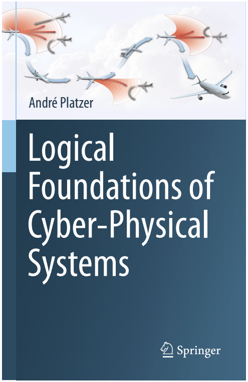
19: Verified Models & Verified Runtime Validation

Logical Foundations of Cyber-Physical Systems



André Platzer

Karlsruhe Institute of Technology
Department of Informatics

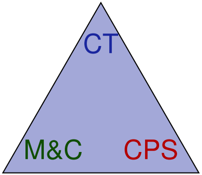
Computer Science Department
Carnegie Mellon University

Outline

Outline

Learning Objectives

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
runtime validation
online monitor
prediction vs. run

Outline

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Wrong?

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \to [(ctrl; plant)^*]S$$

Wrong?

- S Right answer to wrong question.

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Wrong?

S Right answer to wrong question.

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Wrong?

S Right answer to wrong question.

A Proof, so can't forget condition.
Except too picky to turn on.

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Wrong?

S Right answer to wrong question.

A Proof, so can't forget condition.
Except too picky to turn on.

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \to [(ctrl; plant)^*]S$$

Wrong?

- *S* Right answer to wrong question.
- *A* Proof, so can't forget condition.
  Except too picky to turn on.
- *ctrl* Control model vs.
  controller implementation

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Wrong?

- *S* Right answer to wrong question.
- *A* Proof, so can't forget condition. Except too picky to turn on.
- *ctrl* Control model vs. controller implementation Abstraction helps scale!

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Wrong?

- *S* Right answer to wrong question.
- *A* Proof, so can't forget condition. Except too picky to turn on.
- *ctrl* Control model vs. controller implementation Abstraction helps scale!

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Wrong?

- *S* Right answer to wrong question.
- *A* Proof, so can't forget condition.
  Except too picky to turn on.
- *ctrl* Control model vs.
  controller implementation
  Abstraction helps scale!
- *plant* Plant model vs.
  real physics

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

All models are wrong but some are useful. G. Box

- S Right answer to wrong question.
- A Proof, so can't forget condition. Except too picky to turn on.
- *ctrl* Control model vs. controller implementation Abstraction helps scale!
- *plant* Plant model vs. real physics Models are inevitable!

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

All models are wrong but some are useful. G. Box

Models Predictions need models!

- *S* Right answer to wrong question.
- *A* Proof, so can't forget condition. Except too picky to turn on.
- *ctrl* Control model vs. controller implementation Abstraction helps scale!
- *plant* Plant model vs. real physics Models are inevitable!

What Else Could Possibly Go Wrong?

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

All models are wrong but some are useful. G. Box

Models Predictions need models!

- *S* Right answer to wrong question
- *A* Pr…
  Ex…
- *ctrl* Co…
  controller implementation
  Abstraction helps scale!
- *plant* Plant model vs.
  real physics
  Models are inevitable!

Challenge

Verification results about models
only apply if CPS fits to the model

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

All models are wrong but some are useful. G. Box

Models Predictions need models!

- S Right answer to wrong question
- A Pr...
  Ex...
- $ctrl$ Co...
  controller implementation
  Abstraction helps scale!
- $plant$ Plant model vs. real physics
  Models are inevitable!

Challenge

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

Outline

Runtime Monitor for Runtime Validation of Model

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Runtime Monitor for Runtime Validation of Model

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Monitor

Proposition (System Proved Safe)

$$A \to [(ctrl; plant)^*]S$$

Monitor

A Monitor easy if measurable.
Veto turns CPS off.

Runtime Monitor for Runtime Validation of Model

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Monitor

A Monitor easy if measurable.
Veto turns CPS off.

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Monitor

A Monitor easy if measurable.
Veto turns CPS off.

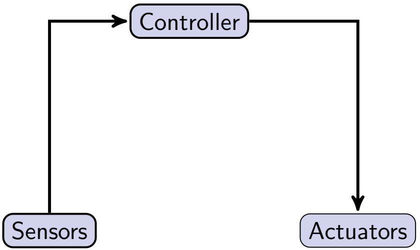
S Too late to monitor.
CPS already unsafe!

Proposition (System Proved Safe)

$$A \to [(ctrl; plant)^*]S$$

Monitor

- *A* Monitor easy if measurable.
  Veto turns CPS off.
- *S* Too late to monitor.
  CPS already unsafe!

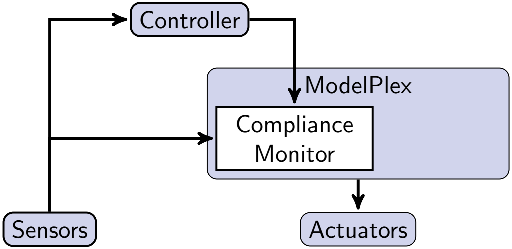


Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Monitor

- *A* Monitor easy if measurable. Veto turns CPS off.
- *S* Too late to monitor. CPS already unsafe!
- *ctrl* Monitor each control decision. Veto overrides decision.



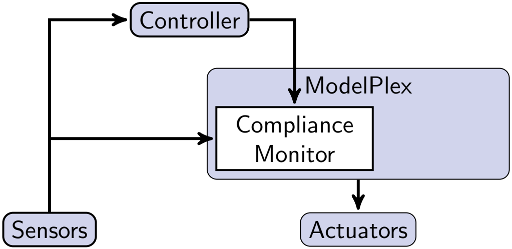
Runtime Monitor for Runtime Validation of Model

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Monitor

- *A* Monitor easy if measurable. Veto turns CPS off.
- *S* Too late to monitor. CPS already unsafe!
- *ctrl* Monitor each control decision. Veto overrides decision.



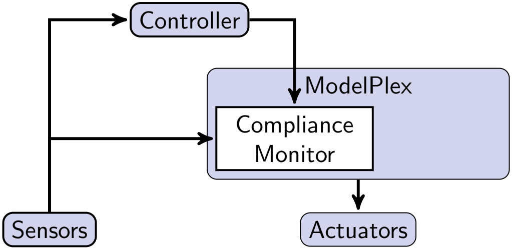
Runtime Monitor for Runtime Validation of Model

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Monitor

- *A* Monitor easy if measurable.
  Veto turns CPS off.
- *S* Too late to monitor.
  CPS already unsafe!
- *ctrl* Monitor each control decision.
  Veto overrides decision.
- *plant* No source code for physics.
  Observe and compare.



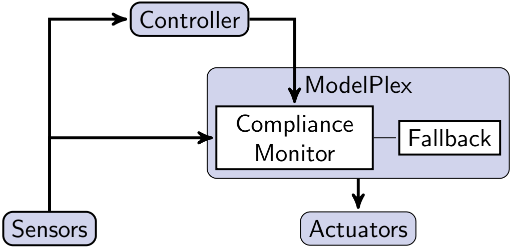
Runtime Monitor for Runtime Validation of Model

Proposition (System Proved Safe)

$$A \to [(ctrl; plant)^*]S$$

Monitor

- *A* Monitor easy if measurable. Veto turns CPS off.
- *S* Too late to monitor. CPS already unsafe!
- *ctrl* Monitor each control decision. Veto overrides decision.
- *plant* No source code for physics. Observe and compare. Veto triggers best fallback.



