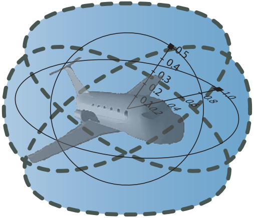
14: Verified Models & Verified Runtime Validation

15-424: Foundations of Cyber-Physical Systems

Stefan Mitsch André Platzer

Computer Science Department
Carnegie Mellon University, Pittsburgh, PA

Simplex for Hybrid System Models (FMSD'16)



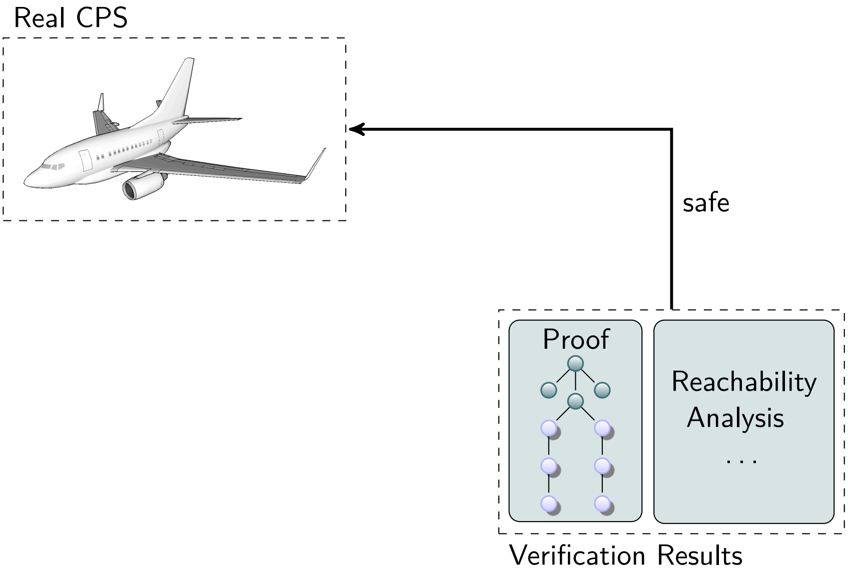
Outline

1. Motivation
2. Learning Objectives
3. ModelPlex Runtime
   - ModelPlex Runtime
   - ModelPlex Compliance
4. ModelPlex
   - Logical State Relations
   - Model Monitors
   - Correct-by-Construction Synthesis
   - Example: Water Tank
   - Controller Monitors
   - Prediction Monitors
5. Evaluation
6. Summary

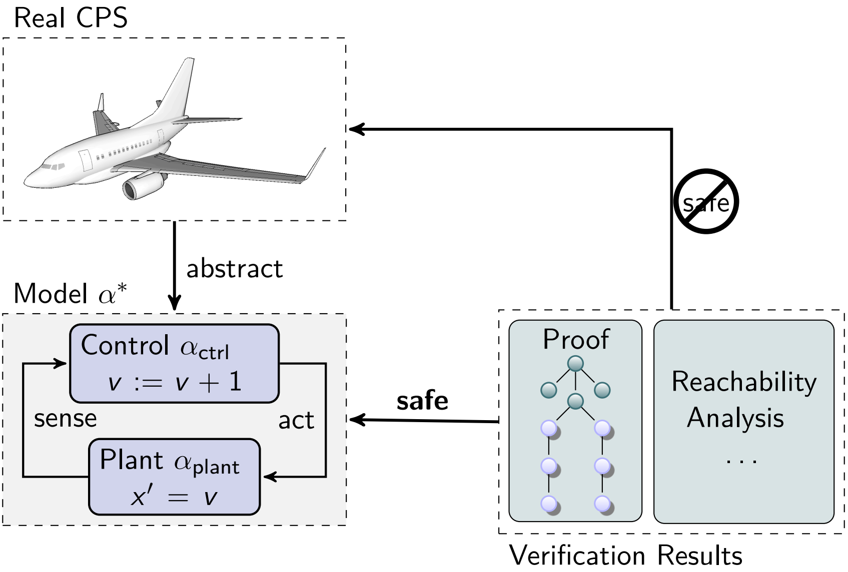
Outline

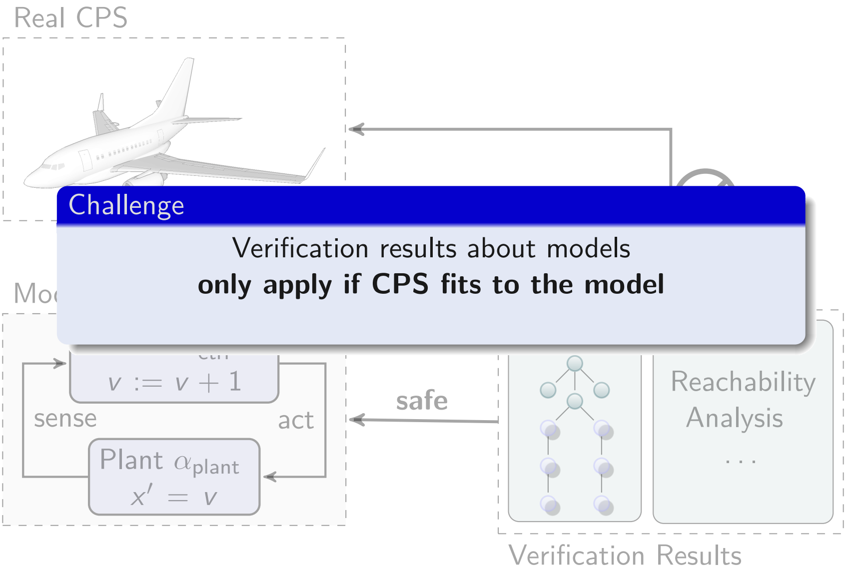
1. Motivation
2. Learning Objectives
3. ModelPlex Runtime
   - ModelPlex Runtime
   - ModelPlex Compliance
4. ModelPlex
   - Logical State Relations
   - Model Monitors
   - Correct-by-Construction Synthesis
   - Example: Water Tank
   - Controller Monitors
   - Prediction Monitors
5. Evaluation
6. Summary

Formal Verification in CPS Development



Formal Verification in CPS Development

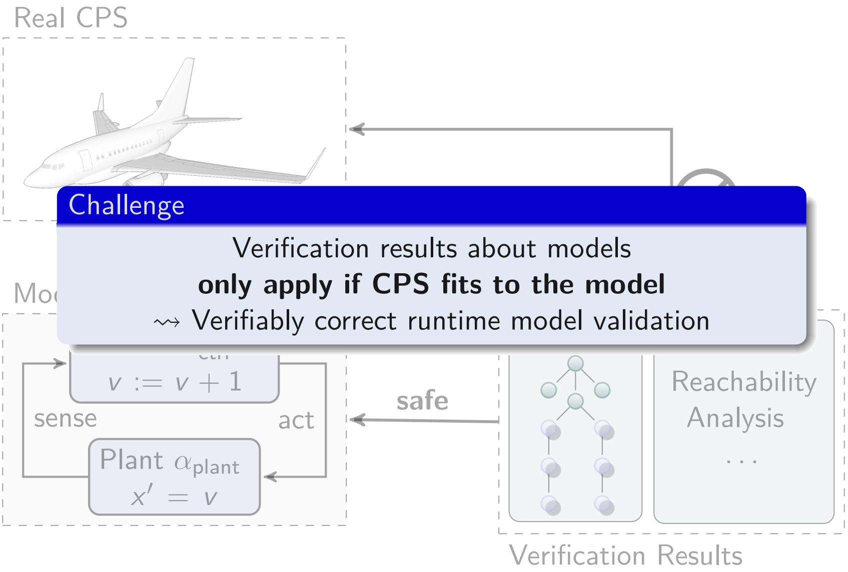




Challenge

Verification results about models

only apply if CPS fits to the model



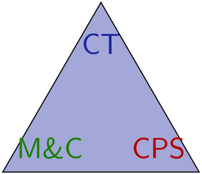
Challenge

Verification results about models
only apply if CPS fits to the model
$\rightsquigarrow$ Verifiably correct runtime model validation

