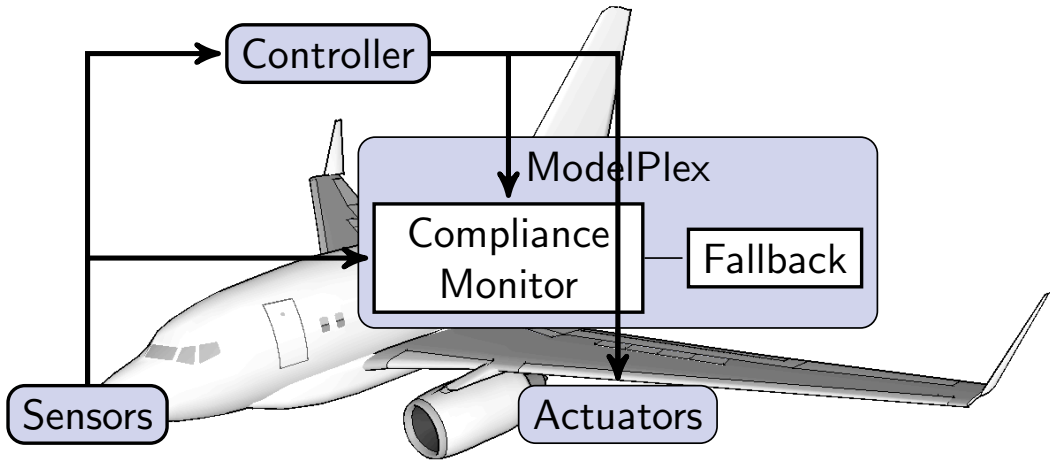
14: Verified Models & Verified Runtime Validation

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

Outline

Correctness Questions in Complex System Design

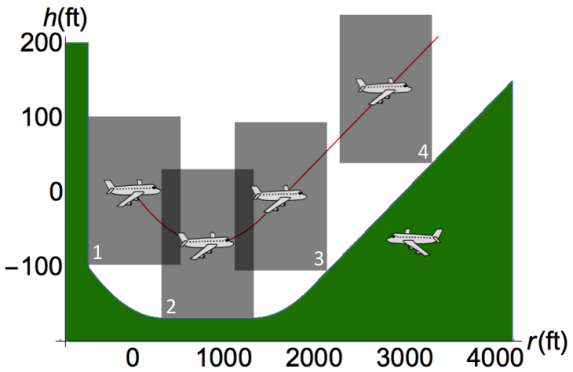
Safety The system must be safe under all circumstances

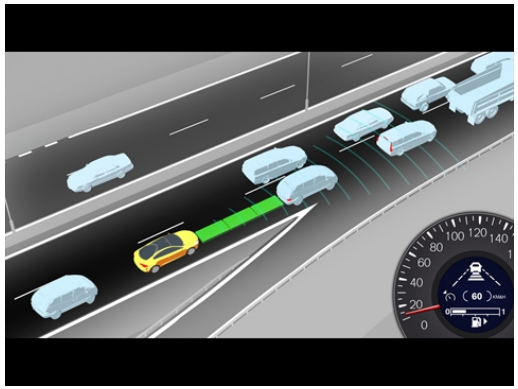
Liveness The system must reach a given goal

How do we make cyber-physical systems safe?

Extensive testing?
Code reviews?

When are we done? How many test cases are enough? Did we cover all relevant tests?





Benefits of Logical Foundations for CPS V & V

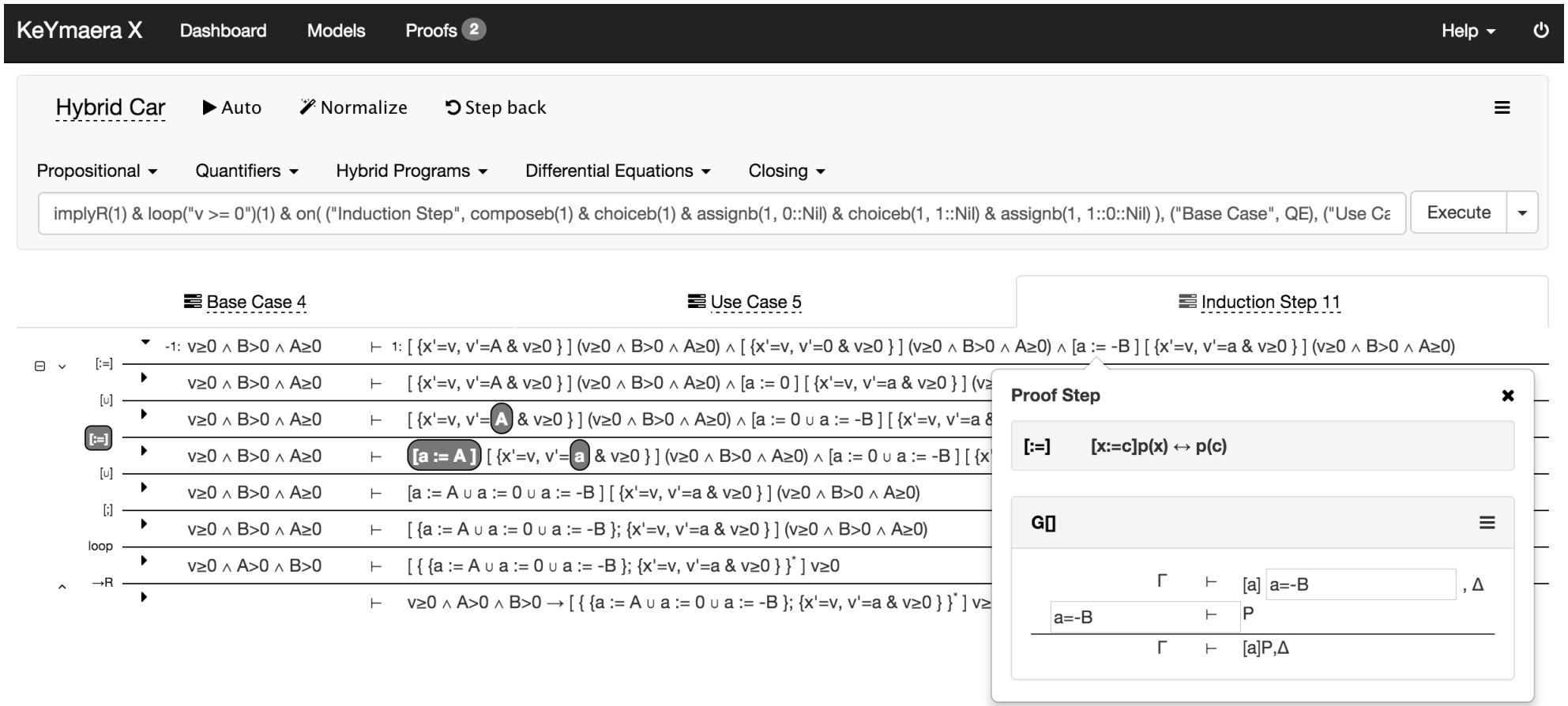
Proofs		LICS'12,JAR'16
Safety	Formalize system properties: What is "Safe"? "Reach goal"?	
Models	Formalize system models, clarify behavior	
Assumptions	Make assumptions explicit rather than silently	
Predictions	Safety analysis predicts behavior for infinitely many cases	
Constraints	Reveal invariants, switching conditions, operating conditions	
Design	Invariants/proofs guide safe controller design	

Byproducts		
Analysis	Determine design trade-offs & feasibility early	arXiv
Synthesis	Turn models into code & safety monitors	ModelPlex
Certificate	Proofs as evidence for certification	CPP'16

Tools		
KeYmaera X	aXiomatic Tactical Theorem Prover for CPS	CADE'15

An aXiomatic Tactical Theorem Prover for CPS

http://keymaeraX.org/



CADE'15

An aXiomatic Tactical Theorem Prover for CPS

`http://keymaeraX.org/`

Small Core	Increases trust, modularity, enables experimentation	(1677)
Tactics	Bridging between small core and powerful reasoning steps	(Hilbert) (Sequent++)
Separation	Tactics can make courageous inferences Core establishes soundness	
Search&Do	Search-based tactics that follow proof search strategies Constructive tactics that directly build a proof	
Interaction	Interactive proofs mixed with tactical proofs and proof search	
Extensible	Flexible for new algorithms, new tactics, new logics, new proof rules, new axioms, ...	
Customize	Modular user interface, API	