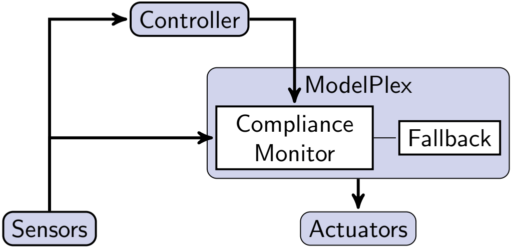
Runtime Monitor for Runtime Validation of Model

Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Monitors must be correct

- *A* Monitor easy if measurable. Veto turns CPS off.
- *S* Too late to monitor. CPS already unsafe!
- *ctrl* Monitor each control decision. Veto overrides decision.
- *plant* No source code for physics. Observe and compare. Veto triggers best fallback.



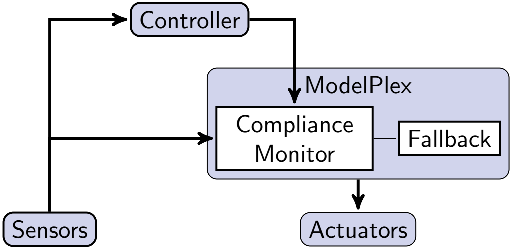
Proposition (System Proved Safe)

$$A \rightarrow [(ctrl; plant)^*]S$$

Monitors must be correct

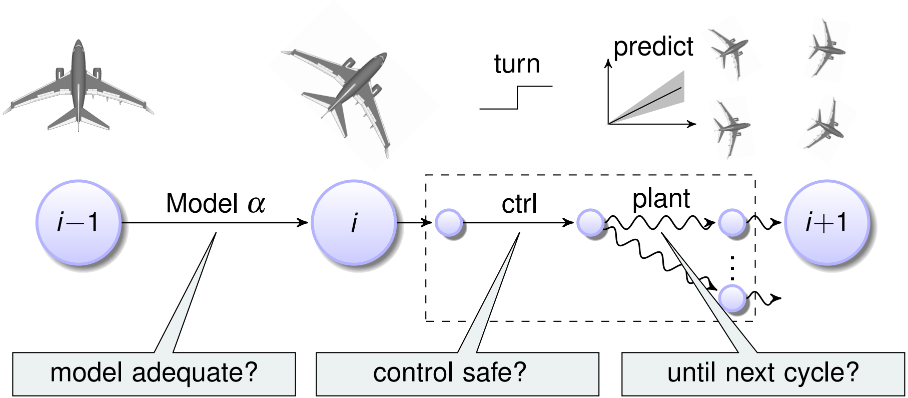
Monitor Verified runtime validation!

- *A* Monitor easy if measurable. Veto turns CPS off.
- *S* Too late to monitor. CPS already unsafe!
- *ctrl* Monitor each control decision. Veto overrides decision.
- *plant* No source code for physics. Observe and compare. Veto triggers best fallback.



ModelPlex: Verified Runtime Validation of Models

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



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

Insights

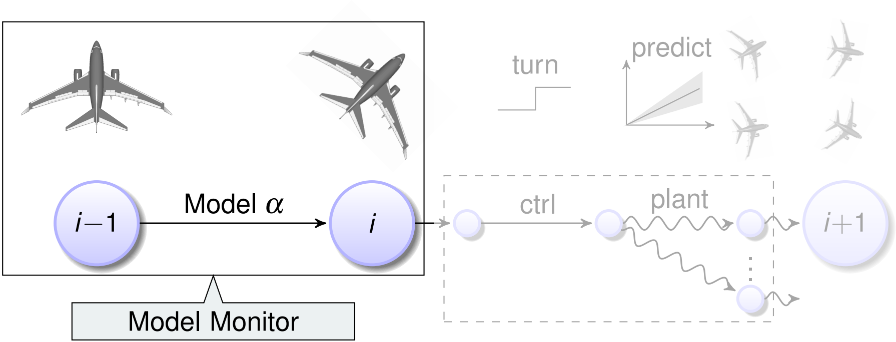
- Verification results about models transfer to CPS when validating model compliance
- Compliance with model is characterizable in logic
- Compliance formula transformed by proof to monitor
- Correct-by-construction verified runtime model validation

model adequate?

control safe?

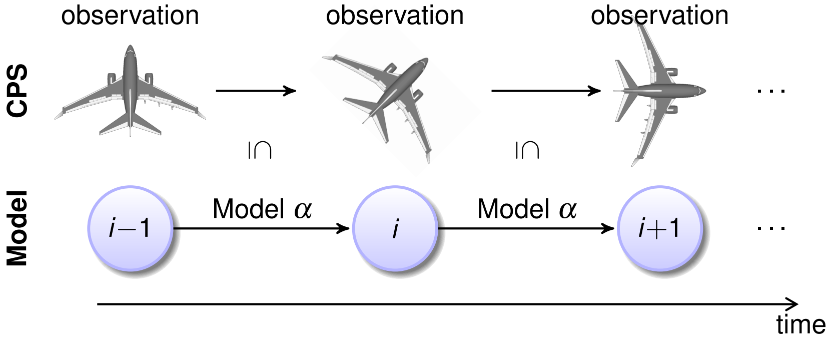
until next cycle?

Outline


turn
predict
Model α
$i-1$
i
ctrl
plant
$i+1$
Model Monitor

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

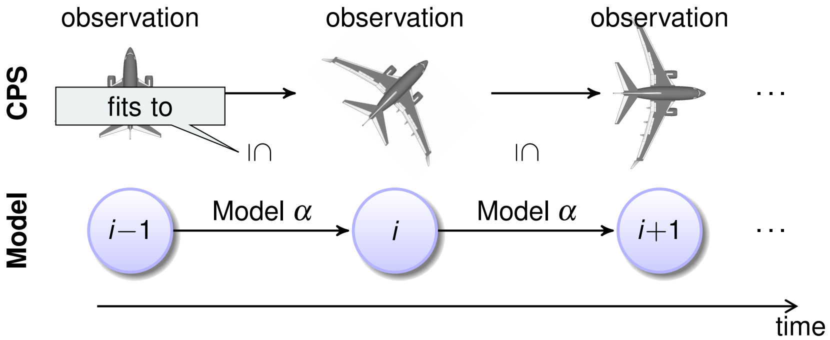
- CPS observed through sensors
- Model describes all possible behavior of CPS between states