Outline

1. Motivation
2. Learning Objectives
3. ModelPlex Runtime
   - ModelPlex Runtime
   - ModelPlex Compliance
4. ModelPlex
   - Logical State Relations
   - Model Monitors
   - Correct-by-Construction Synthesis
   - Example: Water Tank
   - Controller Monitors
   - Prediction Monitors
5. Evaluation
6. Summary

proof in a model vs. truth in reality
tracing assumptions
turning provers upside down
correct-by-construction
dynamic contracts
proofs for CPS implementations



models vs. reality
inevitable differences
model compliance
architectural design

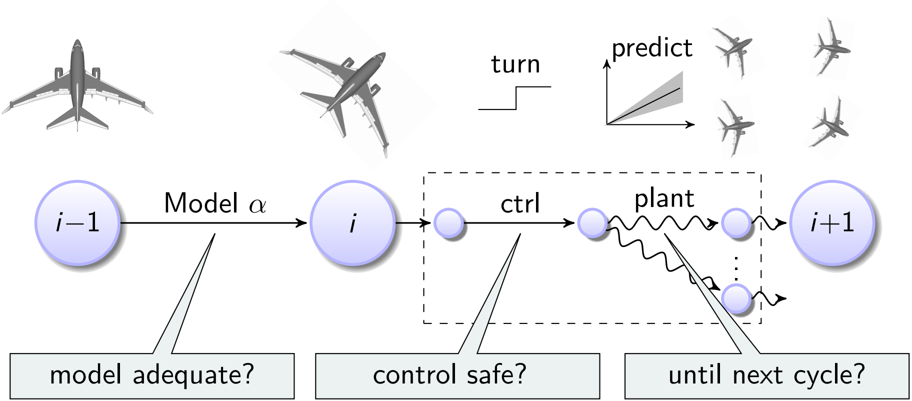
tame CPS complexity
prediction vs. run
runtime validation
online monitor

Outline

1. Motivation
2. Learning Objectives
3. ModelPlex Runtime
   - ModelPlex Runtime
   - ModelPlex Compliance
4. ModelPlex
   - Logical State Relations
   - Model Monitors
   - Correct-by-Construction Synthesis
   - Example: Water Tank
   - Controller Monitors
   - Prediction Monitors
5. Evaluation
6. Summary

ModelPlex Runtime Model Validation

ModelPlex **ensures that verification results** about models **apply to CPS** implementations



ModelPlex Runtime Model Validation

ModelPlex **ensures that verification results** about models **apply to CPS** implementations

Contributions

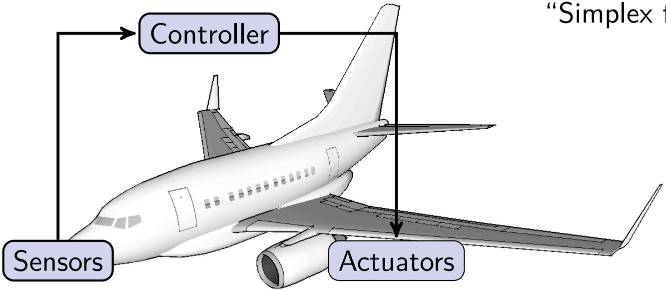
- Verification results about models transfer to CPS when validating model compliance
- Compliance with model is characterizable in logic
- Compliance formula transformed by proof to executable monitor
- Correct-by-construction provably correct runtime model validation

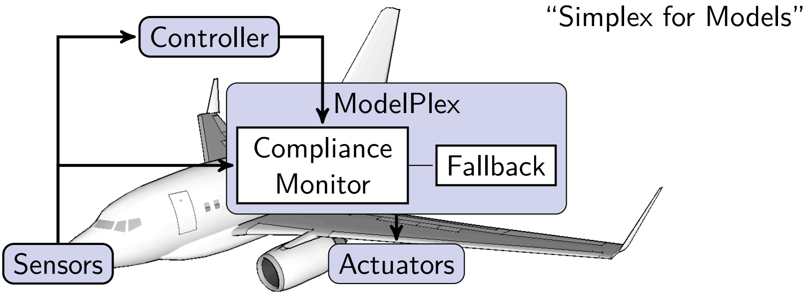
model adequate?

control safe?

until next cycle?

"Simplex for Models"





Compliance Monitor Checks CPS for compliance with model at runtime

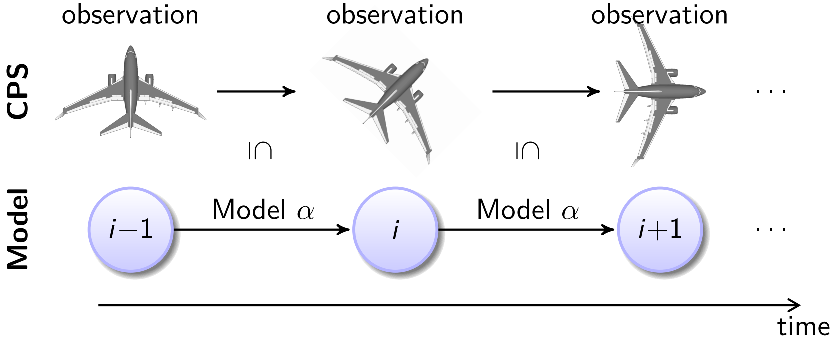
- Model Monitor: model adequate?
- Controller Monitor: control safe?
- Prediction Monitor: until next cycle?

Fallback Safe action, executed when monitor is not satisfied (veto)

Challenge What conditions do the monitors need to check to be safe?

Is current CPS behavior included in the behavior of the model?

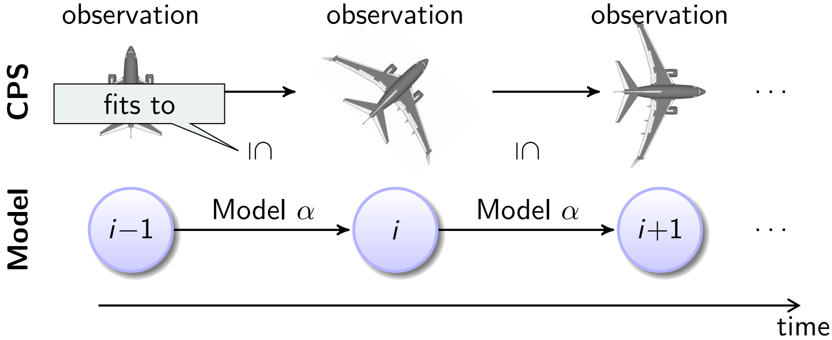
- CPS observed through sensors
- Model describes behavior of CPS between states



Detect non-compliance ASAP to initiate fallback actions while still safe

Is current CPS behavior included in the behavior of the model?

- CPS observed through sensors
- Model describes behavior of CPS between states



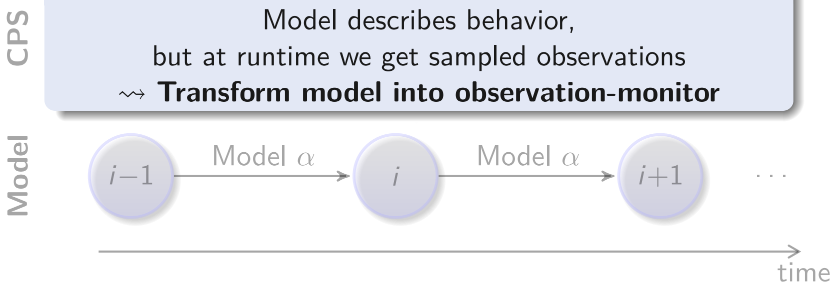
Detect non-compliance ASAP to initiate fallback actions while still safe

Is current CPS behavior included in the behavior of the model?