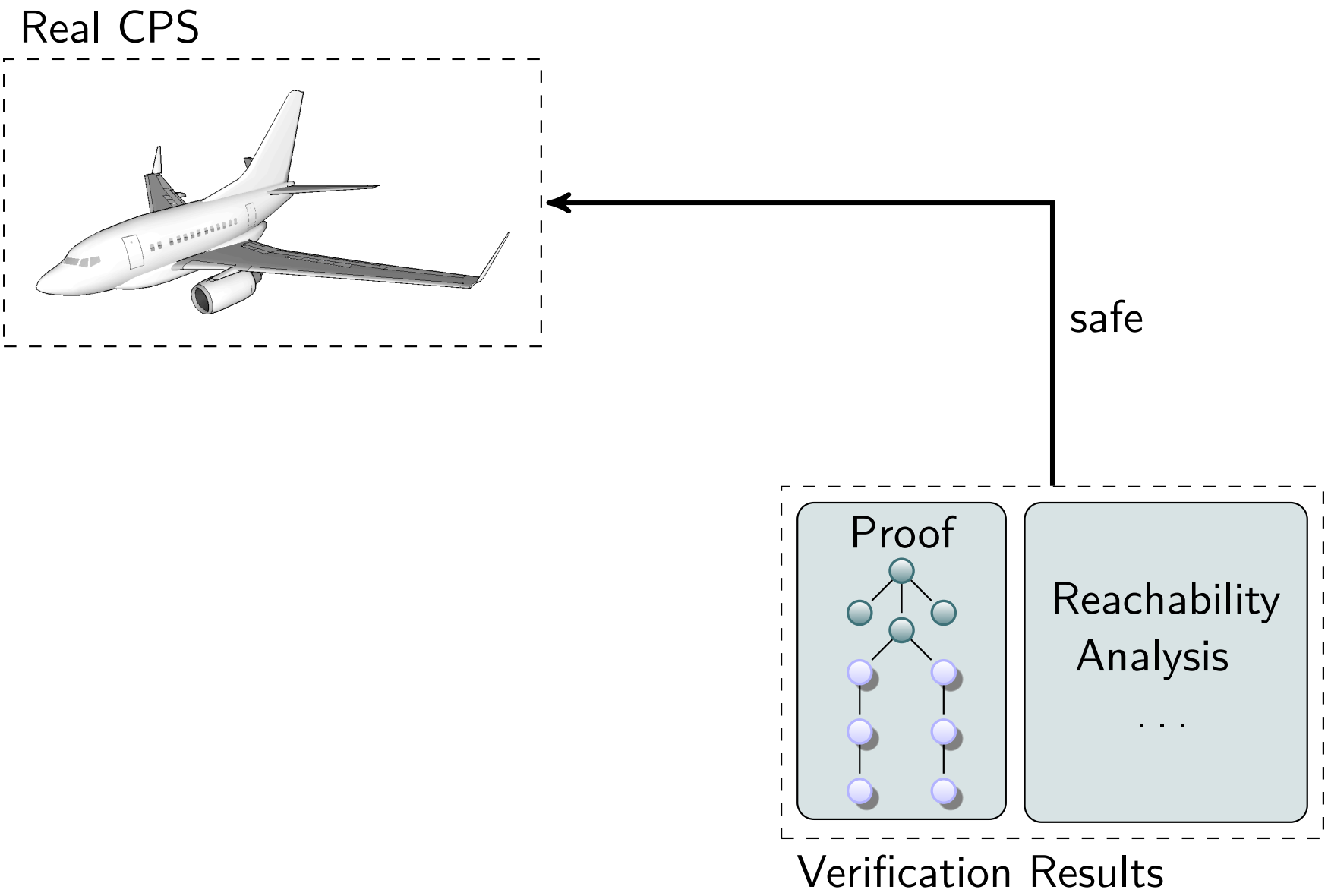
CADE'15

KeYmaera X Microkernel for Soundness

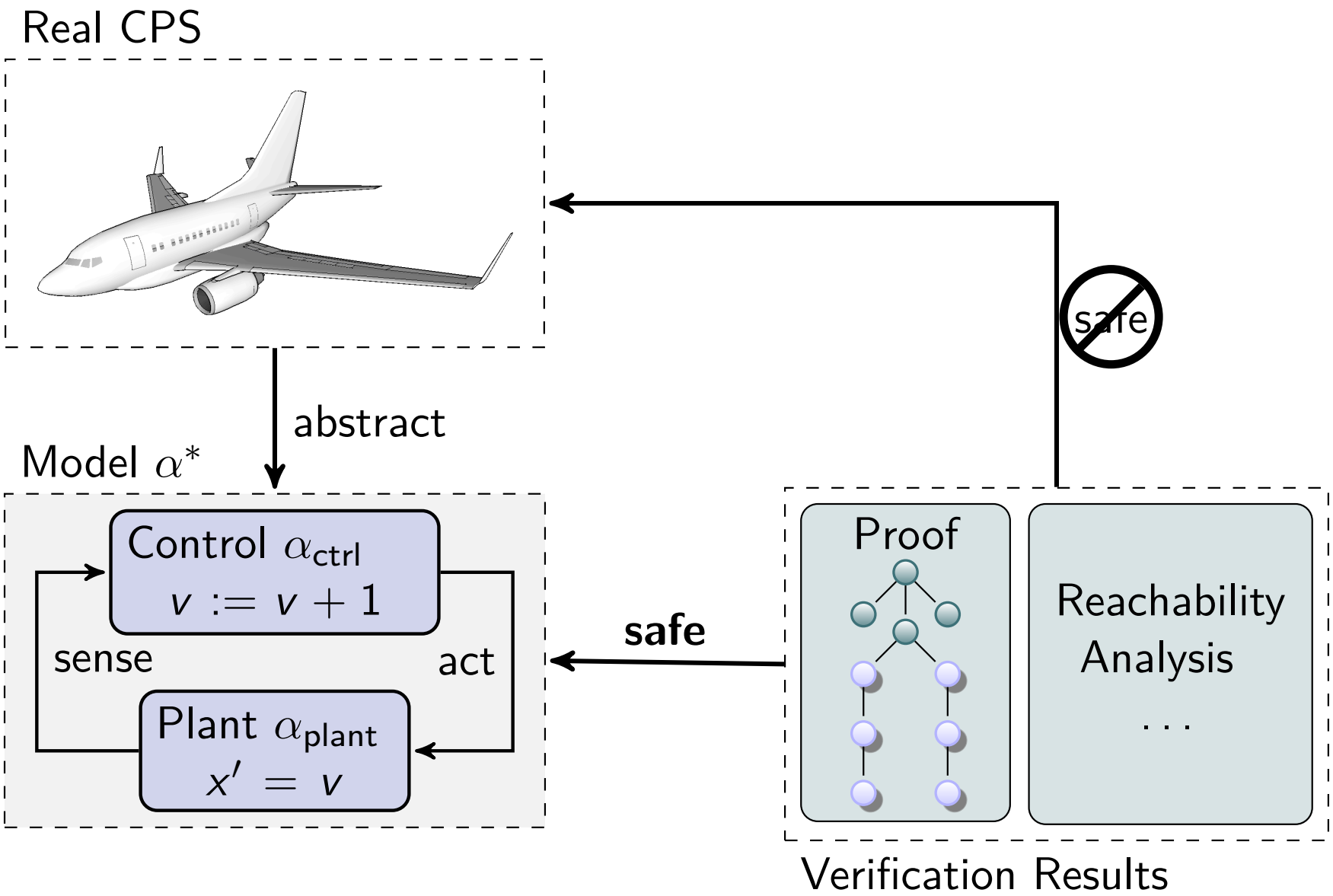
	≈LOC		
KeYmaera X	1 652		hybrid prover
KeYmaera	65 989		
KeY	51 328		Java
Nuprl	15 000	+ 50 000	general math
MetaPRL	8 196		
Isabelle/Pure	8 913		
Coq	16 538		
HOL Light	396		
PHAVer	30 000		hybrid verifier
HSolver	20 000		
SpaceEx	100 000		
Flow*	25 000		
dReal	50 000	+ millions	
HyCreate2	6 081	+ user model analysis	

Disclaimer: Self-reported estimates of the soundness-critical lines of code + rules

