Stefan Mitsch
Logical Systems Lab
Computer Science Department

2019 CyLab Partners Conference
Sept. 25th, 2019
Cyber-Physical Systems

Combine cyber capabilities with physical capabilities to solve problems that neither can solve alone

Challenges

- Interaction between computer and physics
- Verify models to guarantee safety in reality
- For autonomous systems with machine learning components
CPS and Learning Safety: Offline and Online Proofs

Real CPS

Testing and Simulation: Done when?

safe
CPS and Learning Safety: Offline and Online Proofs

Real CPS

Proof

Testing and Simulation: Done when?

safe
CPS and Learning Safety: Offline and Online Proofs

Real CPS

Control $\alpha_{ctrl}$

$$v := v + 1$$

Plant $\alpha_{plant}$

$$x' = v$$

abstract

Proof

Testing and Simulation: Done when?

safe

safe
CPS and Learning Safety: Offline and Online Proofs

Real CPS

abstract

\[ v := K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt} \]

\[ x' \leq v, \ v' = \frac{T_{exg} x_d n}{r_w} - \frac{1}{2} C_d \rho v^2 \]

Proof

Testing and Simulation: Done when?

safe
Real CPS

\[ v := K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt} \]

\[ x' \leq v, v' = \frac{T_{exg} x_d n}{r_w} - \frac{1}{2} C_d \rho v^2 \]

Proof

safe with proof transfer

Testing and Simulation: Done when?

safe
Formal Modeling, Verification, and Monitoring

Analyze the physical effect of software

Control

Sensors

Actuators

Hybrid System Model

Discrete computation + continuous physics
Theorem proving ensures correct model

Proof Strategy

Hybrid System Model

KeYmaera X

Code

Proof

Control Conditions

S. Mitsch—Formally Verified Learning CPS
Transfer safety of model to controller implementation

Research at the Intersection of

- Formal modeling and verification
- Proof automation and pragmatics
- Runtime verification and safe learning
- Control code synthesis
KeYmaera X Theorem Prover for Hybrid Systems

- Safe Reinforcement Learning
- Cross-verified in Isabelle and Coq
- Automation with tactics
- Differential equation reasoning
- ModelPlex: Verified runtime monitoring
- VeriPhy: Verified executables
- Component-based verification
- Adversarial dynamics (Games)
- Refinement

- AAAI’18, TACAS’19
- CPP’17
- ITP’17
- LICS’18, FM’19
- FMSD’16, arXiv
- PLDI’18
- FASE’17, STTT’18
- TCL’15, TCL’17
- FM’14, LICS’16
KeYmaera X Theorem Prover for Hybrid Systems

- **Safe Reinforcement Learning**
  - AAAI’18, TACAS’19

- Cross-verified in Isabelle and Coq
  - CPP’17

- Automation with tactics
  - ITP’17

- Differential equation reasoning
  - LICS’18, FM’19

- ModelPlex: Verified runtime monitoring
  - FMSD’16, arXiv

- **VeriPhy: Verified executables**
  - PLDI’18

- Component-based verification
  - FASE’17, STTT’18

- Adversarial dynamics (Games)
  - TCL’15, TCL’17

- Refinement
  - FM’14, LICS’16
On-Model Learning: ModelPlex Safety Margins

Reinforcement Learning

- Model-based Verification: find out what is safe
- Model-free Learning: find out what is useful
- Reward for achieving task while staying safe
- Safety proof includes tolerances to allow for optimization
VeriPhy: Automatic, Verified EXEs from Controllers

Hybrid Systems
Theorem Proving
VeriPhy: Automatic, Verified EXEs from Controllers

Hybrid Systems
Theorem Proving

Cyber Physical System
VeriPhy: Automatic, Verified EXEs from Controllers

Hybrid Systems
Theorem Proving

Abstract Controllers
and Monitors

Sound Discrete
Arithmetic

Sound Monitor
Compilation

Cyber Physical System
VeriPhy: Automatic, Verified EXEs from Controllers

- Hybrid Systems Theorem Proving
- Abstract Controllers and Monitors
- Sound Discrete Arithmetic
- Sound Monitor Compilation
- Cyber Physical System

- Small Prover Core Proven Sound
- Provably Correct Monitoring Conditions
- Formalized Soundness Theorem
- Verified Compiler
- Verified Executable
VeriPhy: Automatic, Verified EXEs from Controllers

Hybrid Systems Theorem Proving

Abstract Controllers and Monitors

Sound Discrete Arithmetic

Sound Monitor Compilation

Cyber Physical System

Small Prover Core Proven Sound

Provably Correct Monitoring Conditions

Formalized Soundness Theorem

Verified Compiler

Verified Executable

KeYmaera X

Isabelle/HOL

HOL4
KeYmaera X Specifications in Differential Dynamic Logic

\[
\left( (if \ S(d) \ then \ v := \star; \ ?0 \leq v \leq V \ else \ v := 0); \ (d' = -v \ & \ Q) \right)^* 
\]

Control Envelope

Physics
VeriPhy: Automatic, Verified EXEs from Controllers

KeYmaera X Specifications in Differential Dynamic Logic

\[ A \rightarrow \left( (\text{if } S(d) \quad v := \ast ; \quad 0 \leq v \leq V \quad \text{else} \quad v := 0) ; \quad (d' = -v \& Q) \right)^* \]  
\[ d \geq 0 \]
On-model (monitor satisfied): use learning to optimize for performance

Off-model (monitor violated): learn recovery policy
Summary

Proof guarantees correct model

Proof Strategy
Hybrid System
Model
KeYmaera X
Code
Control
Conditions
Proof
Control

Sensors Actuators

Logical foundations make a big difference for CPS
CPS deserve proofs as safety evidence!

- Strong analytic foundations
- Proof automation
- Safe control learning
- Control code synthesis
- Runtime monitoring

KeYmaera X
keymaeraX.org
smitsch@cs.cmu.edu