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

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

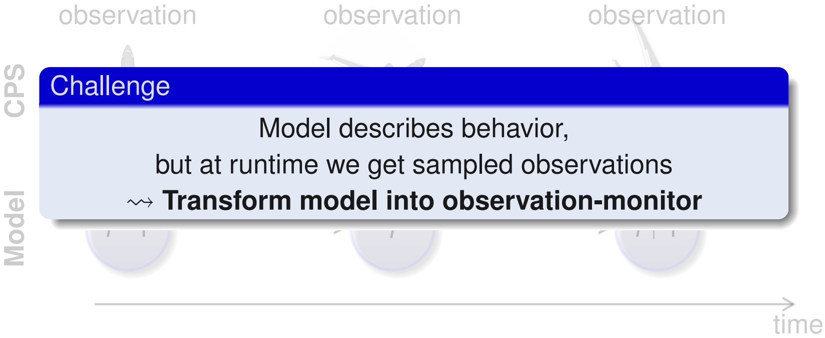
- CPS observed through sensors
- Model describes all possible behavior of CPS between states



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

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

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



Challenge

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

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

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \to$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Proposition (Quantum can bounce around safely)

$$0 \leq x \land x = H \land v = 0 \land g > 0 \land 1 \geq c \geq 0 \rightarrow$$
$$[(\{x' = v, v' = -g \,\&\, x \geq 0\}; (?x = 0; v := -cv \cup ?x \neq 0))^*](0 \leq x \land x \leq H)$$

Example (Controller Monitor)

control changes (x, v) to (x^+, v^+)

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \wedge x = H \wedge v = 0 \wedge g > 0 \wedge 1 \ge c \ge 0 \to$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \wedge x \le H)$$

Example (Controller Monitor)

$$(x = 0 \wedge v^+ = -cv \;\vee\; x > 0 \wedge v^+ = v) \wedge x^+ = x$$

control changes (x, v) to (x^+, v^+)

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \to$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$$(x = 0 \land v^+ = -cv \;\lor\; x > 0 \land v^+ = v) \land x^+ = x$$

test+domain

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \rightarrow$$
$$[\big(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0)\big)^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$$\big(x = 0 \land v^+ = -cv \;\lor\; x > 0 \land v^+ = v\big) \land x^+ = x$$

Example (Plant Monitor)

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \rightarrow$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$$(x = 0 \land v^+ = -cv \ \lor \ x > 0 \land v^+ = v) \land x^+ = x$$

Example (Plant Monitor)

$$(v^+ = v - gt \land x^+ = x + vt - \tfrac{g}{2}t^2)$$

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \rightarrow$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$$(x = 0 \land v^+ = -cv \ \lor \ x > 0 \land v^+ = v) \land x^+ = x$$

Example (Plant Monitor)

$$2g(x^+ - x) = v^2 - (v^+)^2$$

from invariant

$2gx = 2gH - v^2$

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \to$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$$\big(x = 0 \land v^+ = -cv \;\lor\; x > 0 \land v^+ = v\big) \land x^+ = x$$

Example (Plant Monitor)

$$2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v$$

directionality: always falling

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \rightarrow$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$$(x = 0 \land v^+ = -cv \;\lor\; x > 0 \land v^+ = v) \land x^+ = x$$

Example (Plant Monitor)

$$2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v \land x \ge 0 \land x^+ \ge 0$$

domain

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \rightarrow$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$$(x = 0 \land v^+ = -cv \ \lor \ x > 0 \land v^+ = v) \land x^+ = x$$

Example (Plant Monitor)

$$2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v \land x \ge 0 \land x^+ \ge 0$$

Example (Model Monitor)

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \rightarrow$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$$(x = 0 \land v^+ = -cv \;\lor\; x > 0 \land v^+ = v) \land x^+ = x$$

Example (Plant Monitor)

$$2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v \land x \ge 0 \land x^+ \ge 0$$

Example (Model Monitor)

$$x^+ > 0 \land 2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v \land x \ge 0$$
$$\lor\, x^+ = 0 \land c^2 2g(x^+ - x) = c^2 v^2 - (v^+)^2 \land v^+ \ge -cv \land x \ge 0$$

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \to$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$$(x = 0 \land v^+ = -cv \;\lor\; x > 0 \land v^+ = v) \land x^+ = x$$

Example (Plant Monitor)

$$2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v \land x \ge 0 \land x^+ \ge 0$$

substitute in

Example (Model Monitor)

$$x^+ > 0 \land 2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v \land x \ge 0$$
$$\lor\, x^+ = 0 \land c^2 2g(x^+ - x) = c^2 v^2 - (v^+)^2 \land v^+ \ge -cv \land x \ge 0$$

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \rightarrow$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$$(x = 0 \land v^+ = -cv \;\lor\; x > 0 \land v^+ = v) \land x^+ = x$$

Example (Plant Monitor)

$$2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v \land x \ge 0 \land x^+ \ge 0$$

substitute in

Example (Model Monitor)

$$x^+ > 0 \land 2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v \land x \ge 0$$
$$\lor\, x^+ = 0 \land c^2 2g(x^+ - x) = c^2 v^2 - (v^+)^2 \land v^+ \ge -cv \land x \ge 0$$

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \leq x \land x = H \land v = 0 \land g > 0 \land 1 \geq c \geq 0 \rightarrow$$
$$[(\{x' = v, v' = -g \,\&\, x \geq 0\}; (?x = 0; v := -cv \cup ?x \neq 0))^*](0 \leq x \land x \leq H)$$

Example (Controller Monitor)

$$(x = 0 \land v^+ = -cv \ \lor\ x > 0 \land v^+ = v) \land x^+ = x$$

Example (Plant Monitor)

$$2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \leq v \land x \geq 0 \land x^+ \geq 0$$

substitute in

Example (Model Monitor)

$$x^+ > 0 \land 2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \leq v \land x \geq 0$$
$$\lor\, x^+ = 0 \land c^2 2g(x^+ - x) = c^2 v^2 - (v^+)^2 \land v^+ \geq -cv \land x \geq 0$$

Quantum's Bouncing Ball Monitors

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \to$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$$\big(x = 0 \land v^+ = -cv \ \lor\ x > 0 \land v^+ = v\big) \land x^+ = x$$

Example (Plant Monitor)

$$2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v \land x \ge 0 \land x^+ \ge 0$$

Example (Model Monitor)

$$x^+ > 0 \land 2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v \land x \ge 0$$
$$\lor\, x^+ = 0 \land c^2 2g(x^+ - x) = c^2 v^2 - (v^+)^2 \land v^+ \ge -cv \land x \ge 0$$

Proposition (Quantum can bounce around safely)

$$0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \rightarrow$$
$$[(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0))^*](0 \le x \land x \le H)$$

Example (Controller Monitor)

$(x = 0$ …

Examp…

$2g(x^+ -$ …

Takeaway

Monitors are subtle, in desperate need of correctness proof.
What proof implies a safe system if the monitors pass?