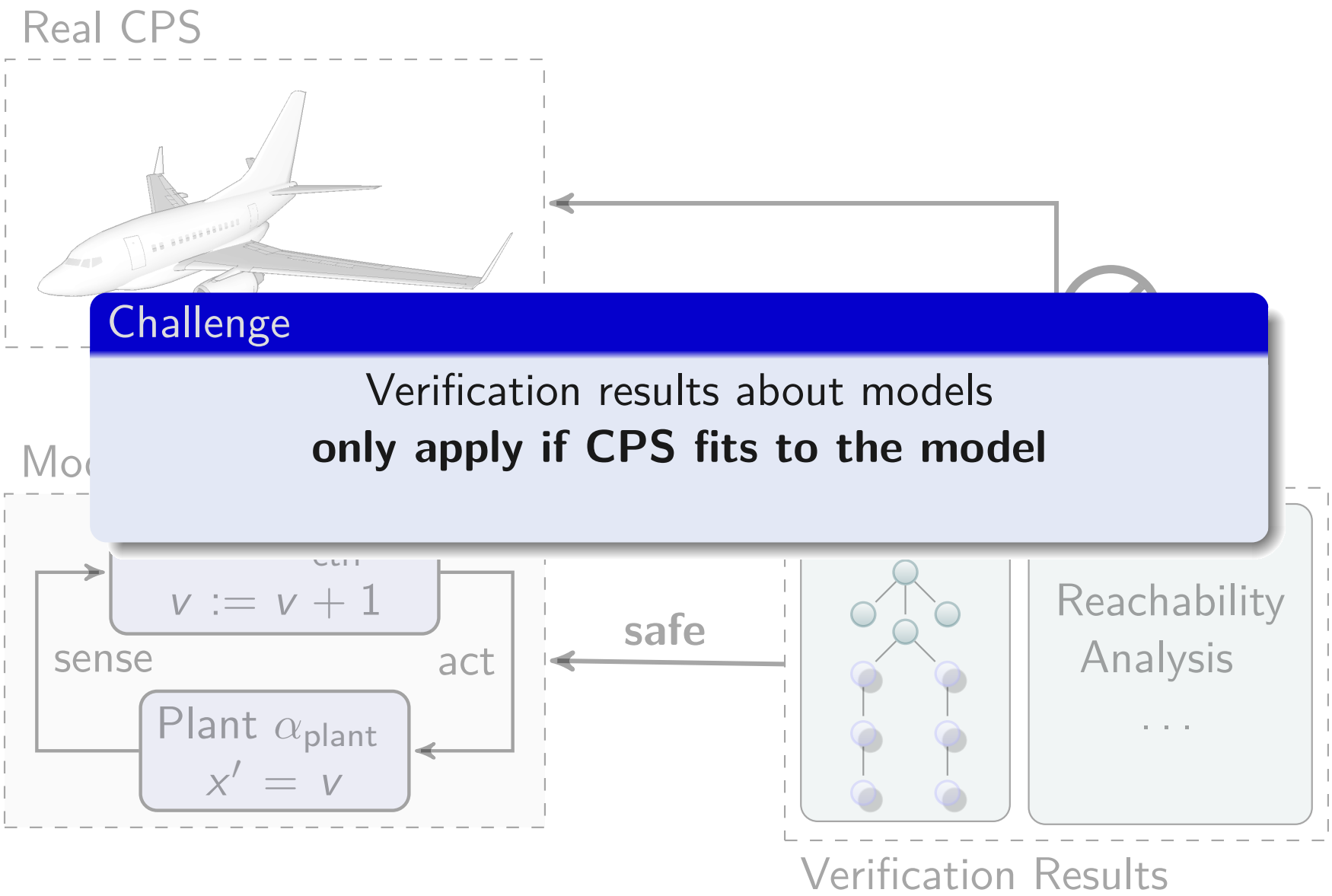
Formal Verification in CPS Development



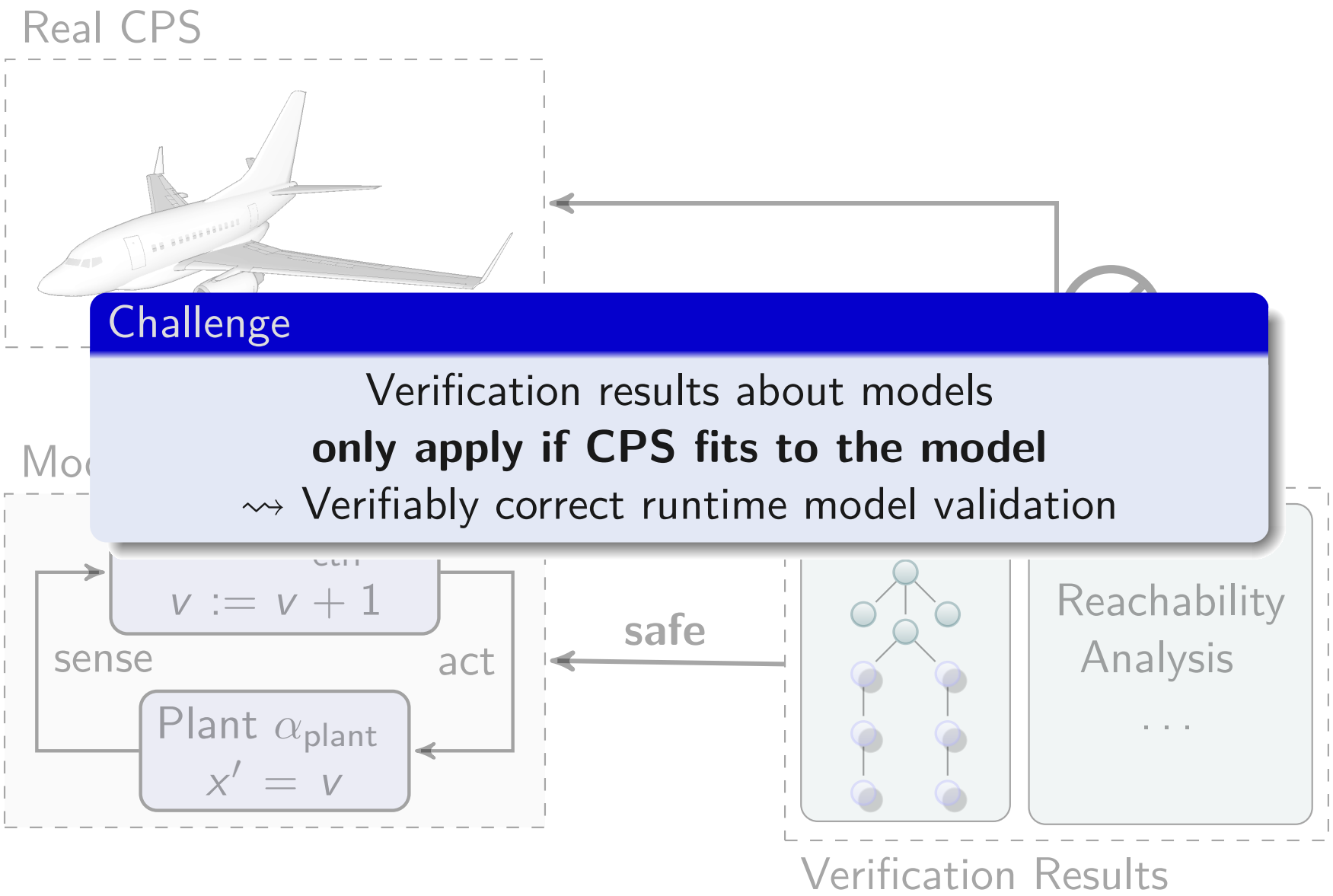
Formal Verification in CPS Development



Formal Verification in CPS Development



Formal Verification in CPS Development

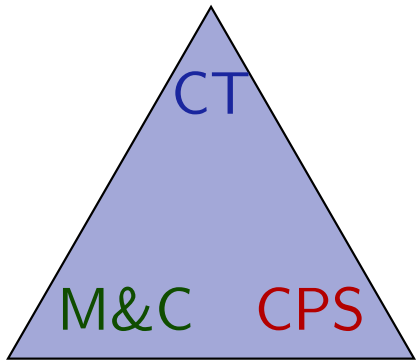


Outline

Learning Objectives

Verified Models & Verified Runtime Validation

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

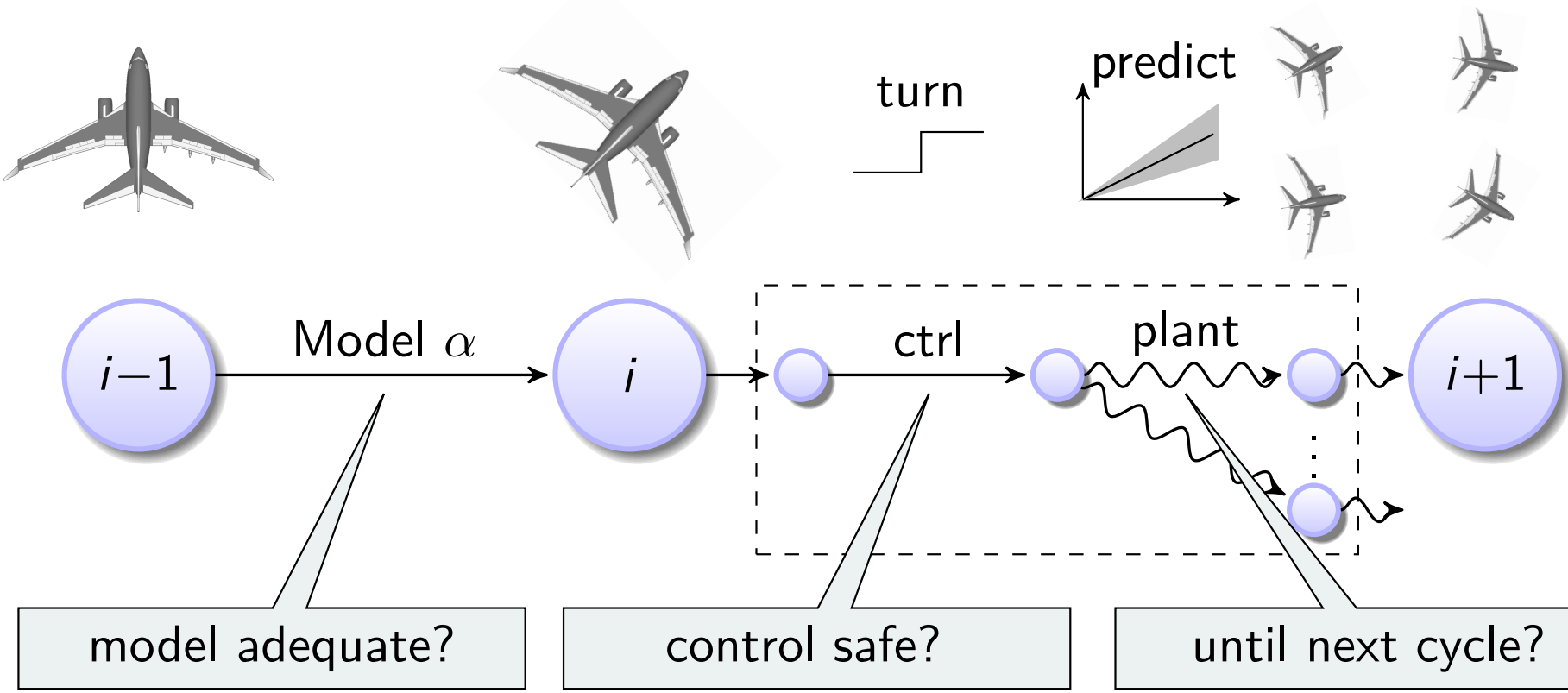
Contribution

ModelPlex ensures that verification results about models apply to CPS implementations

Outline

ModelPlex Runtime Model Validation

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

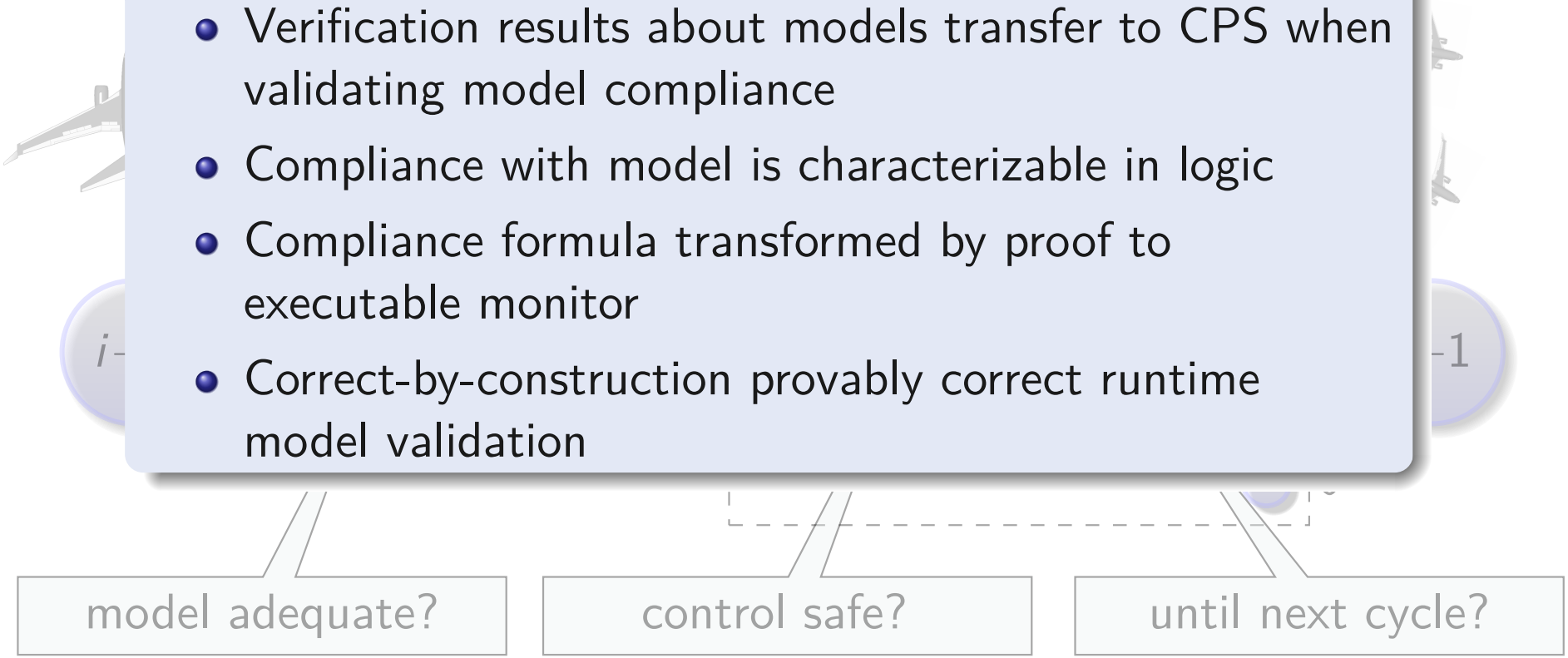


ModelPlex Runtime Model Validation

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

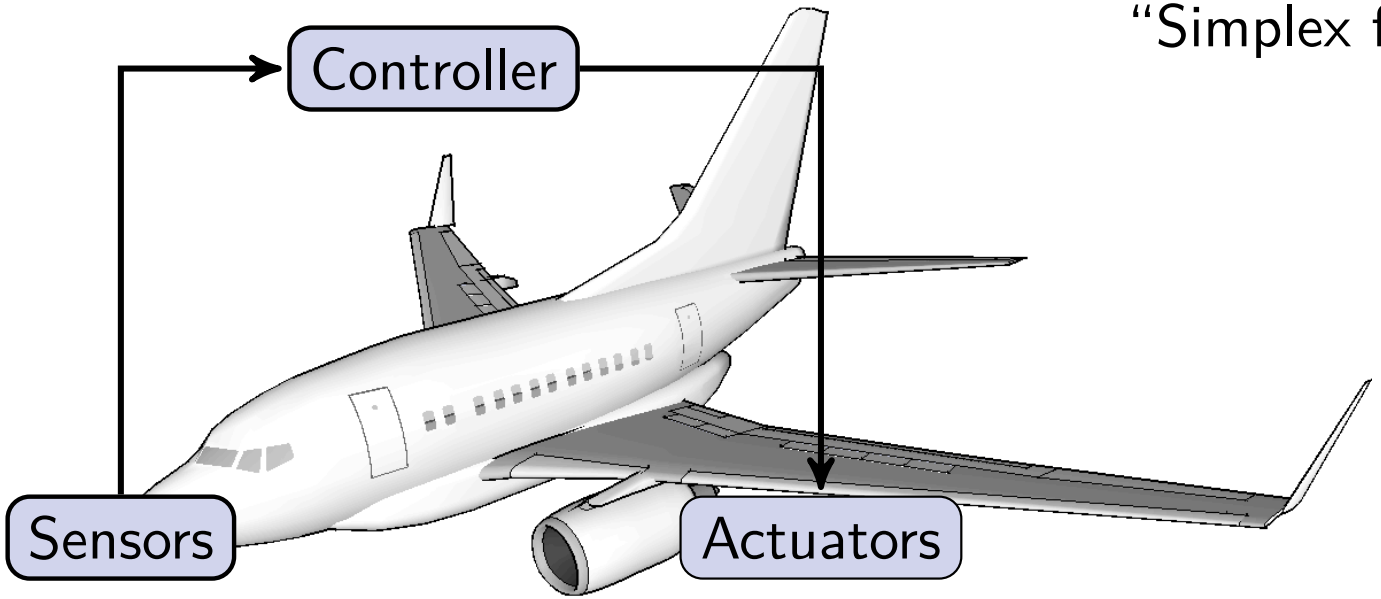
Contributions

- 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

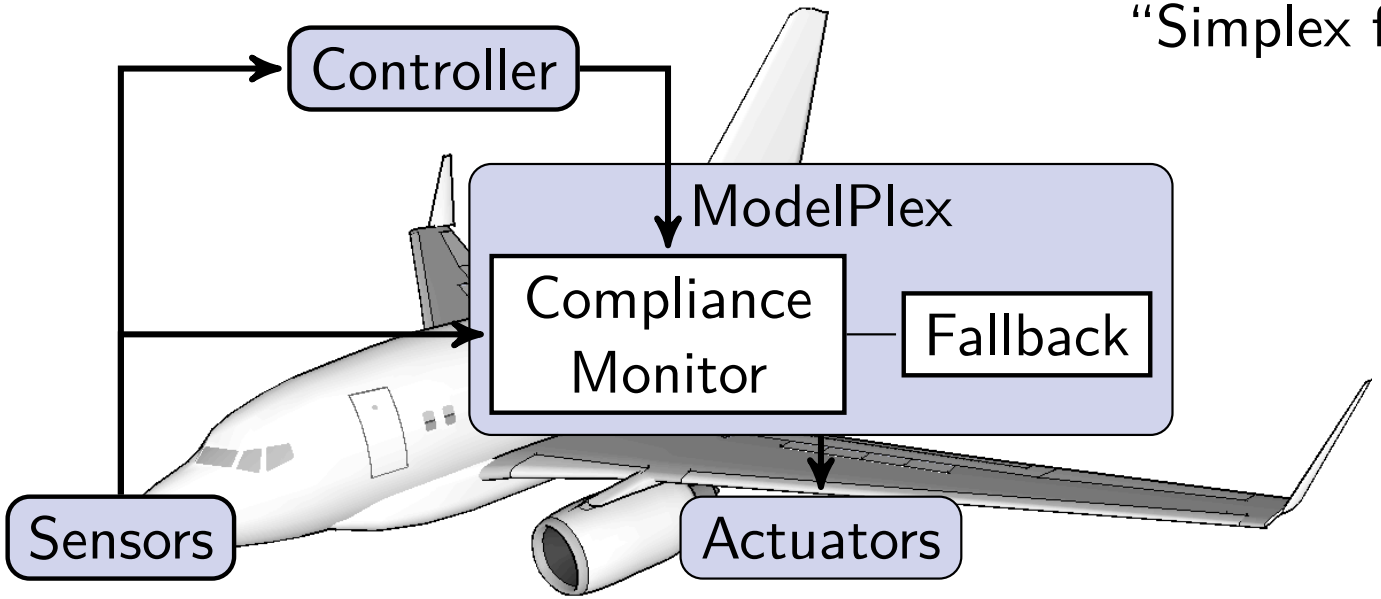


ModelPlex at Runtime

"Simplex for Models"