- CPS observed through sensors
- Model describes behavior of CPS between states

Challenge

Model describes behavior,
but at runtime we get sampled observations
$\leadsto$ **Transform model into observation-monitor**

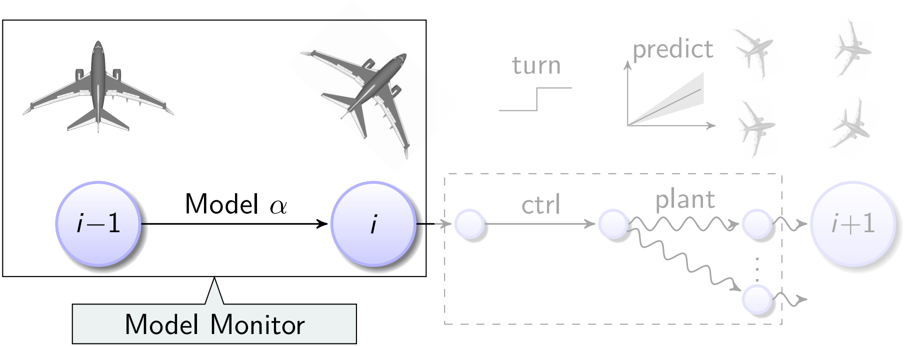


Detect non-compliance ASAP to initiate fallback actions while still safe

Outline

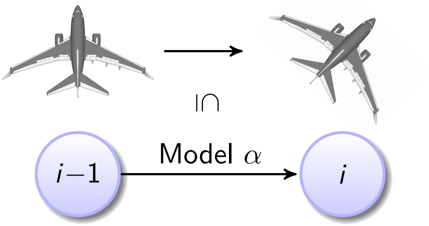
1. Motivation
2. Learning Objectives
3. ModelPlex Runtime
   - ModelPlex Runtime
   - ModelPlex Compliance
4. ModelPlex
   - Logical State Relations
   - Model Monitors
   - Correct-by-Construction Synthesis
   - Example: Water Tank
   - Controller Monitors
   - Prediction Monitors
5. Evaluation
6. Summary

Outline



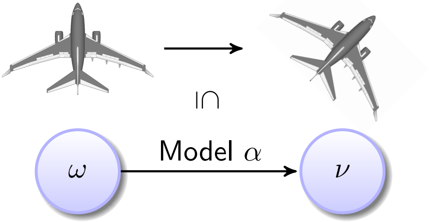
Characterizing State Relations in Logic

When are two states linked through a run of model α?

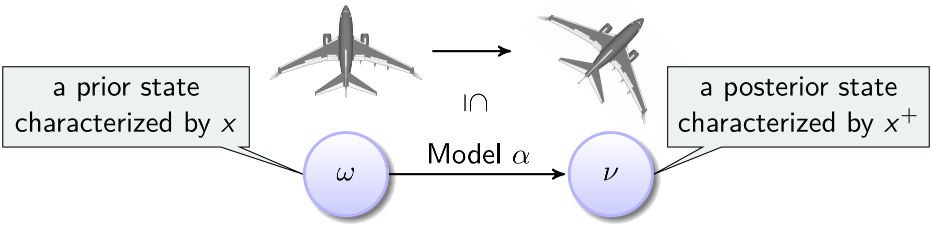


Characterizing State Relations in Logic

When are two states linked through a run of model α?



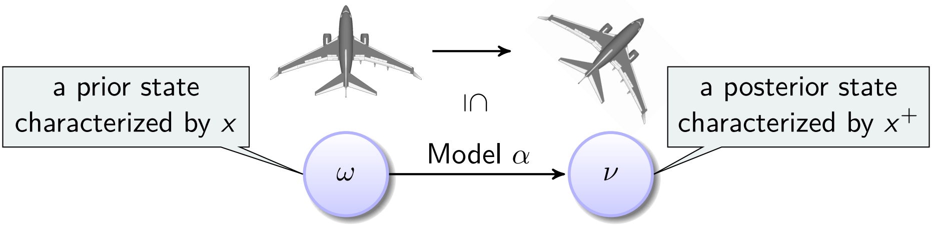
When are two states linked through a run of model α?



Semantical: $(\omega, \nu) \in [\![\alpha]\!]$ — reachability relation of α

Characterizing State Relations in Logic

When are two states linked through a run of model α?



Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

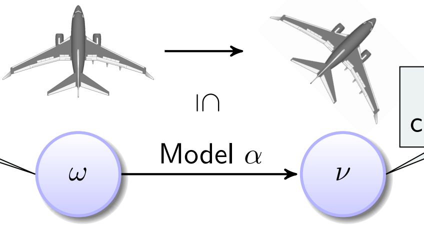
$\Updownarrow$ Lemma

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

exists a run of α to a state where $x = x^+$

When are two states linked through a run of model α?

a prior state characterized by x



a posterior state characterized by x^+

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle \alpha \rangle (x = x^+)$

exists a run of α to a state where $x = x^+$

$\Updownarrow$ d$\mathcal{L}$ proof

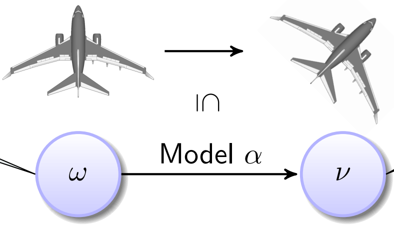
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Characterizing State Relations in Logic

When are two states linked through a run of model α?

a prior state characterized by x



a posterior state characterized by x^+

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

exists a run of α to a state where $x = x^+$

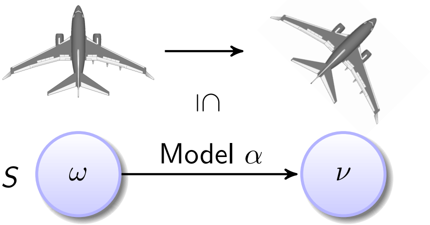
$\Uparrow$ d$\mathcal{L}$ proof

Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof

d$\mathcal{L}$ proof $A \to [\alpha]S$



Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

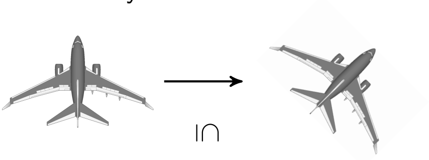
Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ d$\mathcal{L}$ proof

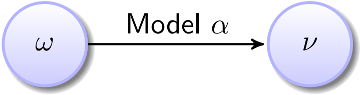
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



d$\mathcal{L}$ proof $A \to [\alpha]S$



Offline

Init $\omega \models A$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

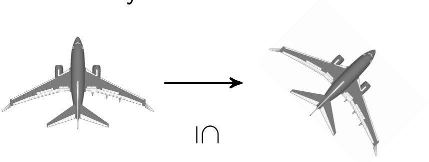
Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ d$\mathcal{L}$ proof

Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



$\subseteq$

d$\mathcal{L}$ proof $A \to [\alpha]S$ — ω —Model α→ ν

Offline

Init $\omega \models A$ Safe $\nu \models S$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

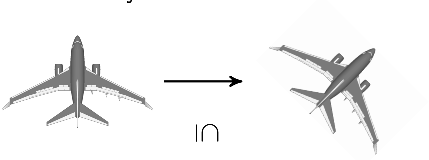
Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ d$\mathcal{L}$ proof

Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



$\subseteq$

dℒ proof $A \rightarrow [\alpha]S$ ω —Model α→ ν

Offline

Init $\omega \models A$ Safe $\nu \models S$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

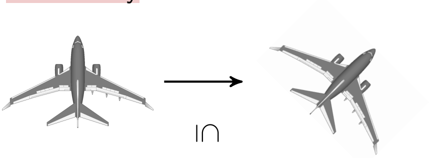
Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ dℒ proof

Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



$\cap\!\!\mid$

dℒ proof $A \to [\alpha]S$ — ω —Model α→ ν

Offline

Init $\omega \models A$ Safe $\nu \models S$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