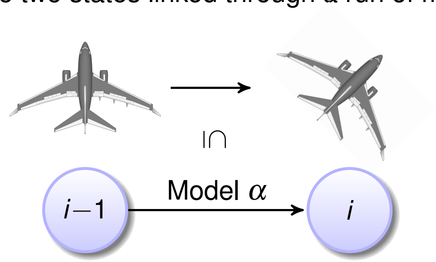
Example (Model Monitor)

$$x^+ > 0 \land 2g(x^+ - x) = v^2 - (v^+)^2 \land v^+ \le v \land x \ge 0$$
$$\lor\, x^+ = 0 \land c^2 2g(x^+ - x) = c^2 v^2 - (v^+)^2 \land v^+ \ge -cv \land x \ge 0$$

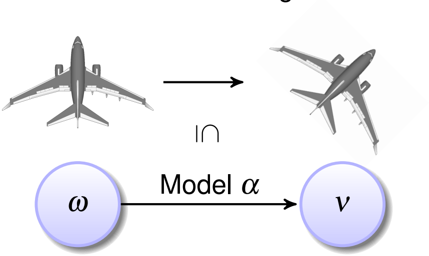
Outline

1. Learning Objectives
2. Fundamental Challenges with Inevitable Models
3. Runtime Monitors
4. Model Compliance
5. Provably Correct Monitor Synthesis
   - Logical State Relations
   - Model Monitors
   - Correct-by-Construction Synthesis
   - Controller Monitors
   - Prediction Monitors
6. Summary

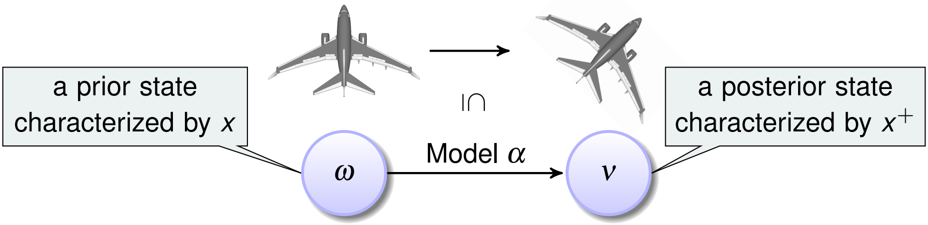
When are two states linked through a run of model α?



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



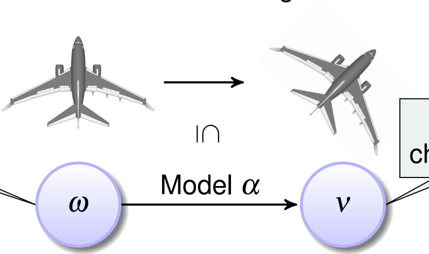
When are two states linked through a run of model α?



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

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

a prior state characterized by x



a posterior state characterized by x^+

Offline

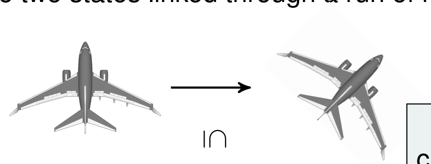
Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

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

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

$\cap\!\!|$

a posterior state characterized by x^+

ω —Model α→ ν

Offline ↓

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

$\Updownarrow$ Lemma

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

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

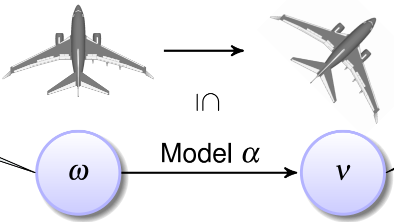
$\Updownarrow$ dL proof

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

check at runtime (efficient)

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

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

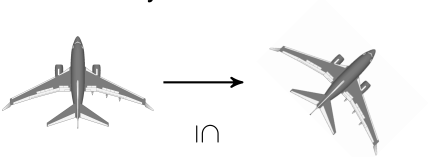
$\Uparrow$ dL proof

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



$\cap$ with a vertical bar, i.e. $\supseteq$-style inclusion

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

Offline

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

$\Updownarrow$ Lemma

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

$\Uparrow$ dL proof

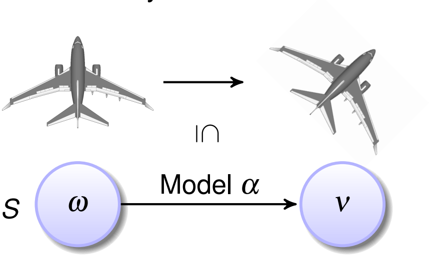
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logical Reductions for Model Safety Transfer

Logic reduces CPS safety to runtime monitor with offline proof

dL proof $A \rightarrow [\alpha]S$



Offline

Init $\omega \in [\![A]\!]$

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

$\Updownarrow$ Lemma

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

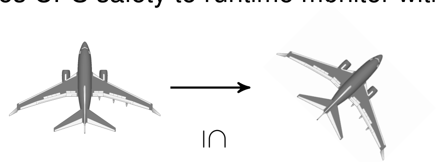
$\Uparrow$ dL proof

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



dL proof $A \rightarrow [\alpha]S$

ω —Model α→ ν

Offline

Init $\omega \in [\![A]\!]$

Safe $\nu \in [\![S]\!]$

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

$\Updownarrow$ Lemma

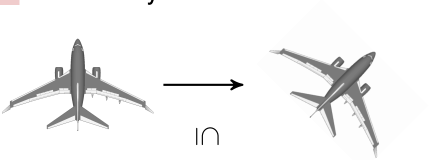
Logical dL: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ dL proof

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

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



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

Offline

Init $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

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

$\Updownarrow$ Lemma

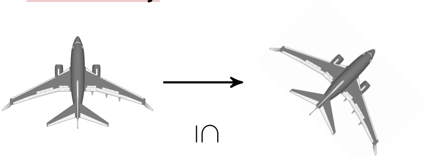
Logical dL: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ 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 $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

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

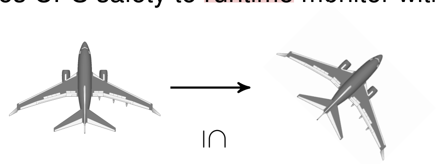
$\Updownarrow$ Lemma

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 $\omega \in [\![A]\!]$

Safe $\nu \in [\![S]\!]$

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

$\Updownarrow$ Lemma

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

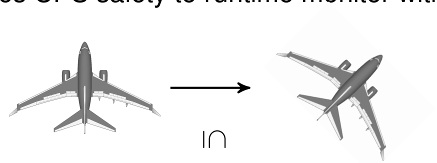
$\Uparrow$ dL proof

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



dL proof $A \rightarrow [\alpha]S$

ω —Model α→ ν

Offline

Init $\omega \in [\![A]\!]$

Safe $\nu \in [\![S]\!]$

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

$\Updownarrow$ Lemma

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

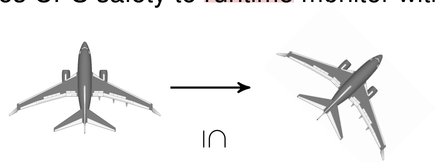
$\Uparrow$ dL proof

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



$\cap\!|$ (the set below is contained in the transition shown above)

dL proof $A \rightarrow [\alpha]S$

$\omega \xrightarrow{\text{Model } \alpha} \nu$

Offline

Init $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

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

$\Updownarrow$ Lemma

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

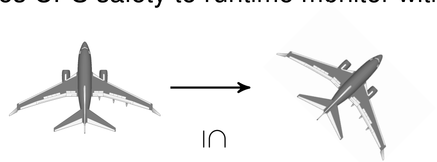
$\Uparrow$ dL proof

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



$\subseteq$

Not initial state.
Model repeats. . .