ModelPlex at Runtime



Compliance Monitor Checks CPS for compliance with model at runtime

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

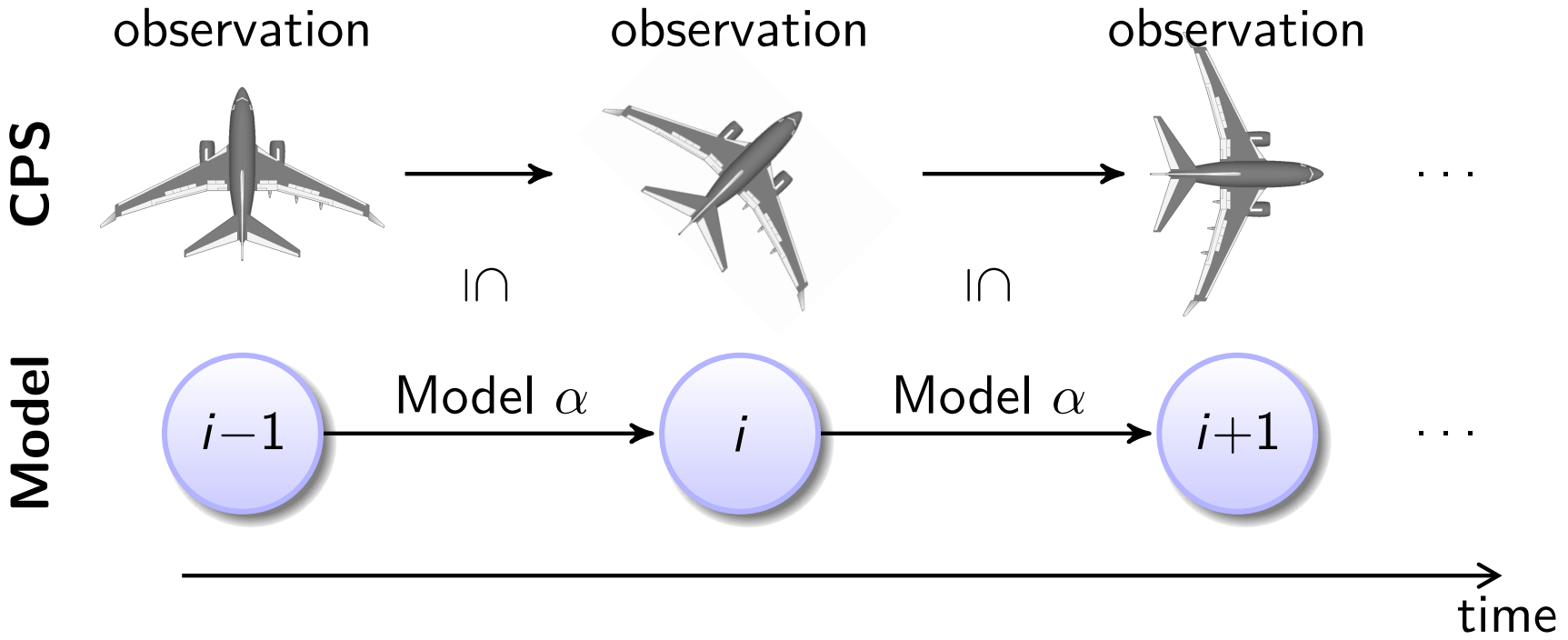
Fallback Safe action, executed when monitor is not satisfied (veto)

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

ModelPlex Compliance

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

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

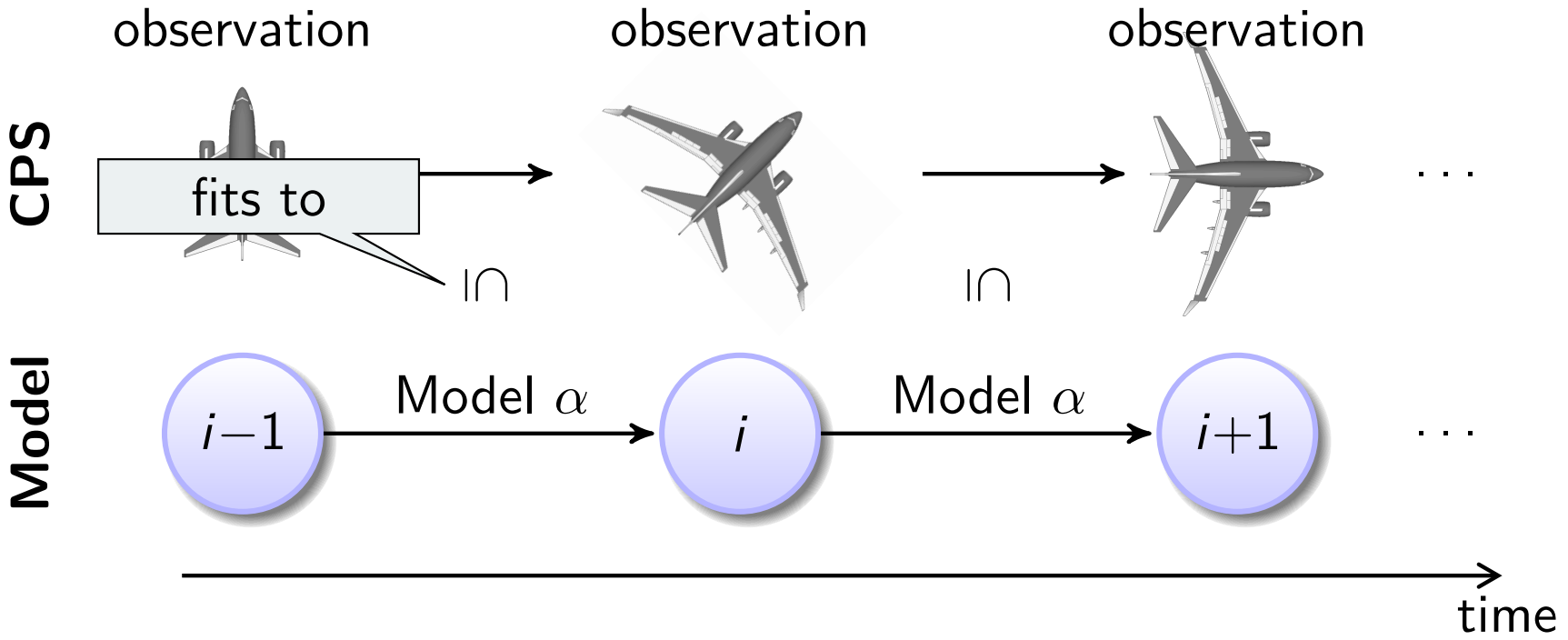


Detect non-compliance ASAP to initiate fallback actions while still safe

ModelPlex Compliance

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

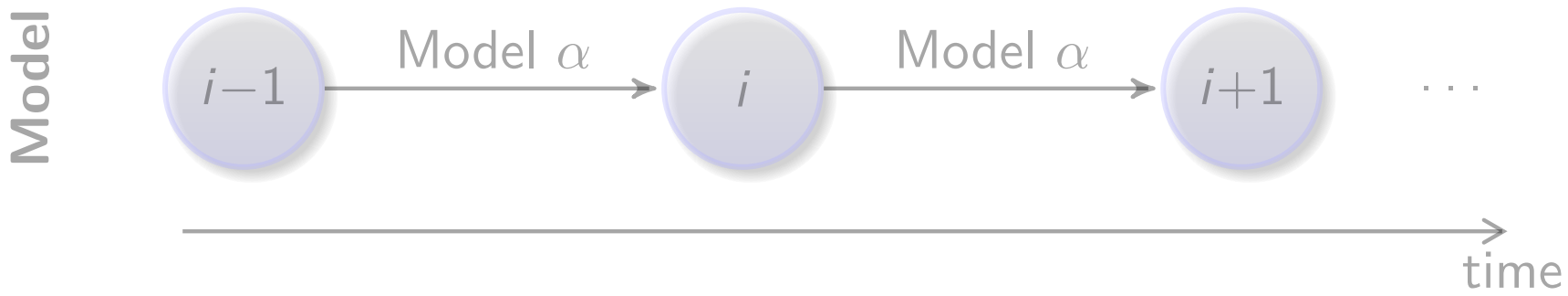
ModelPlex Compliance

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

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

Challenge

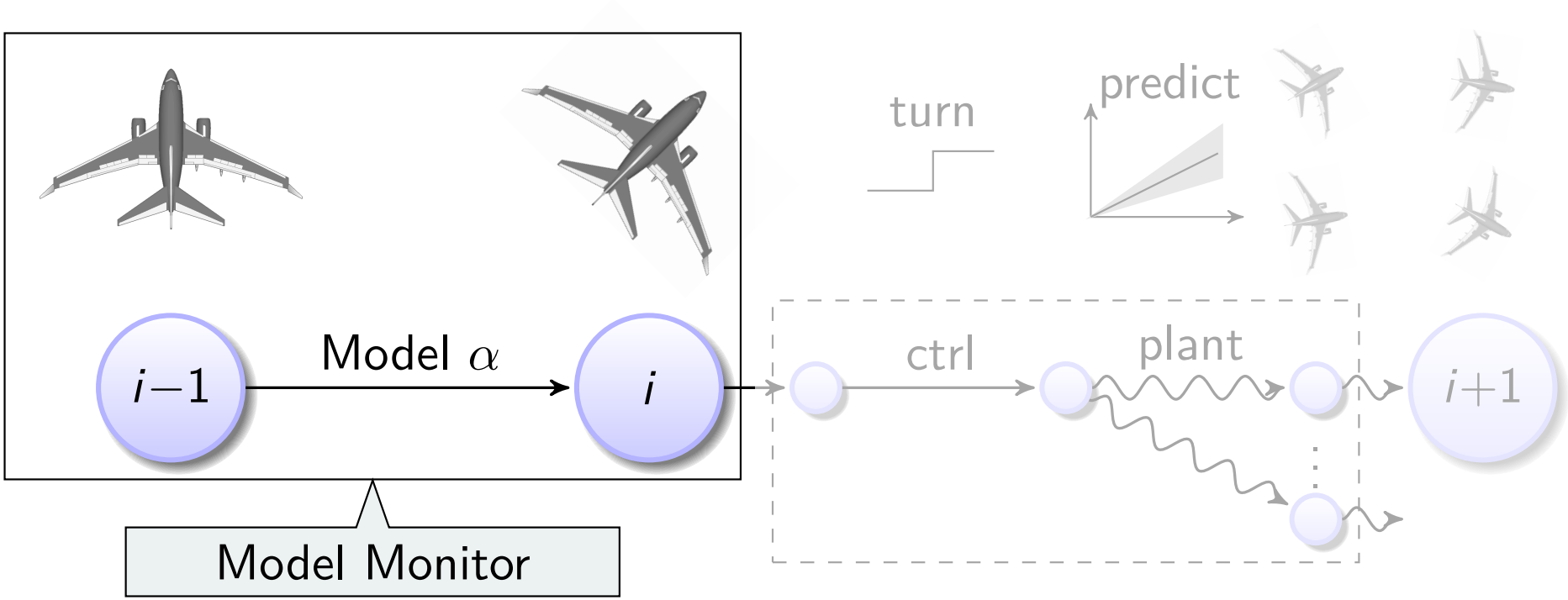
Model describes behavior,
but at runtime we get sampled observations
$\rightsquigarrow$ **Transform model into observation-monitor**