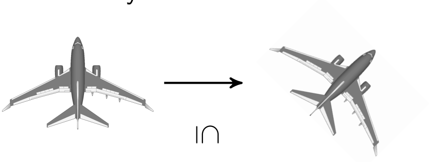
$\Uparrow$ dℒ proof

Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logical Reductions for Model Safety Transfer

Logic reduces CPS safety to runtime monitor with offline proof



$\subseteq$

dℒ proof $A \rightarrow [\alpha]S$ $\quad \omega \xrightarrow{\text{Model } \alpha} \nu$

Offline

Init $\omega \models A$ Safe $\nu \models S$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

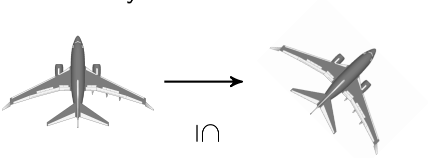
Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ dℒ proof

Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



$\subseteq$

dL proof $A \to [\alpha]S$ (ω) — Model α → (ν)

Offline

Init $\omega \models A$ Safe $\nu \models S$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

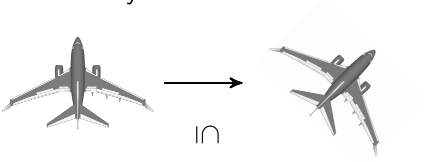
Logical dL: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ dL proof

Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



$\subseteq$

dℒ proof $A \rightarrow [\alpha]S$ ω —Model α→ ν

Offline

Init $\omega \models A$ Safe $\nu \models S$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

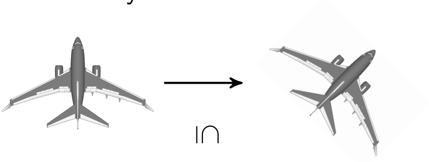
Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ dℒ proof

Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



Not initial state. Model repeats...

dℒ proof $A \to [\alpha]S$

ω —Model α→ ν

Offline

Init $\omega \models A$ Safe $\nu \models S$

Semantical: $(\omega,\nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

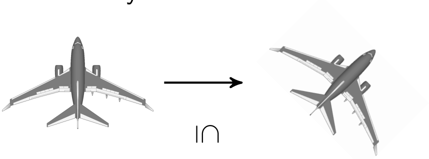
Logical dℒ: $(\omega,\nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ dℒ proof

Arithmetical: $(\omega,\nu) \models F(x,x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



$\subseteq$

dℒ proof $A \rightarrow [\alpha^*]S$ — ω —Model α^*→ ν

Init $\omega \models A$ Safe $\nu \models S$

Offline

Semantical: $(\omega, \nu) \in [\![\alpha^*]\!]$

$\Updownarrow$ Lemma

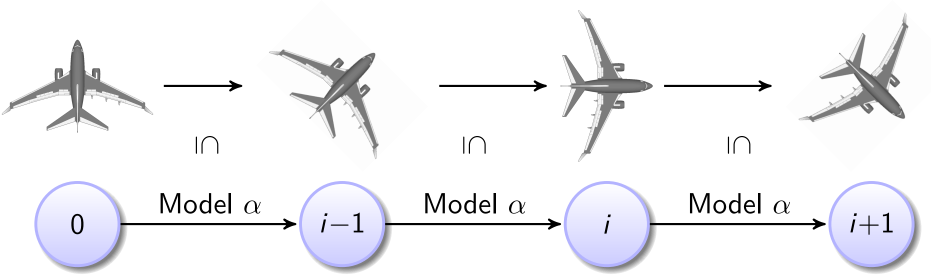
Logical dℒ: $(\omega, \nu) \models \langle \alpha^* \rangle (x = x^+)$

$\Uparrow$ dℒ proof

Arithmetical: $(\omega, \nu) \models F(x, x^+)$

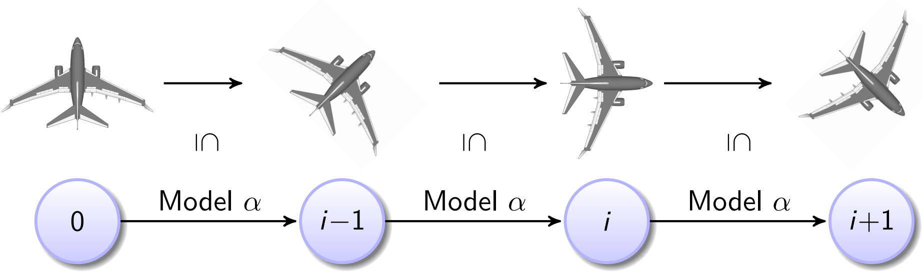
check at runtime (efficient)

ModelPlex Model Monitor Correctness

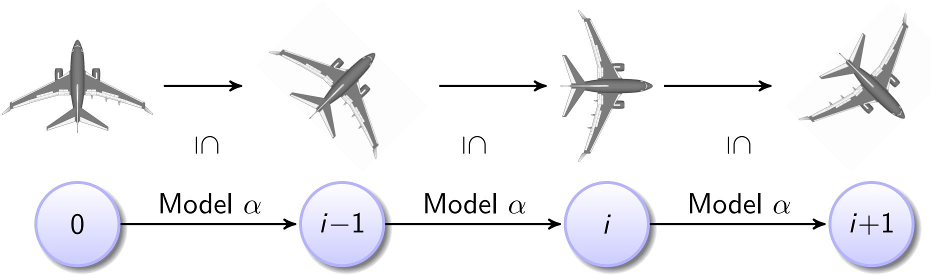


ModelPlex Model Monitor Correctness

d$\mathcal{L}$ proof $A \rightarrow [\alpha^*]S$

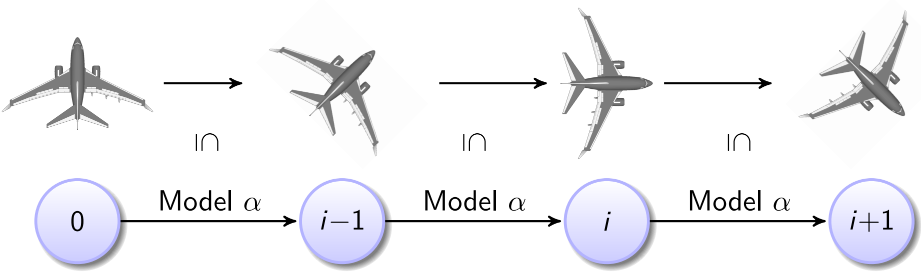


dℒ proof $A \rightarrow [\alpha^*]S$



Init $0 \models A$

d$\mathcal{L}$ proof $A \rightarrow [\alpha^*]S$

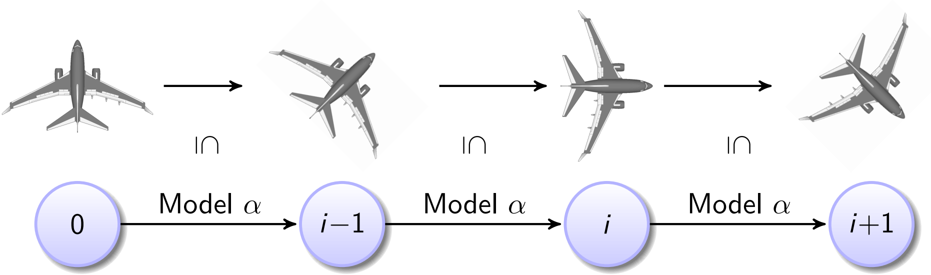


Init $0 \models A$

Check $(i, i+1) \models \langle\alpha\rangle x = x^+$

ModelPlex Model Monitor Correctness

d$\mathcal{L}$ proof $A \rightarrow [\alpha^*]S$

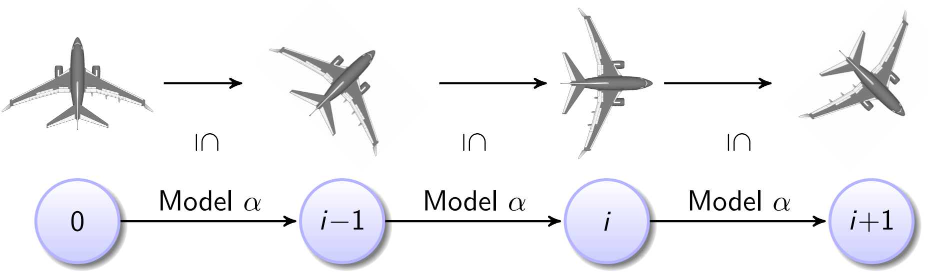


Init $0 \models A$

Check $(i, i{+}1) \models \langle\alpha\rangle x{=}x^+$

Safe $i{+}1 \models S$

d$\mathcal{L}$ proof $A \rightarrow [\alpha^*]S$



Init $0 \models A$

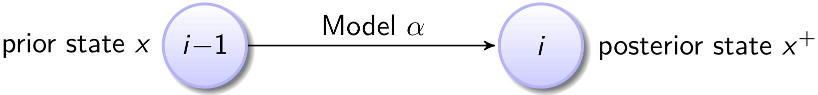
Check $(i, i{+}1) \models \langle\alpha\rangle x{=}x^+$