dL proof $A \to [\alpha]S$

ω —Model α→ ν

Offline

Init $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

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

$\Updownarrow$ Lemma

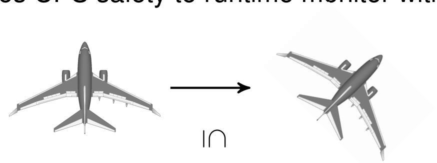
Logical dL: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ 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

Init $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

Offline

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

$\Updownarrow$ Lemma

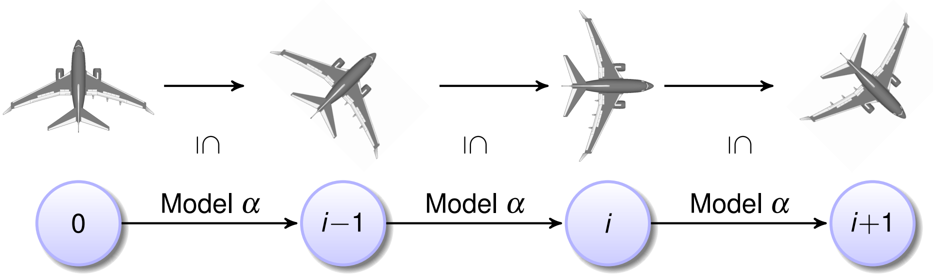
Logical dL: $(\omega, \nu) \models \langle \alpha^* \rangle (x = x^+)$

$\Uparrow$ dL proof

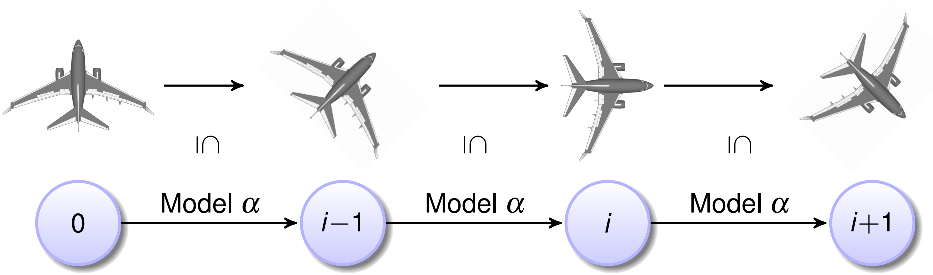
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

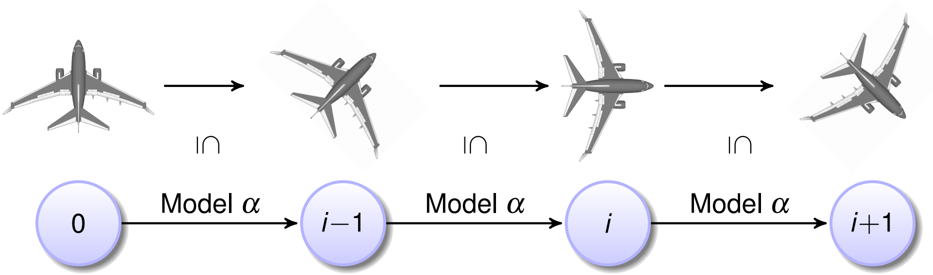
ModelPlex Model Monitor Correctness



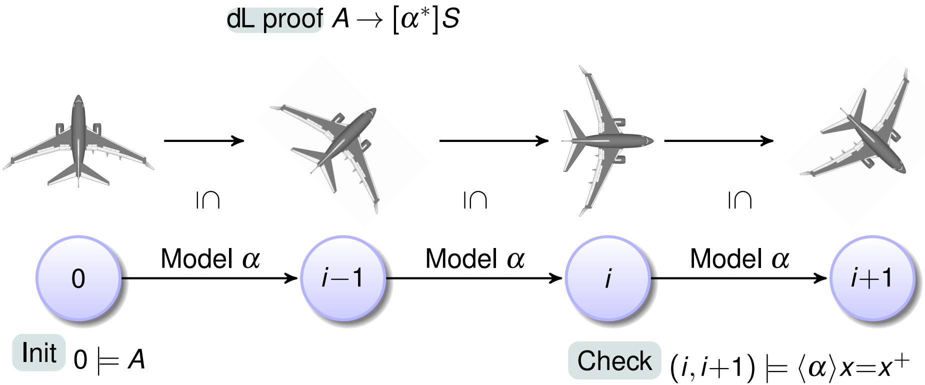
dL proof $A \rightarrow [\alpha^*]S$



dL proof $A \rightarrow [\alpha^*]S$

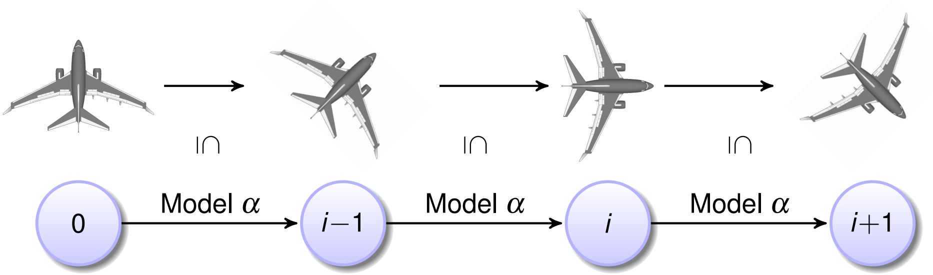


Init $0 \models A$


dL proof $A \rightarrow [\alpha^*]S$
$\subseteq$
$\subseteq$
$\subseteq$
0
Model α
$i-1$
Model α
i
Model α
$i+1$
Init $0 \models A$
Check $(i, i+1) \models \langle\alpha\rangle x = x^+$

ModelPlex Model Monitor Correctness

dL proof $A \rightarrow [\alpha^*]S$

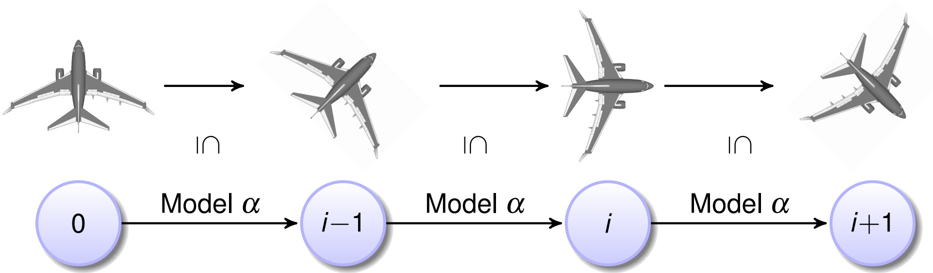


Init $0 \models A$

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

Safe $i+1 \models S$

dL proof $A \rightarrow [\alpha^*]S$



Init $0 \models A$

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

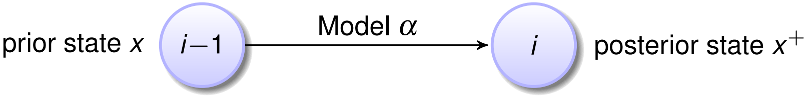
Safe $i+1 \models S$

Theorem (Model Monitor Correctness) (FMSD'16)

System safe as long as monitor satisfied.