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

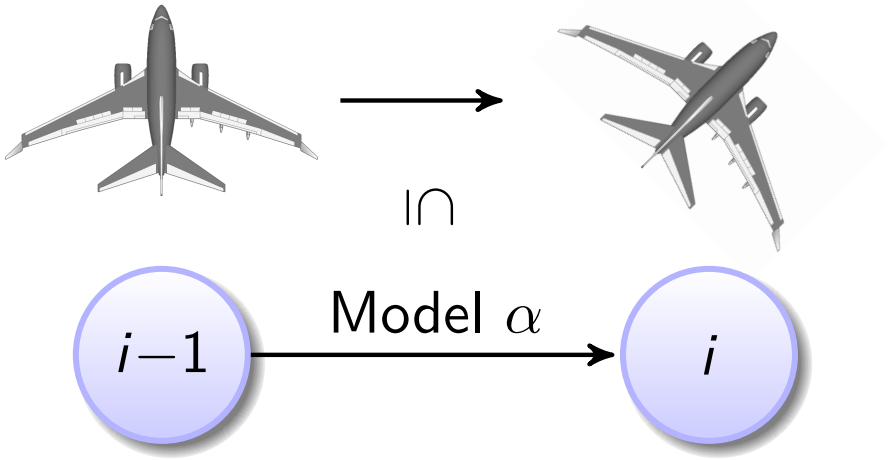
Outline

Outline



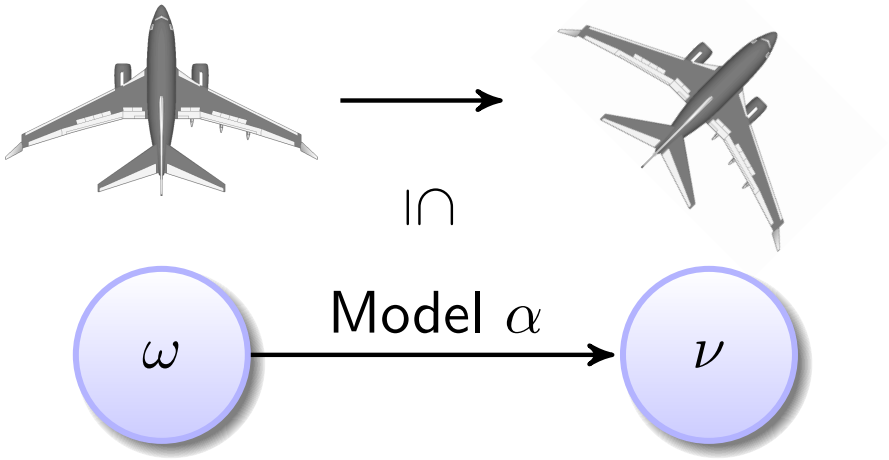
Characterizing State Relations in Logic

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



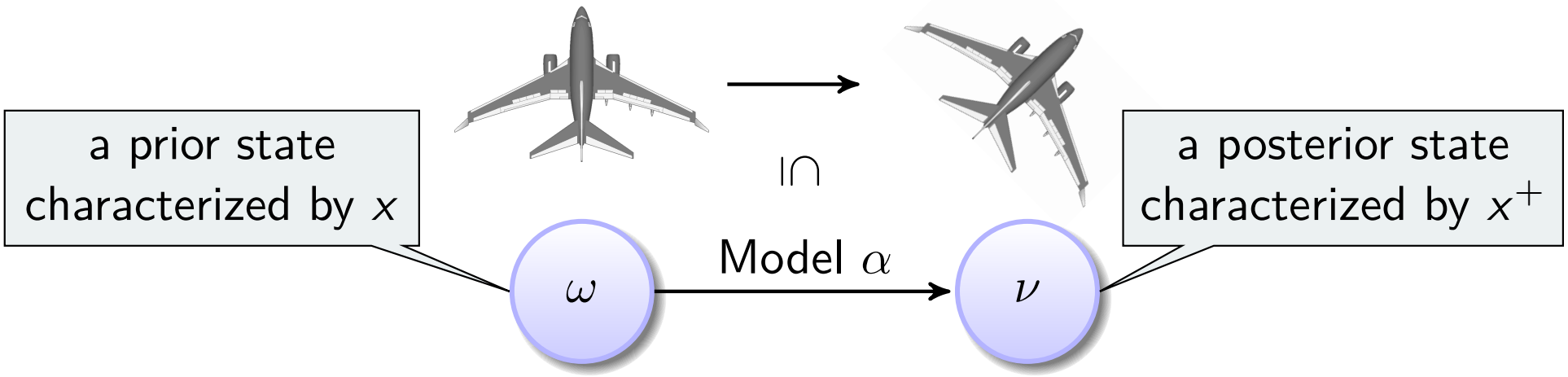
Characterizing State Relations in Logic

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



Characterizing State Relations in Logic

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?

a prior state characterized by x

a posterior state characterized by x^+

ω — Model α → ν

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 $x = x^+$

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^+

ω — Model α → ν

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?

$\upharpoonright\!\cap$

a prior state characterized by x

a posterior state characterized by x^+

ω — Model α → ν

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 $x = x^+$

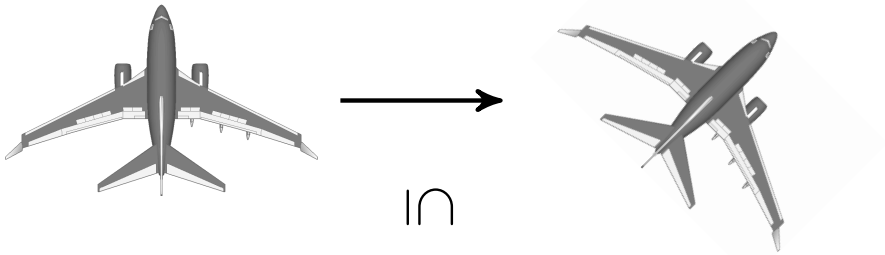
$\Uparrow$ d$\mathcal{L}$ proof

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

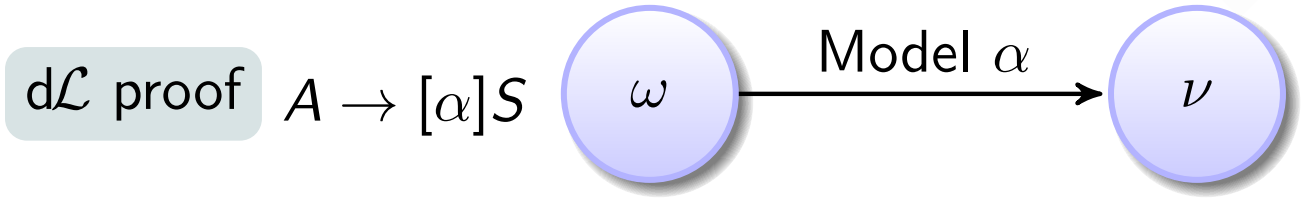
check at runtime (efficient)

Logical Reductions for Model Safety Transfer

Logic reduces CPS safety to runtime monitor with offline proof



$\in$



Offline

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

$\Updownarrow$ Lemma

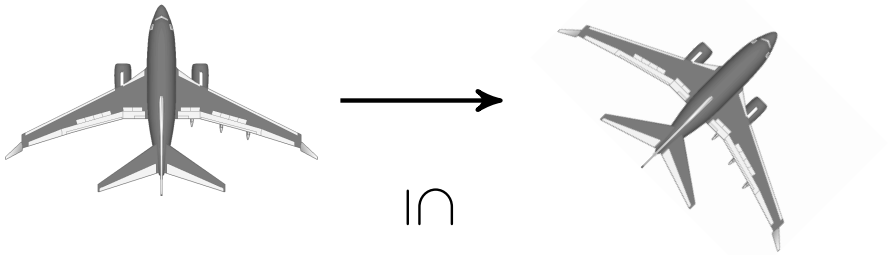
Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ dℒ proof

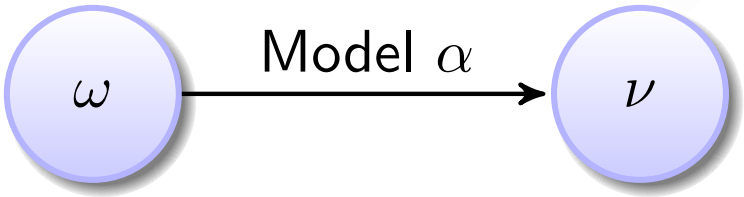
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



$\in$



Offline

Init $\omega \models A$

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

$\Updownarrow$ Lemma

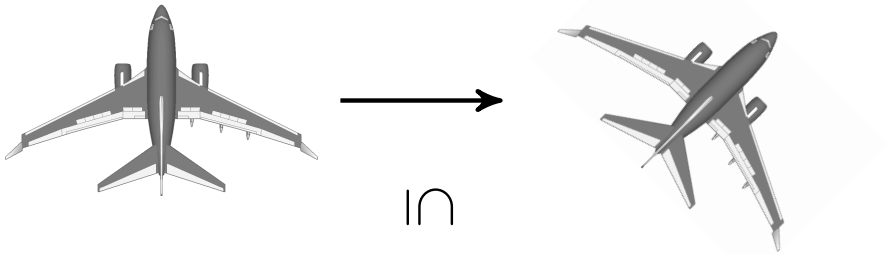
Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ dℒ proof

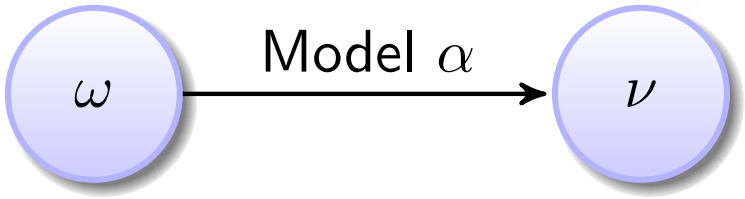
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



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



Offline

Init $\omega \models A$

Safe $\nu \models S$

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

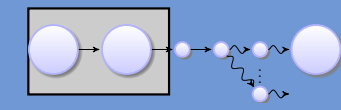
$\Updownarrow$ Lemma

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

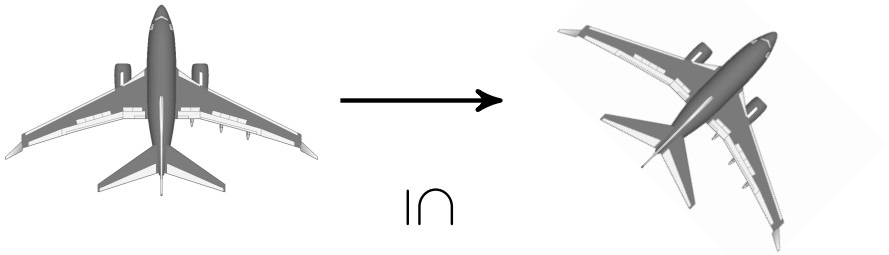
$\Uparrow$ d$\mathcal{L}$ proof

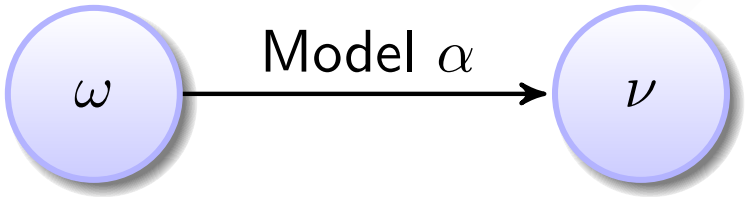
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)



Logic reduces CPS safety to runtime monitor with offline proof





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

Offline

Init $\omega \models A$

Safe $\nu \models S$

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

$\Updownarrow$ Lemma

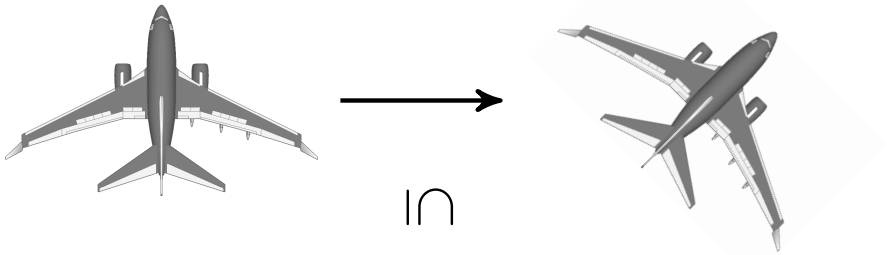
Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

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

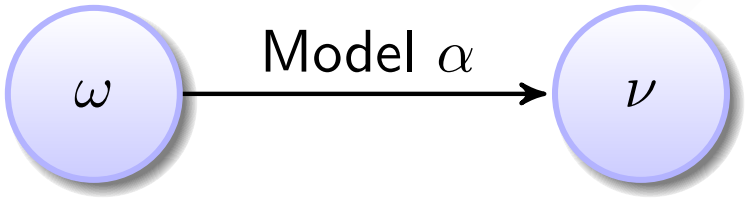
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



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



Offline

Init $\omega \models A$

Safe $\nu \models S$

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

$\Updownarrow$ Lemma

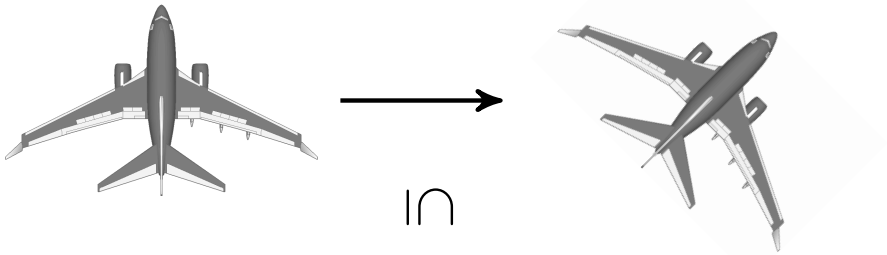
Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

