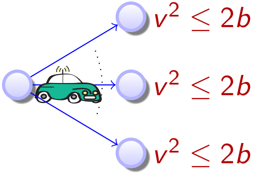


Logic for Hybrid Systems

differential dynamic logic

$\mathsf{d}\mathcal{L} = \mathrm{FOL}_{\mathbb{R}} + \mathrm{DL} + \mathrm{HP}$

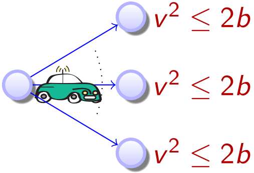
$$\mathcal{C} \to [\underbrace{\mathtt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \le 2b$$



differential dynamic logic

$d\mathcal{L} = \mathrm{FOL}_{\mathbb{R}} + \mathrm{DL} + \mathrm{HP}$

$$\mathcal{C} \rightarrow [\underbrace{\mathtt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \leq 2b$$



Initial condition

System dynamics

Post condition

Differential Dynamic Logic: Axiomatization

[:=]	$[x := \theta][(x)]\phi x \leftrightarrow [(x)]\phi\theta$
[?]	$[?H]\phi \leftrightarrow (H \rightarrow \phi)$
[′]	$[x' = f(x)]\phi \leftrightarrow \forall t{\geq}0\, [x := y(t)]\phi \qquad (y'(t) = f(y))$
[∪]	$[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$
[;]	$[\alpha; \beta]\phi \leftrightarrow [\alpha][\beta]\phi$
[*]	$[\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi$
K	$[\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$
I	$[\alpha^*](\phi \rightarrow [\alpha]\phi) \rightarrow (\phi \rightarrow [\alpha^*]\phi)$
C	$[\alpha^*]\forall v{>}0\, (\varphi(v) \rightarrow \langle\alpha\rangle\varphi(v-1)) \rightarrow \forall v\, (\varphi(v) \rightarrow \langle\alpha^*\rangle\exists v{\leq}0\, \varphi(v))$

André Platzer.
Differential dynamic logic for hybrid systems.
J. Autom. Reas., 41(2):143–189, 2008.

André Platzer.
Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.
Springer, Heidelberg, 2010.

André Platzer.
Logics of dynamical systems.
In *LICS*, pages 13–24. IEEE, 2012.

André Platzer and Jan-David Quesel.
KeYmaera: A hybrid theorem prover for hybrid systems.
In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *IJCAR*, volume 5195 of *LNCS*, pages 171–178. Springer, 2008.