Correct-by-Construction Synthesis

- dL proof calculus executes models symbolically

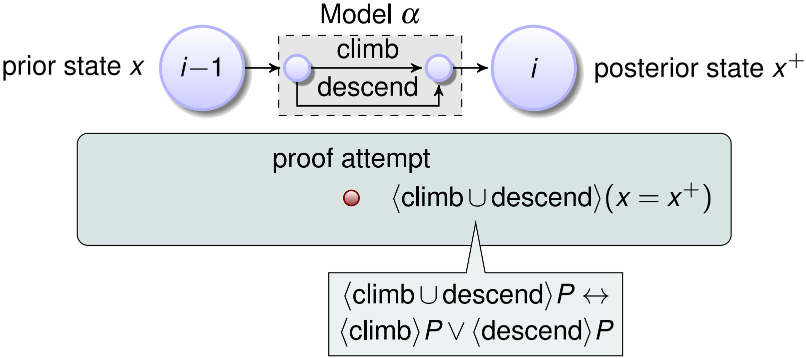


proof attempt

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

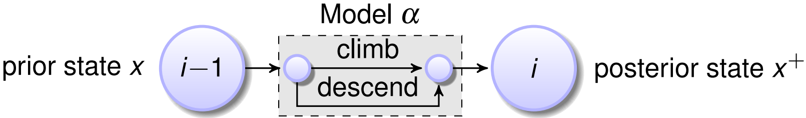
Correct-by-Construction Synthesis

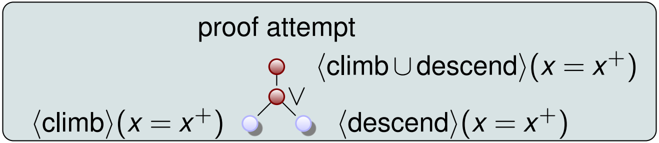
- dL proof calculus executes models symbolically



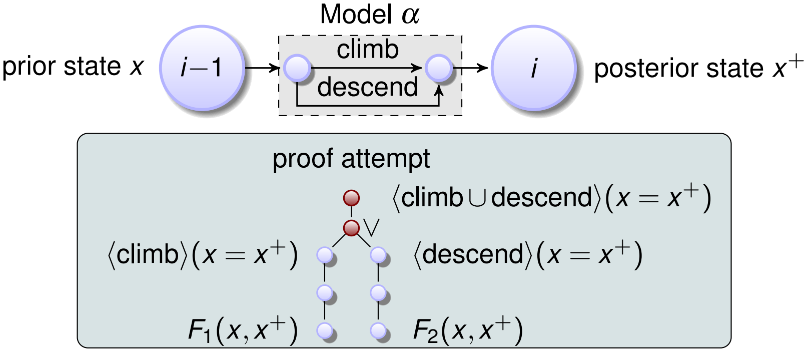
Correct-by-Construction Synthesis

- dL proof calculus executes models symbolically

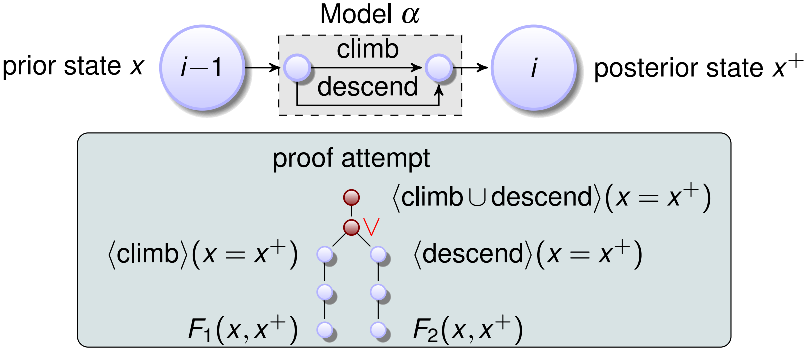




- dL proof calculus executes models symbolically

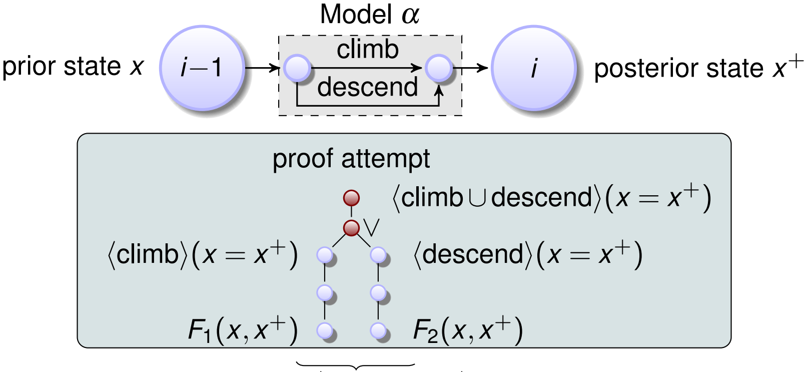


- dL proof calculus executes models symbolically



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

- dL proof calculus executes models symbolically

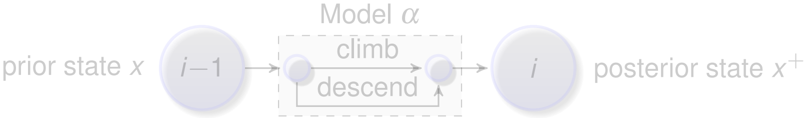


Monitor: $F_1(x,x^+) \lor F_2(x,x^+)$

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

Correct-by-Construction Synthesis

- dL proof calculus executes models symbolically



proof attempt

Model Monitor

Immediate detection of model violation

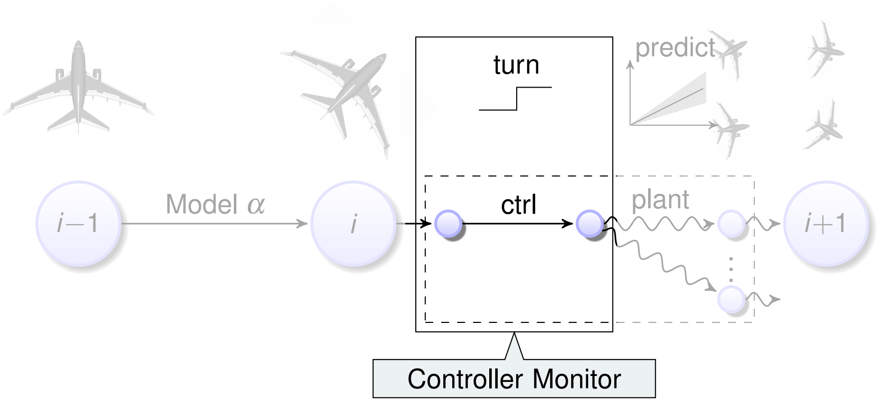
$\rightsquigarrow$ Mitigates safety issues with safe fallback action