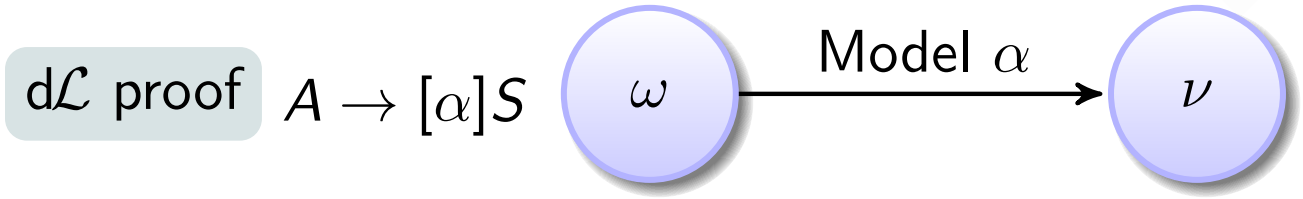
$\Uparrow$ d$\mathcal{L}$ proof

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

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof





Offline

Init $\omega \models A$

Safe $\nu \models S$

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

$\Updownarrow$ Lemma

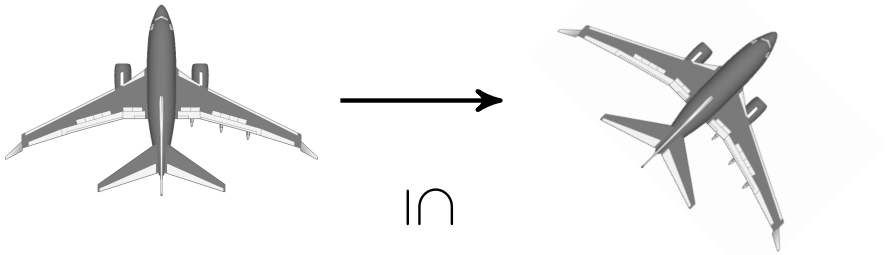
Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ dℒ proof

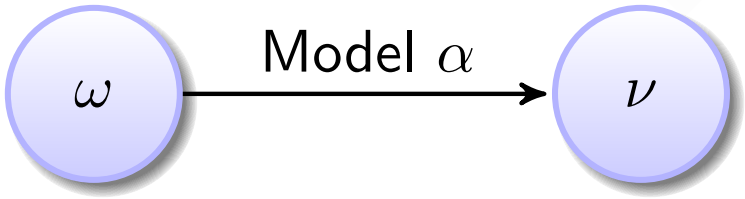
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof



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



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

Offline

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

$\Updownarrow$ Lemma

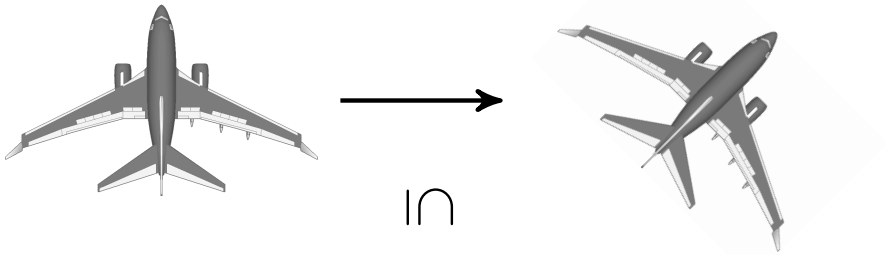
Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

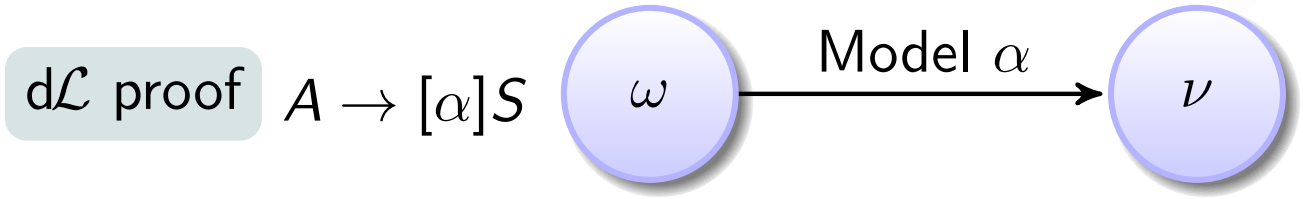
$\Uparrow$ d$\mathcal{L}$ proof

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

check at runtime (efficient)

Logic reduces CPS safety to runtime monitor with offline proof





Offline

Init $\omega \models A$

Safe $\nu \models S$

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

$\Updownarrow$ Lemma

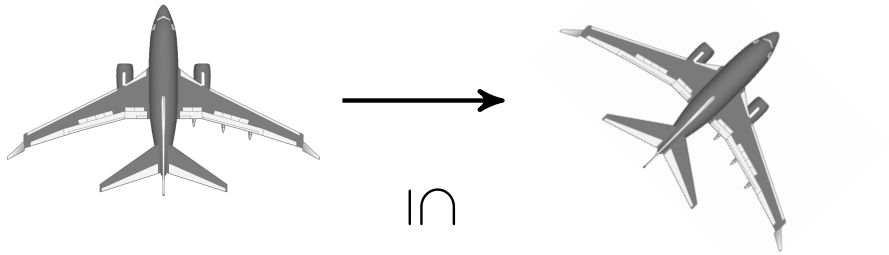
Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ 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...

d$\mathcal{L}$ 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

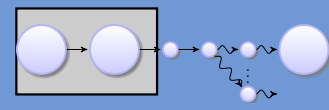
Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

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

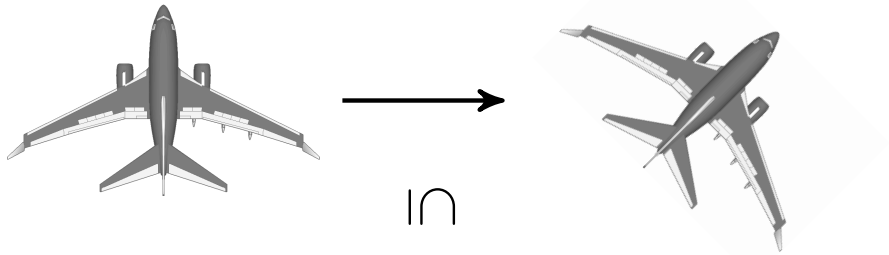
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

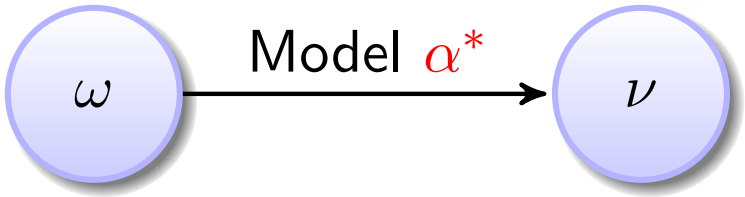
Logical Reductions for α^* Model Safety Transfer



Logic reduces CPS safety to runtime monitor with offline proof



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



Offline

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

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

$\Updownarrow$ Lemma

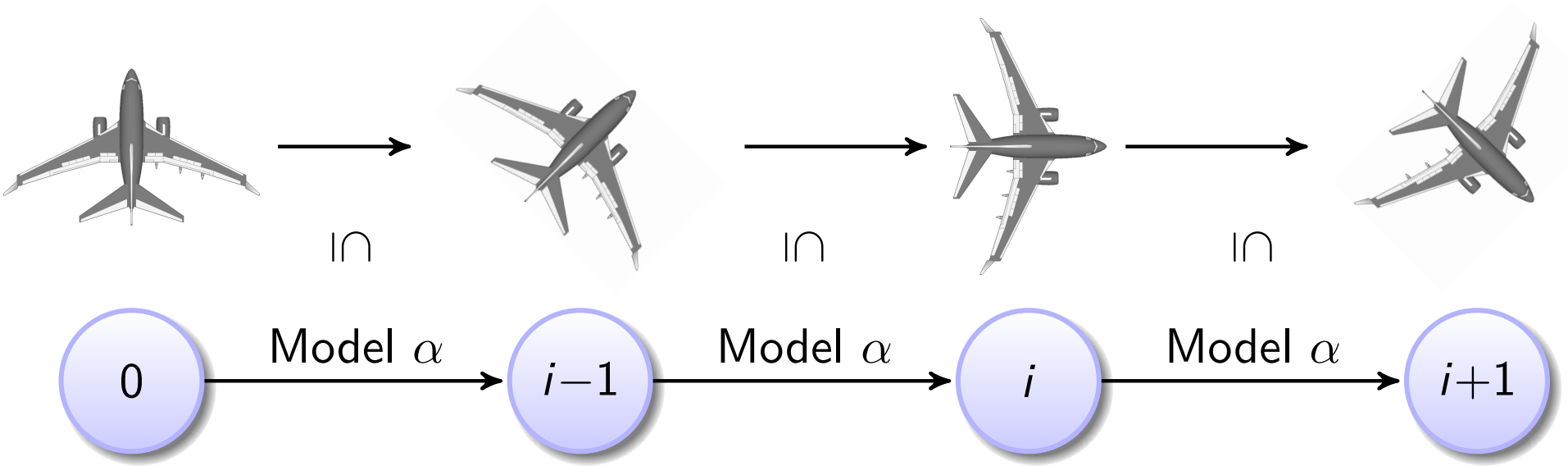
Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha^*\rangle(x = x^+)$

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

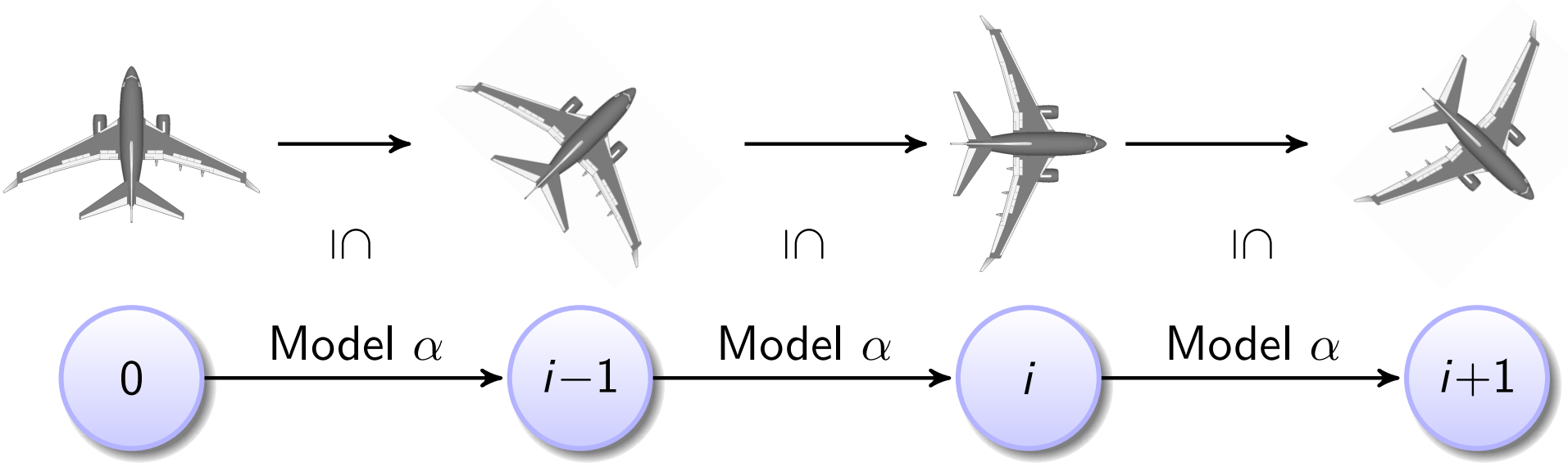
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

check at runtime (efficient)