Safe $i{+}1 \models S$

Theorem (Model Monitor Correctness)

"System safe as long as monitor satisfied." *FMSD'16*

- Proof calculus of d$\mathcal{L}$ executes models symbolically

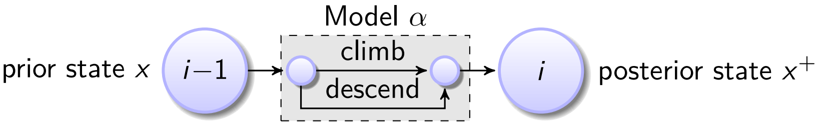


proof attempt

- $\langle \alpha_{(x)} \rangle (x = x^+)$

Provably Correct Synthesis of Monitors

- Proof calculus of d$\mathcal{L}$ executes models symbolically

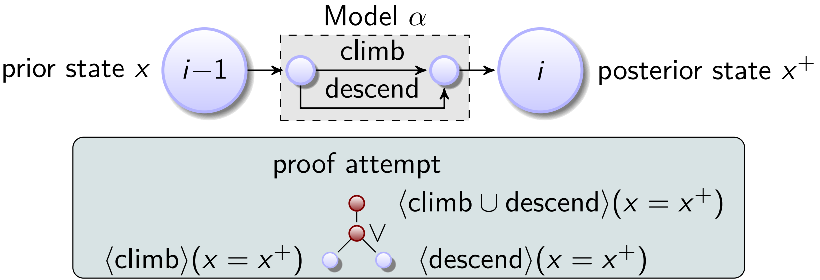


proof attempt

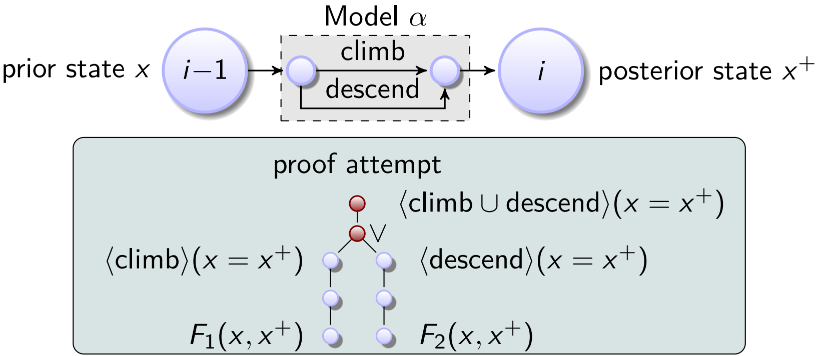
$\langle \text{climb} \cup \text{descend} \rangle (x = x^+)$

$\langle \text{climb} \cup \text{descend} \rangle P \leftrightarrow$
$\langle \text{climb} \rangle P \lor \langle \text{descend} \rangle P$

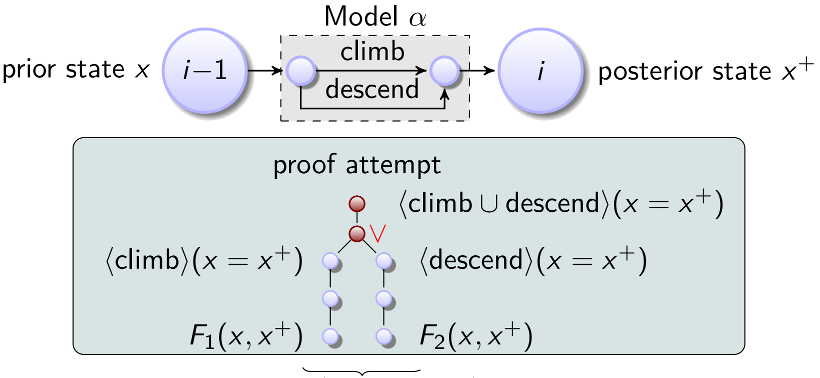
- Proof calculus of d$\mathcal{L}$ executes models symbolically



- Proof calculus of d$\mathcal{L}$ executes models symbolically

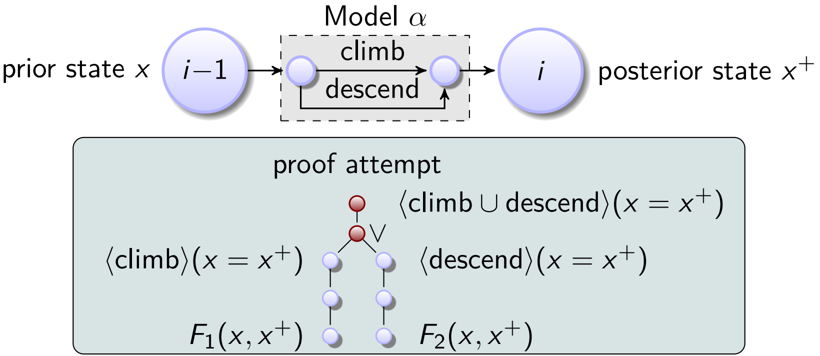


- Proof calculus of d$\mathcal{L}$ executes models symbolically



Monitor: $F_1(x, x^+) \vee F_2(x, x^+)$

- Proof calculus of d$\mathcal{L}$ executes models symbolically

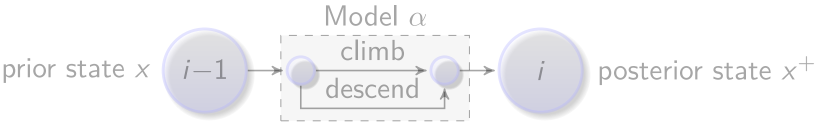


Monitor: $F_1(x, x^+) \vee F_2(x, x^+)$

- The subgoals that cannot be proved express all the conditions on the relations of variables imposed by the model $\rightsquigarrow$ close at runtime

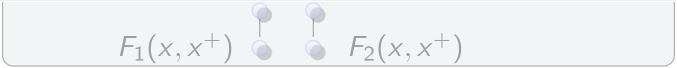
Provably Correct Synthesis of Monitors

- Proof calculus of d$\mathcal{L}$ executes models symbolically



Model Monitor

Immediate detection of model violation
$\rightsquigarrow$ Mitigates safety issues with safe fallback action



Monitor: $\underbrace{F_1(x, x^+) \lor F_2(x, x^+)}$

- The subgoals that cannot be proved express all the conditions on the relations of variables imposed by the model $\rightsquigarrow$ close at runtime

Water Tank Example: Monitor Conjecture

Variables

x	current level	ε	control cycle
m	maximum level	f	flow

Model and Safety Property

$$\underbrace{0 \le x \le m \land \varepsilon > 0}_{A} \to \Big[\Big(f := *;\ ?\left(-1 \le f \le \tfrac{m-x}{\varepsilon}\right);\ t := 0;\ (x' = f,\ t' = 1\ \&\ x \ge 0 \land t \le \varepsilon)\Big)^* \Big] \underbrace{(0 \le x \le m)}_{S}$$