$F_1(x, x^+)$ $F_2(x, x^+)$

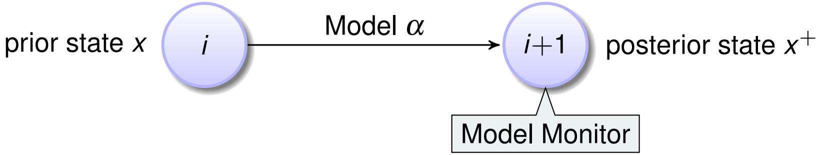
Monitor: $F_1(x, x^+) \lor F_2(x, x^+)$

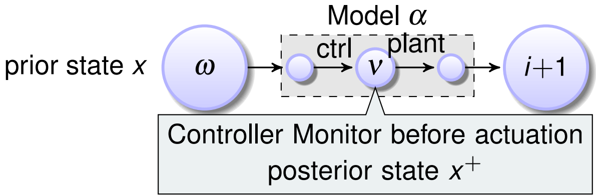
- The subgoals that cannot be proved express all the conditions on the relations of variables imposed by the model $\rightsquigarrow$ prove at runtime

Typical $(\text{ctrl}; \text{plant})^*$ models can check earlier

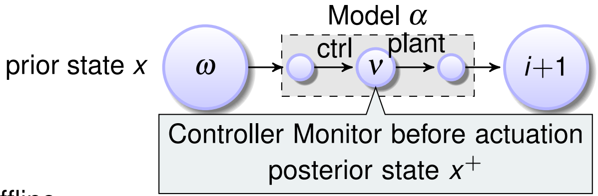


Controller Monitor: Veto Early If Noncompliant





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



Offline

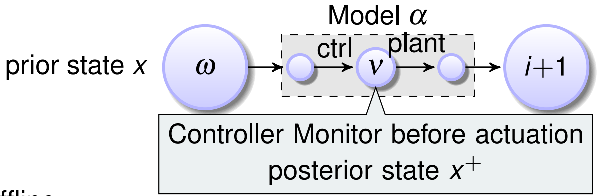
Semantical: $(\omega, \nu) \in [\![\text{ctrl}]\!]$

$\Updownarrow$ Theorem

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

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

Controller Monitor: Veto Early If Noncompliant



Offline

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

$\Updownarrow$ Theorem

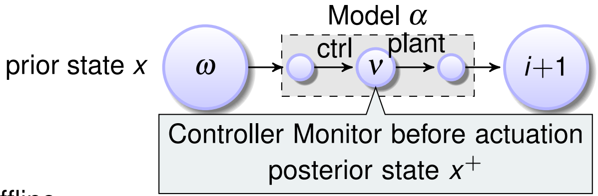
Logical dL: $(\omega, \nu) \models \langle \text{ctrl} \rangle (x = x^+)$

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

$\Uparrow$ dL proof

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

check at runtime (efficient)

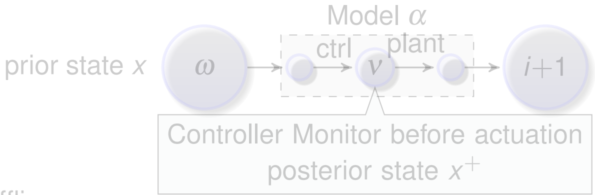


Offline

Semantical:	$(\omega, \nu) \in [\![\text{ctrl}]\!]$	
	$\Updownarrow$ Theorem	
Logical dL:	$(\omega, \nu) \models \langle \text{ctrl} \rangle (x = x^+)$	exists a run of ctrl to a state where $x = x^+$
	$\Uparrow$ dL proof	
Arithmetical:	$(\omega, \nu) \models F(x, x^+)$	check at runtime (efficient)

Theorem (Controller Monitor Correctness) (FMSD'16)

Controller safe and in plant bounds as long as monitor satisfied.



Offline

Controller Monitor

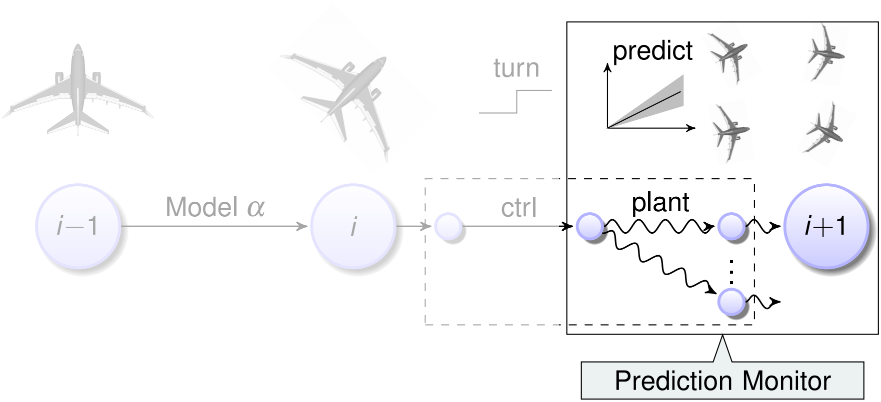
Immediate detection of unsafe control before actuation
$\rightsquigarrow$ Safe execution of unverified implementations
in perfect environments

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

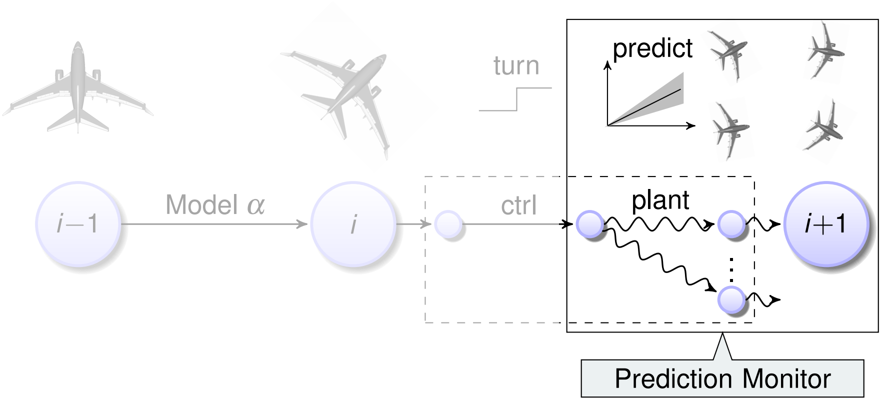
Theorem (Controller Monitor Correctness) (FMSD'16)

Controller safe and in plant bounds as long as monitor satisfied.