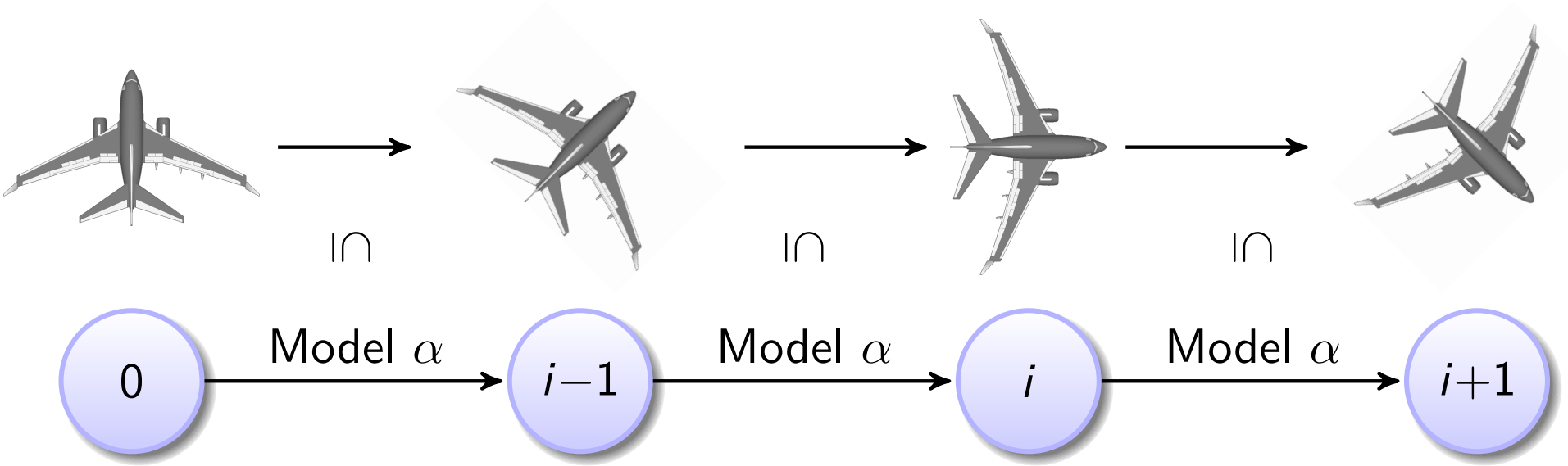
ModelPlex Model Monitor Correctness



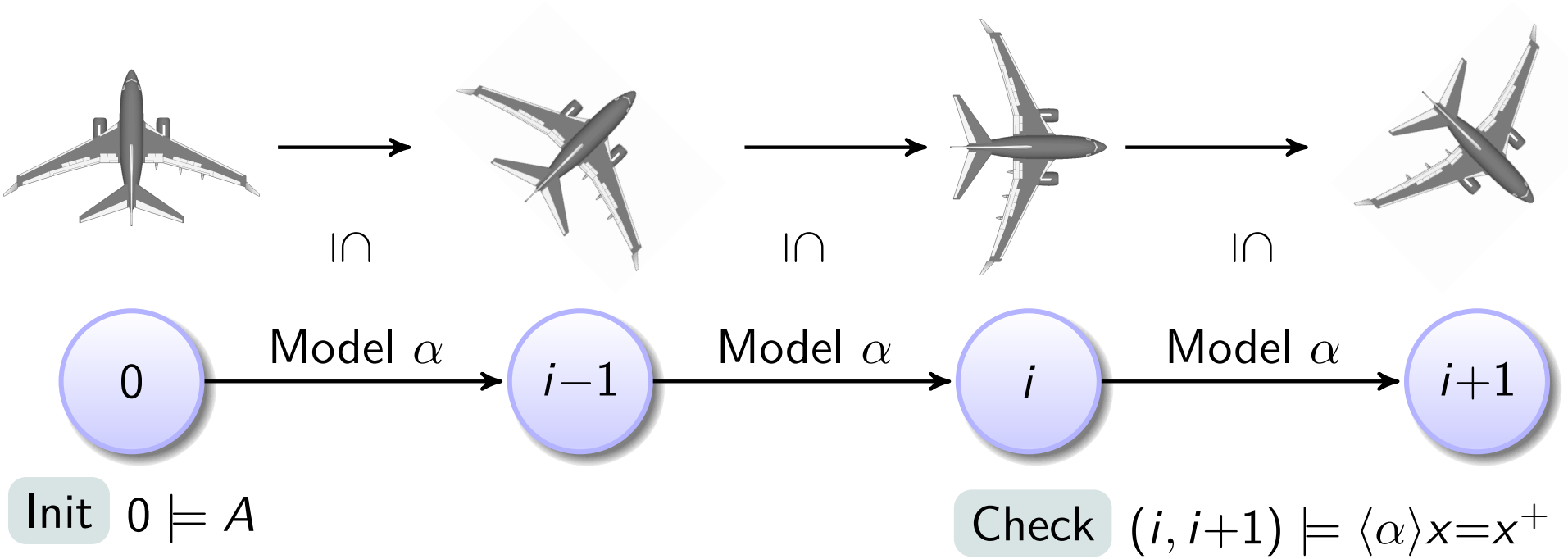
ModelPlex Model Monitor Correctness



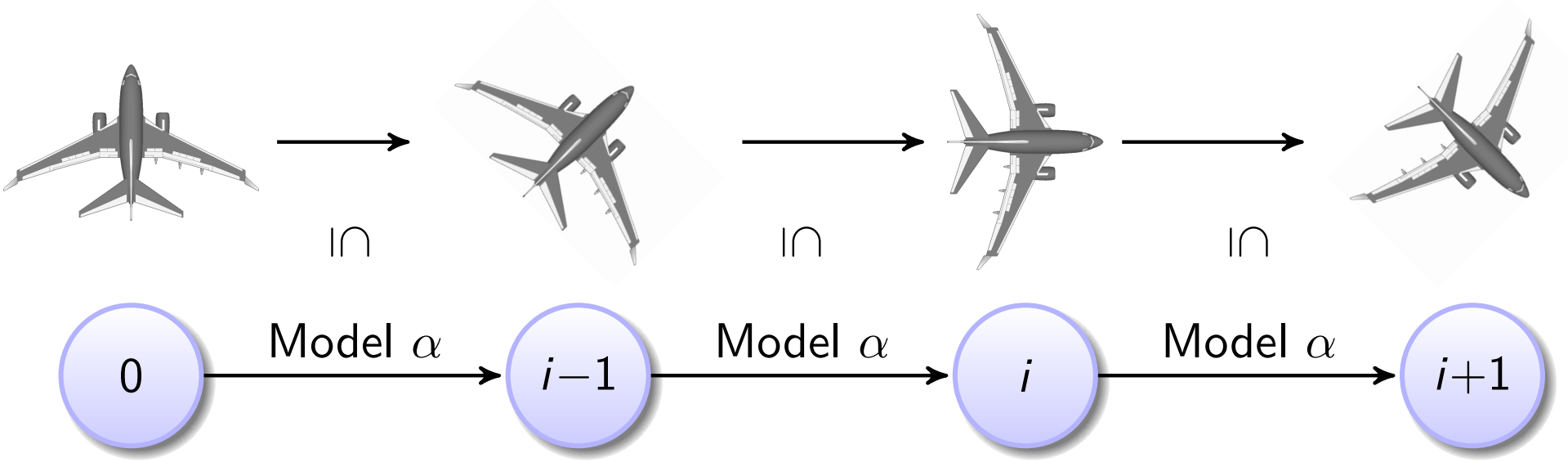
ModelPlex Model Monitor Correctness



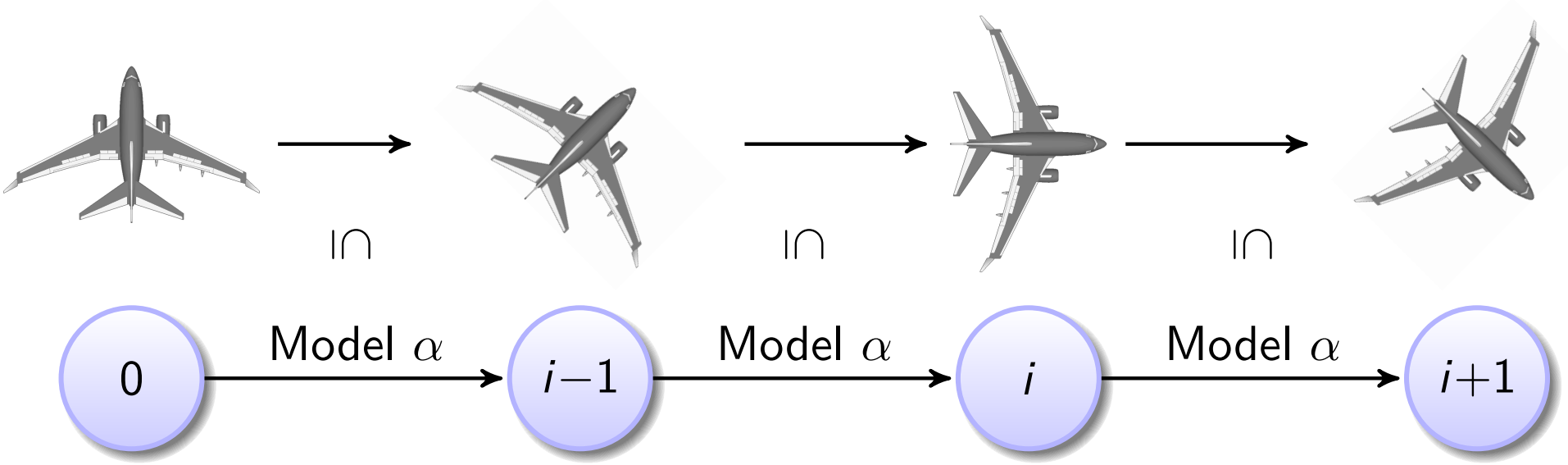
ModelPlex Model Monitor Correctness



ModelPlex Model Monitor Correctness



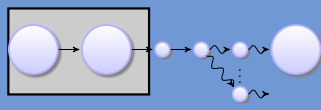
ModelPlex Model Monitor Correctness



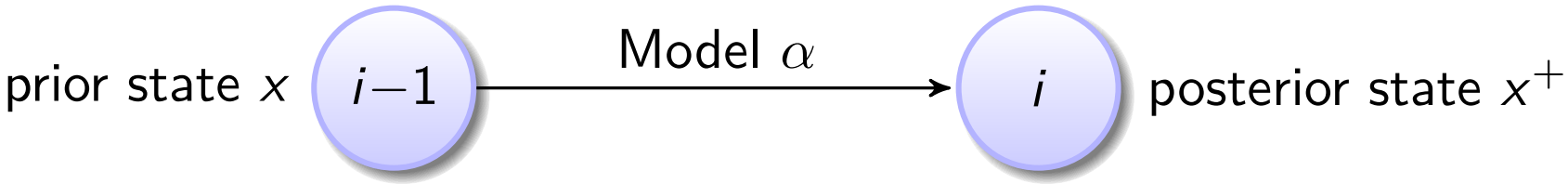
Theorem (Model Monitor Correctness) (FMSD'16)

"System safe as long as monitor satisfied."