Model Monitor Specification Conjecture

$$\underbrace{\varepsilon > 0}_{A|_{\text{const}}} \to \Big\langle f := *; ?\left(-1 \le f \le \tfrac{m-x}{\varepsilon}\right);\ t := 0; (x' = f, t' = 1\ \&\ x \ge 0 \land t \le \varepsilon)\Big\rangle \overbrace{(x{=}x^+ \land f{=}f^+ \land t{=}t^+)}^{\Upsilon^+_{V_m}}$$

Water Tank Example: Nondeterministic Assignment

Proof Rules

$$\langle *\rangle \ \frac{\Gamma \vdash \exists X \langle x := X\rangle P, \Delta}{\Gamma \vdash \langle x := *\rangle P, \Delta} \quad (X \text{ is a new logical variable})$$

$$\exists\mathrm{R} \ \frac{\Gamma \vdash p(e), \exists x\, p(x), \Delta}{\Gamma \vdash \exists x\, p(x), \Delta} \quad (e \text{ is any arbitrary term})$$

$$\mathrm{WR} \ \frac{\Gamma \vdash \Delta}{\Gamma \vdash \phi, \Delta}$$

Sequent Deduction

w/o Opt. 1

$$\langle *\rangle \ \dfrac{\exists\mathrm{R,WR} \ \dfrac{A \vdash \langle f := F\rangle\langle ?{-1} \le f \le \frac{m-x}{\varepsilon}\rangle\langle \text{plant}\rangle\Upsilon^+}{A \vdash \exists F \langle f := F\rangle\langle ?{-1} \le f \le \frac{m-x}{\varepsilon}\rangle\langle \text{plant}\rangle\Upsilon^+}}{A \vdash \langle f := *; ?{-1} \le f \le \frac{m-x}{\varepsilon}\rangle\langle \text{plant}\rangle\Upsilon^+}$$

$$\exists\mathrm{R,WR} \ \dfrac{A \vdash \langle f := f^+\rangle \langle ?{-1} \le f \le \frac{m-x}{\varepsilon}\rangle\langle \text{plant}\rangle\Upsilon^+}{\dots}$$

with Opt. 1 (anticipate $f = f^+$ from Υ^+)

Water Tank Example: Differential Equations

Proof Rules

$$\langle ' \rangle \quad \frac{\exists T{\geq}0\, ((\forall 0{\leq}t{\leq}T\, \langle x := y(t)\rangle Q) \land \langle x := y(T)\rangle P)}{\langle x' = f(x)\,\&\, Q\rangle P} (y(t) \text{ solution } T, t \text{ new})$$

$$\text{QE} \quad \frac{\text{QE}(P)}{P} (\text{iff } \phi \leftrightarrow \text{QE}(\phi) \text{ in first-order real arithmetic})$$

Sequent Deduction

$$A \vdash F = f^+ \land x^+ = x + Ft^+ \land t^+ \geq 0 \land x \geq 0 \land \varepsilon \geq t^+ \geq 0 \land Ft^+ + x \geq 0$$

QE

$$A \vdash \forall 0{\leq}\tilde{t}{\leq}T\ (x + f^+\tilde{t} \geq 0 \land \tilde{t} \leq \varepsilon) \land F = f^+ \land x^+ = x + Ft^+ \land t^+ = t^+$$

∃R,WR

$$A \vdash \exists T{\geq}0((\forall 0{\leq}\tilde{t}{\leq}T\ (x + f^+\tilde{t} \geq 0 \land \tilde{t} \leq \varepsilon)) \land F = f^+ \land (x^+ = x + FT \land t^+ = T))$$

⟨′⟩

$$A \vdash \langle f := F; t := 0\rangle\langle\{x' = f, t' = 1\ \&\ x \geq 0 \land t \leq \varepsilon\}\rangle\Upsilon^+$$

Water Tank Example: Synthesized Model Monitor

Input: Model and Safety Property

$$\underbrace{0 \le x \le m \land \varepsilon > 0}_{A} \rightarrow \left[\left(\begin{array}{l} f := *;\ ?\left(-1 \le f \le \frac{m-x}{\varepsilon}\right); \\ t := 0;\ (x' = f,\ t' = 1\ \&\ x \ge 0 \land t \le \varepsilon) \end{array} \right)^* \right] \underbrace{(0 \le x \le m)}_{S}$$

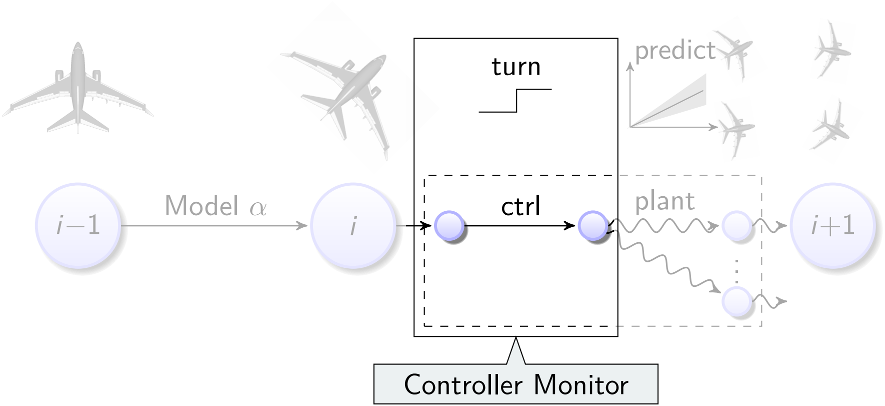
Output: Synthesized Model Monitor

$$-1 \le f^+ \le \frac{m-x}{\varepsilon} \land x^+ = x + f^+ t^+ \land x \ge 0 \land x + f^+ t^+ \ge 0 \land \varepsilon \ge t^+ \ge 0$$

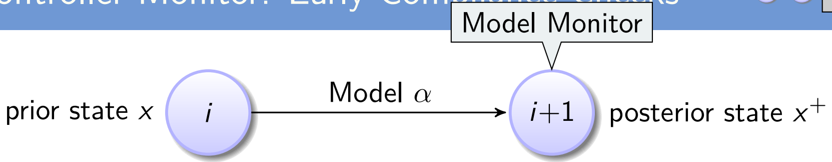
Proof (Generated by ModelPlex tactic).

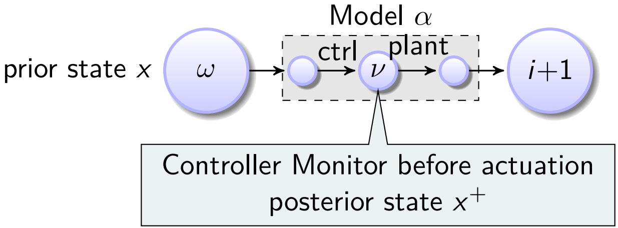
A proof of correctness of the synthesized model monitor. □

For typical models ctrl; plant we can check earlier

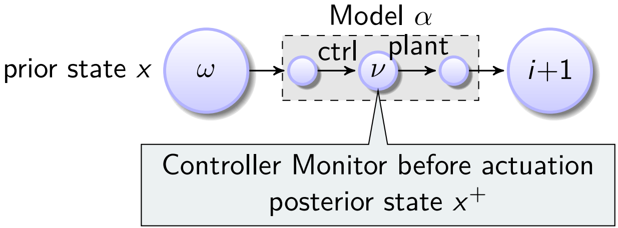


Controller Monitor: Early Compliance Checks





Semantical: $(\omega, \nu) \in [\![\text{ctrl}]\!]$ — reachability relation of ctrl