Safe despite evolution with disturbance?

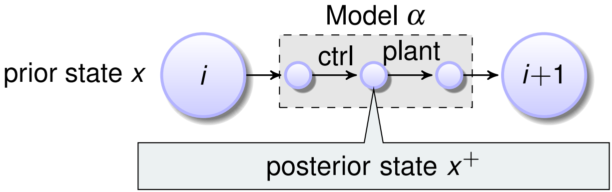


Safe despite evolution with disturbance?

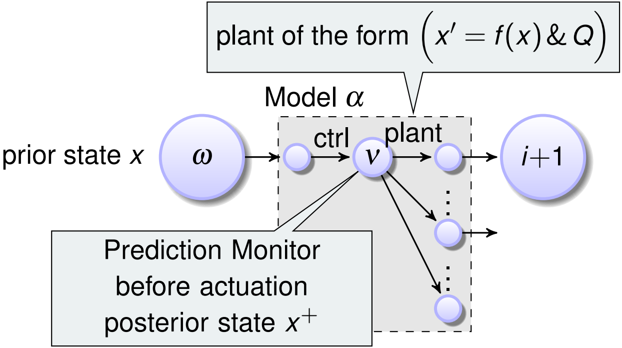


"Prediction is very difficult, especially if it's about the future." [Nils Bohr]

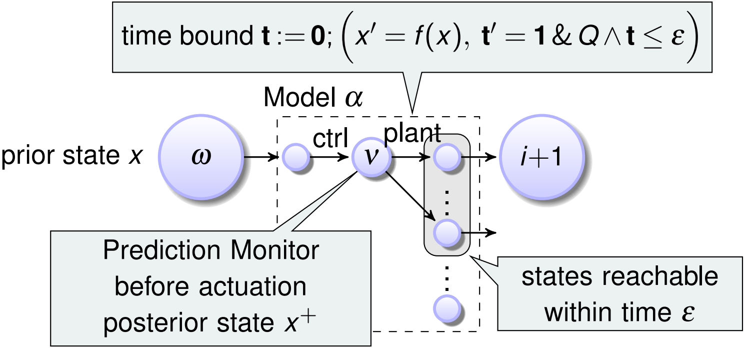
Prediction Monitor: Compliance with Disturbance



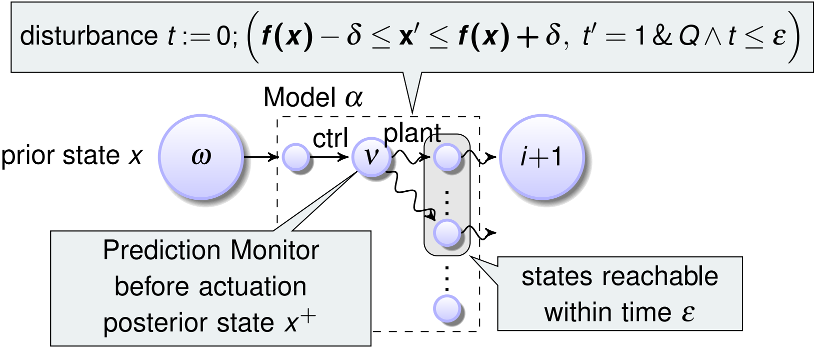
Prediction Monitor: Compliance with Disturbance



Prediction Monitor: Compliance with Disturbance

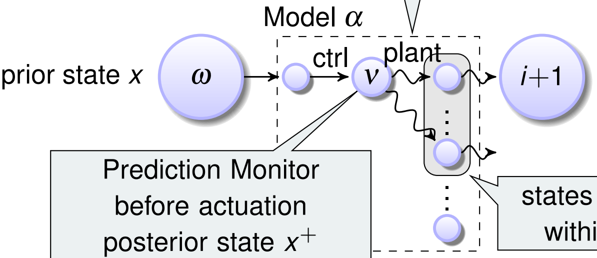


Prediction Monitor: Compliance with Disturbance



Prediction Monitor: Compliance with Disturbance

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



Prediction Monitor
before actuation
posterior state x^+

states reachable
within time ε

Offline

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

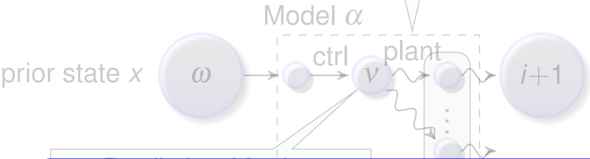
$\Uparrow$ dL proof

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

Invariant J implies safety S
(known from safety proof)

Prediction Monitor: Compliance with Disturbance

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



Prediction Monitor with Disturbance

Detect unsafe control before actuation despite disturbance

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

Offline

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

$\Uparrow$ dL proof

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

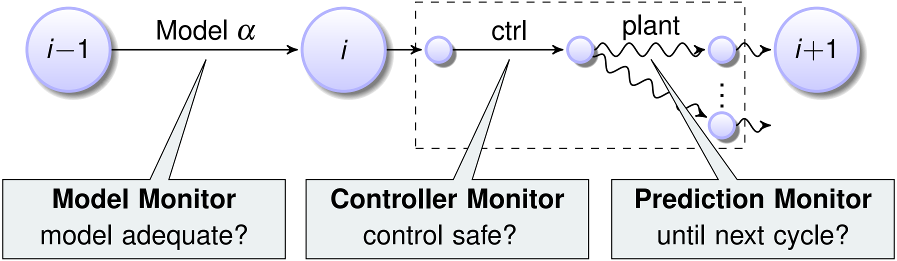
Invariant J implies safety S
(known from safety proof)

Outline

Summary

ModelPlex ensures that proofs transfer to real CPS

- Validate model compliance
- Characterize compliance with model in logic
- Prover transforms compliance formula to executable monitor
- Provably correct runtime model validation by offline + online proof



André Platzer.
Logical Foundations of Cyber-Physical Systems.
Springer, Cham, 2018.
doi:10.1007/978-3-319-63588-0.

Stefan Mitsch and André Platzer.
ModelPlex: Verified runtime validation of verified cyber-physical system models.
Form. Methods Syst. Des., 49(1-2):33–74, 2016.
Special issue of selected papers from RV'14.
doi:10.1007/s10703-016-0241-z.

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.

André Platzer.
A complete uniform substitution calculus for differential dynamic logic.

J. Autom. Reas., 59(2):219–265, 2017.
doi:10.1007/s10817-016-9385-1.