Offline

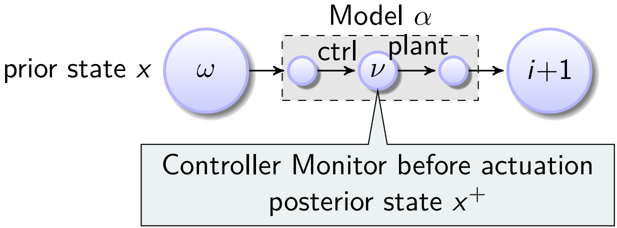
Semantical: $(\omega, \nu) \in [\![\text{ctrl}]\!]$

$\Updownarrow$ Theorem

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle \text{ctrl} \rangle (x = x^+)$

exists a run of ctrl to a state where $x = x^+$

Controller Monitor: Early Compliance Checks



Offline

Semantical: $(\omega, \nu) \in [\![\text{ctrl}]\!]$

$\Updownarrow$ Theorem

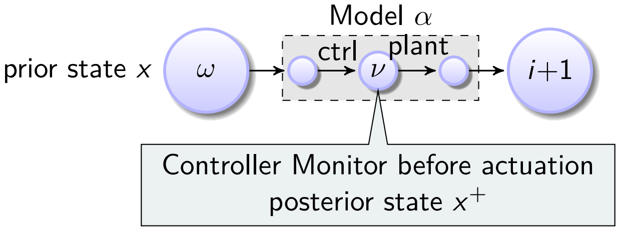
Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle \text{ctrl} \rangle (x = x^+)$

exists a run of ctrl to a state where $x = x^+$

$\Uparrow$ d$\mathcal{L}$ proof

Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)



Offline

Semantical: $(\omega,\nu) \in [\![\text{ctrl}]\!]$

$\Updownarrow$ Theorem

Logical d$\mathcal{L}$: $(\omega,\nu) \models \langle \text{ctrl} \rangle (x = x^+)$

exists a run of ctrl to a state where $x = x^+$

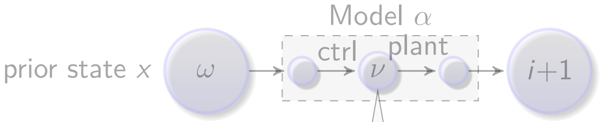
$\Uparrow$ d$\mathcal{L}$ proof

Arithmetical: $(\omega,\nu) \models F(x,x^+)$

check at runtime (efficient)

Theorem (Controller Monitor Correctness)

"Controller safe & in plant bounds as long as monitor satisfied." FMSD'16



Controller Monitor before actuation
posterior state x^+

Offline

Controller Monitor

Immediate detection of unsafe control before actuation
$\rightsquigarrow$ Safe execution of unverified implementations
in perfect environments

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle \text{ctrl} \rangle (x = x^+)$

$\Uparrow$ d$\mathcal{L}$ proof

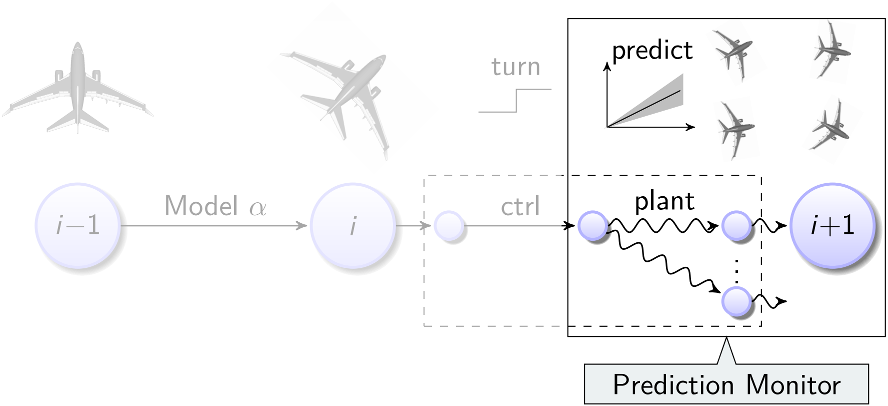
Arithmetical: $(\omega, \nu) \models F(x, x^+)$ — check at runtime (efficient)

Theorem (Controller Monitor Correctness)

"Controller safe & in plant bounds as long as monitor satisfied." FMSD'16

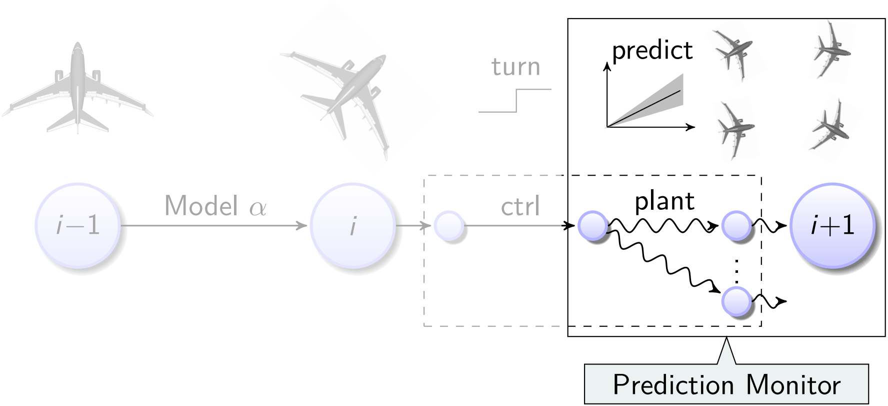
Outline

Safe despite evolution with disturbance?

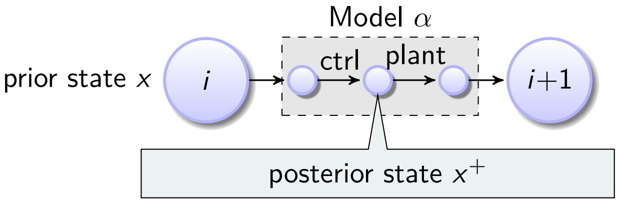


Outline

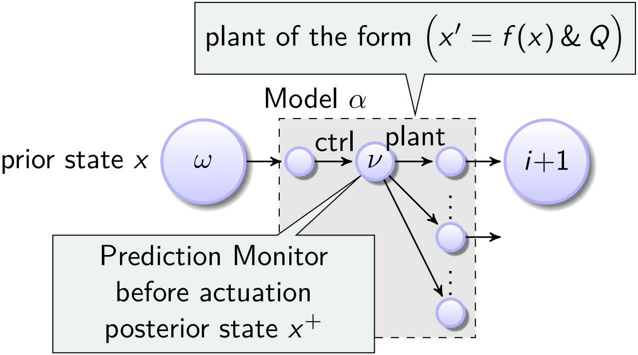
Safe despite evolution with disturbance?

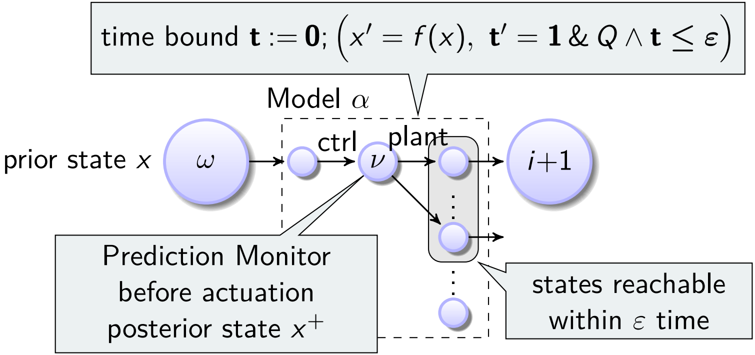


"Prediction is very difficult, especially if it's about the future." [Nils Bohr]

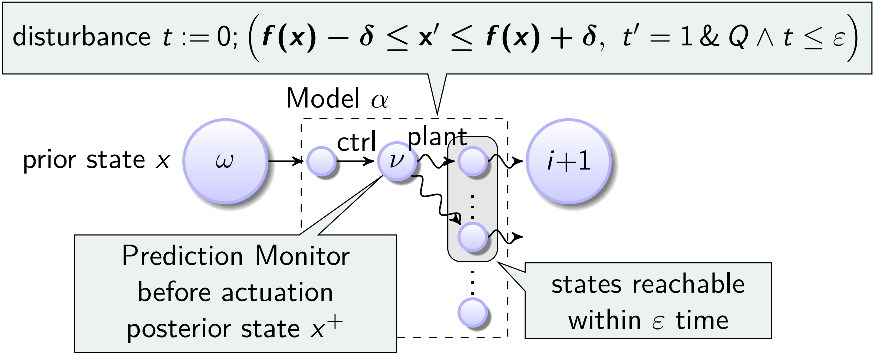

Model α
prior state x
i
ctrl
plant
$i+1$
posterior state x^+

Prediction Monitor: Compliance with Disturbance

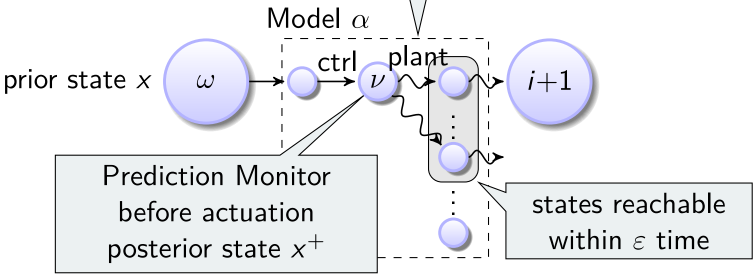




time bound $\mathbf{t} := \mathbf{0}; \left(x' = f(x),\ \mathbf{t}' = \mathbf{1} \,\&\, Q \land \mathbf{t} \leq \varepsilon\right)$
Model α
prior state x
ω
ctrl
ν
plant
$i+1$
Prediction Monitor before actuation posterior state x^+
states reachable within ε time


disturbance $t := 0; \left(\boldsymbol{f(x)} - \boldsymbol{\delta} \le \mathbf{x}' \le \boldsymbol{f(x)} + \boldsymbol{\delta},\ t' = 1 \,\&\, Q \wedge t \le \varepsilon\right)$
Model α
prior state x
ω
ctrl
ν
plant
$i+1$
Prediction Monitor
before actuation
posterior state x^+
states reachable
within ε time

disturbance $t := 0; \left(\boldsymbol{f(x)} - \boldsymbol{\delta} \le \mathbf{x}' \le \boldsymbol{f(x)} + \boldsymbol{\delta},\ t' = 1 \,\&\, Q \wedge t \le \varepsilon \right)$



Offline

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle \text{ctrl} \rangle (x = x^+ \wedge [\text{plant}]\varphi)$

$\Uparrow$ d$\mathcal{L}$ proof

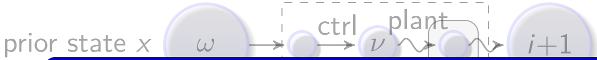
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

Invariant φ implies safety S
(known from safety proof)

Prediction Monitor: Compliance with Disturbance

disturbance $t := 0; \left(\boldsymbol{f(x)} - \boldsymbol{\delta} \leq \mathbf{x}' \leq \boldsymbol{f(x)} + \boldsymbol{\delta},\ t' = 1 \,\&\, Q \land t \leq \varepsilon\right)$

Model α



Prediction Monitor with Disturbance

Proactive detection of unsafe control before actuation despite disturbance

$\rightsquigarrow$ **Safety in realistic environments**

Offline

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle \text{ctrl} \rangle (x = x^+ \land [\text{plant}]\varphi)$

$\Uparrow$ d$\mathcal{L}$ proof

Arithmetical: $(\omega, \nu) \models F(x, x^+)$

Invariant φ implies safety S (known from safety proof)

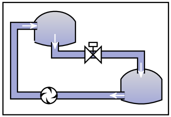
Outline

1. Motivation
2. Learning Objectives
3. ModelPlex Runtime
   - ModelPlex Runtime
   - ModelPlex Compliance
4. ModelPlex
   - Logical State Relations
   - Model Monitors
   - Correct-by-Construction Synthesis
   - Example: Water Tank
   - Controller Monitors
   - Prediction Monitors
5. Evaluation
6. Summary

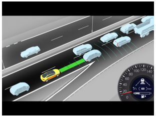
Evaluation

- Evaluated on hybrid system case studies

Water tank



Cruise control



© Volvo

Traffic control

© ASFINAG

Ground robots

© Black-I Robotics

Train control

© Harald Eisenberger

- Model sizes: 5–16 variables
- Monitor sizes: 20–150 operations
- Synthesis duration: 0.3–23 seconds (axiomatic) 6.2–211 (sequent)
- ModelPlex tactic produces correct-by-construction monitor in KeYmaera X
- Theorem: ModelPlex is decidable and monitor synthesis fully automated for controller monitor synthesis and for important classes

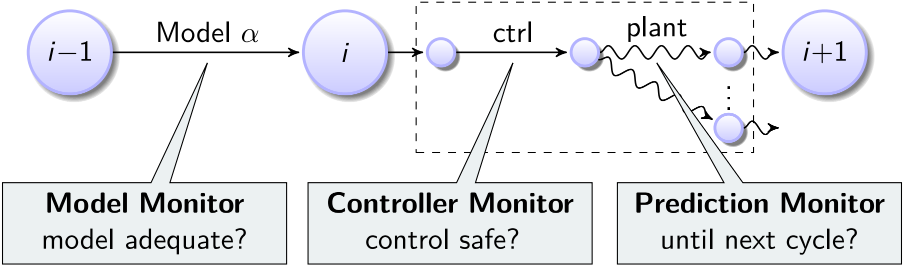
Outline

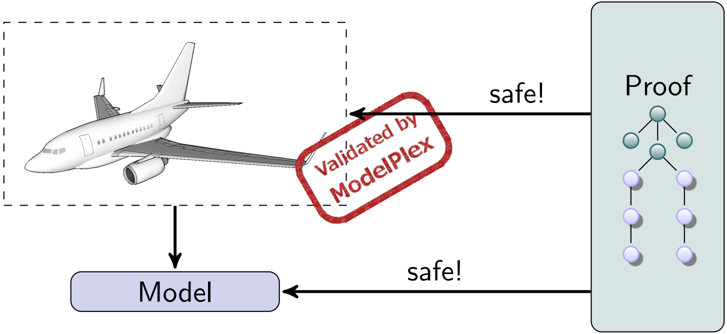
1. Motivation
2. Learning Objectives
3. ModelPlex Runtime
   - ModelPlex Runtime
   - ModelPlex Compliance
4. ModelPlex
   - Logical State Relations
   - Model Monitors
   - Correct-by-Construction Synthesis
   - Example: Water Tank
   - Controller Monitors
   - Prediction Monitors
5. Evaluation
6. Summary

Summary

ModelPlex ensures that proofs apply to real CPS

- Validate model compliance
- Characterize compliance with model in logic
- Prover transforms compliance formula to executable monitor
- Provably correct runtime model validation




safe!
Proof
Validated by ModelPlex
Model
safe!

Stefan Mitsch and André Platzer.
ModelPlex: Verified runtime validation of verified cyber-physical system models.
In Borzoo Bonakdarpour and Scott A. Smolka, editors, *RV*, volume 8734 of *LNCS*, pages 199–214. Springer, 2014.
doi:10.1007/978-3-319-11164-3_17.

Stefan Mitsch and André Platzer.
ModelPlex: Verified runtime validation of verified cyber-physical system models.
Form. Methods Syst. Des., 2016.
Special issue of selected papers from RV'14.
doi:10.1007/s10703-016-0241-z.

André Platzer.
Differential dynamic logic for hybrid systems.
J. Autom. Reas., 41(2):143–189, 2008.
doi:10.1007/s10817-008-9103-8.

André Platzer.

A uniform substitution calculus for differential dynamic logic.
In Amy Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 467–481. Springer, 2015.
doi:10.1007/978-3-319-21401-6_32.