Provably Correct Synthesis of Monitors



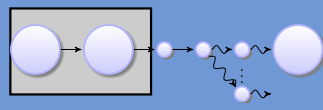
- Proof calculus of d$\mathcal{L}$ executes models symbolically



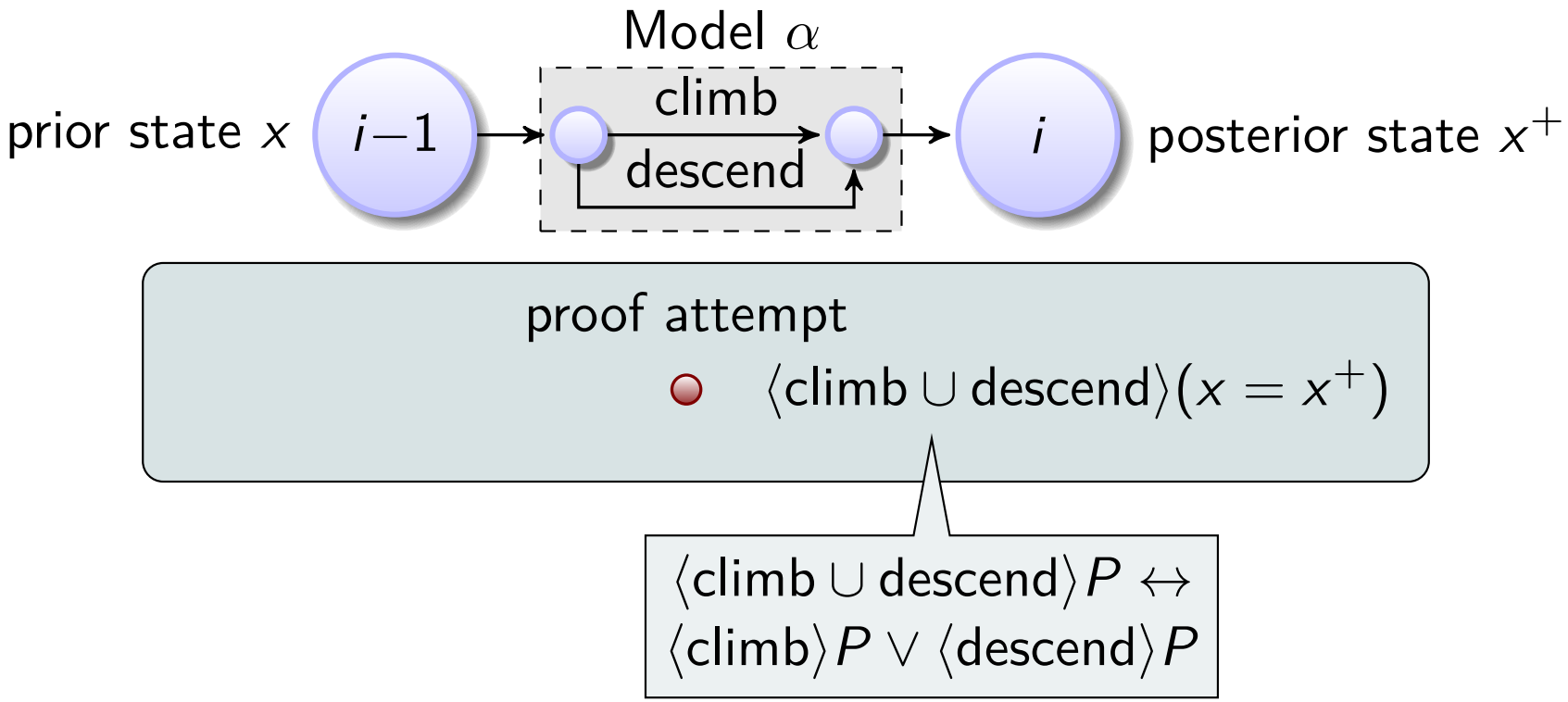
proof attempt

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

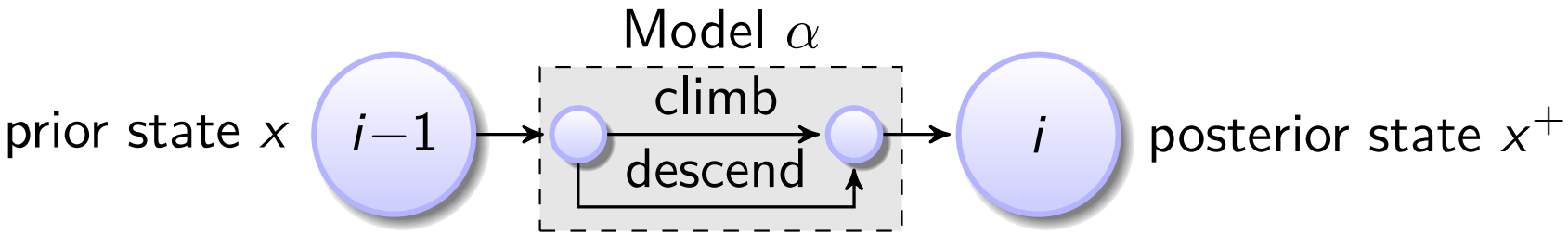
Provably Correct Synthesis of Monitors



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



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



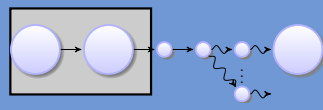
proof attempt

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

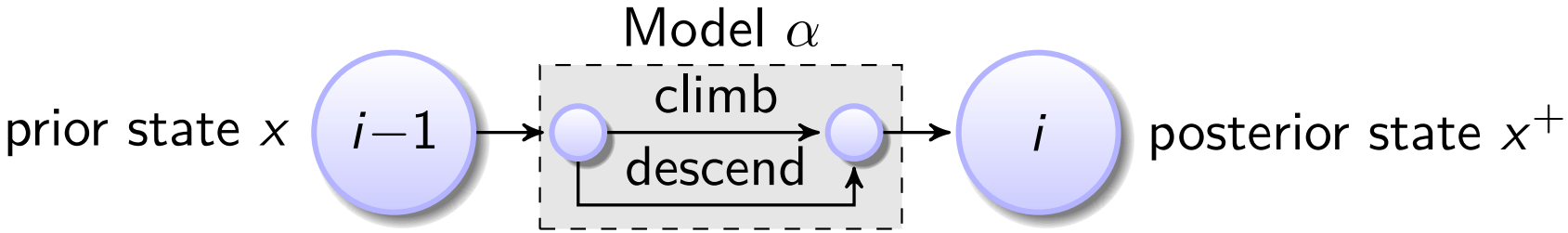
$\vee$

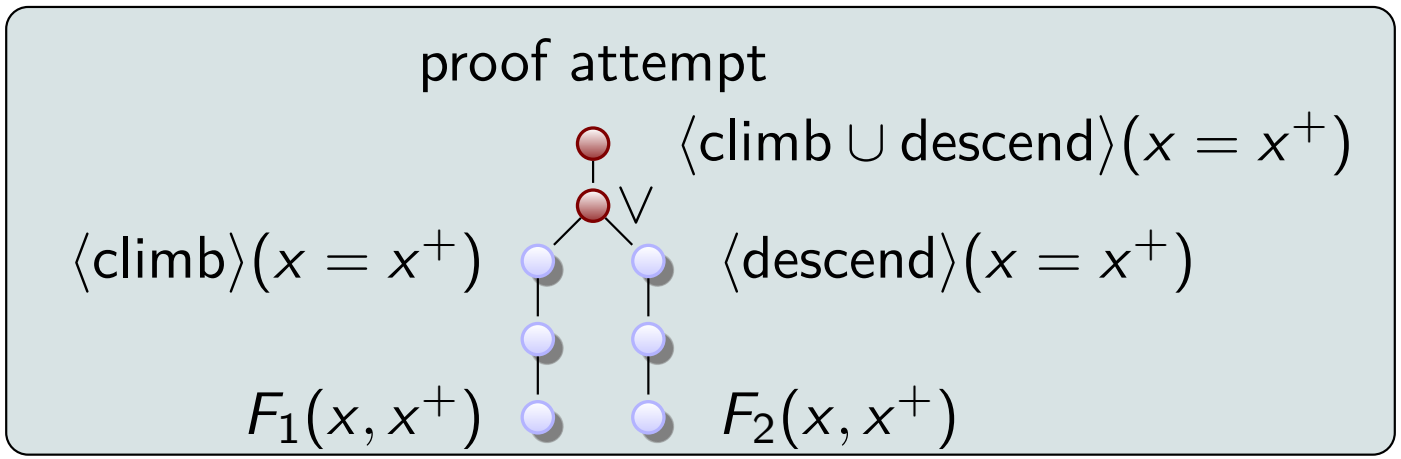
$\langle \text{climb} \rangle (x = x^+)$ $\quad$ $\langle \text{descend} \rangle (x = x^+)$

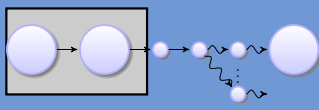
Provably Correct Synthesis of Monitors



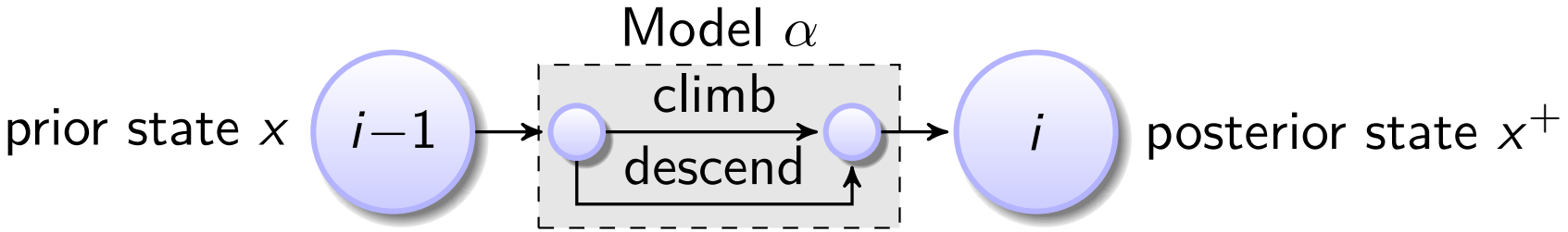
- Proof calculus of d$\mathcal{L}$ executes models symbolically







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



proof attempt

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

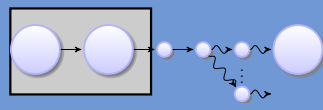
$\vee$

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

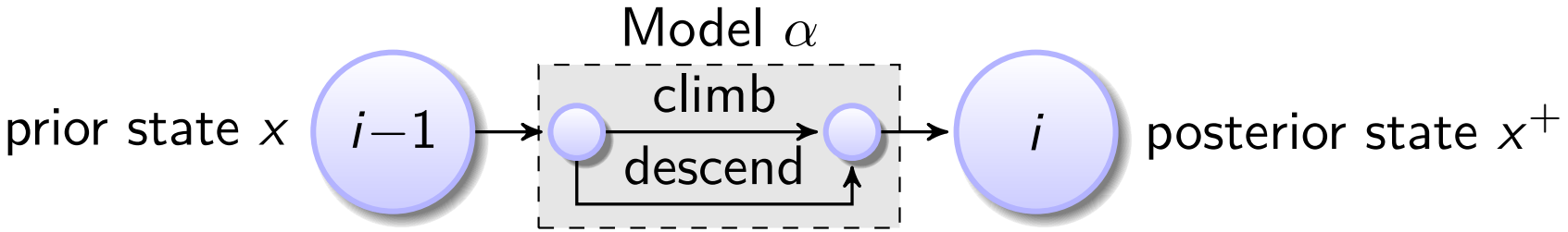
$F_1(x, x^+)$ $\quad$ $F_2(x, x^+)$

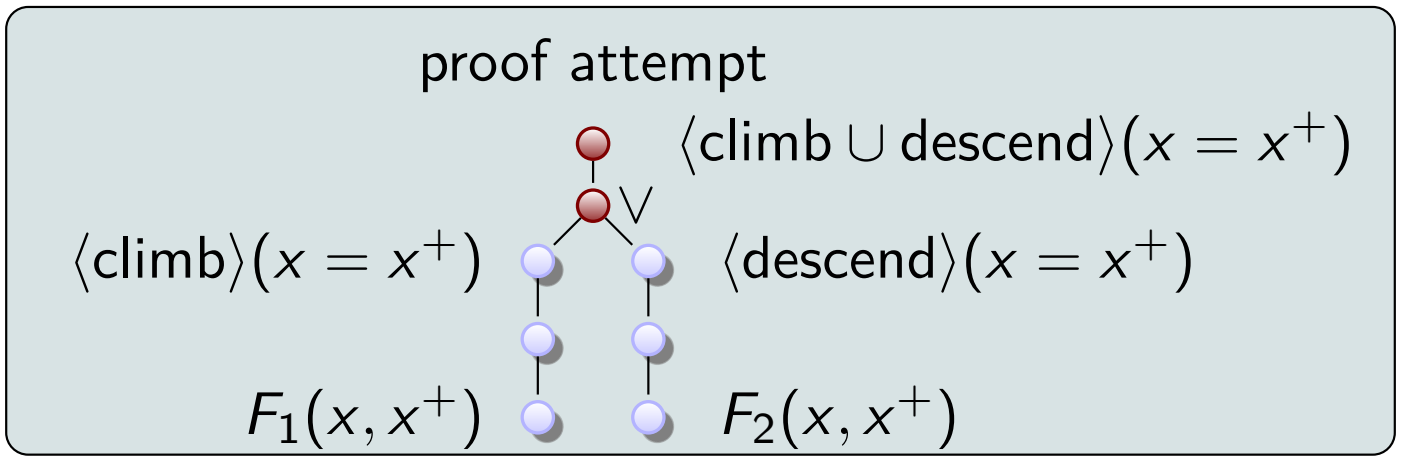
Monitor: $F_1(x, \overbrace{x^+) \vee F_2(x, x^+)}$

Provably Correct Synthesis of Monitors



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

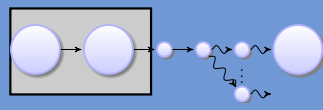




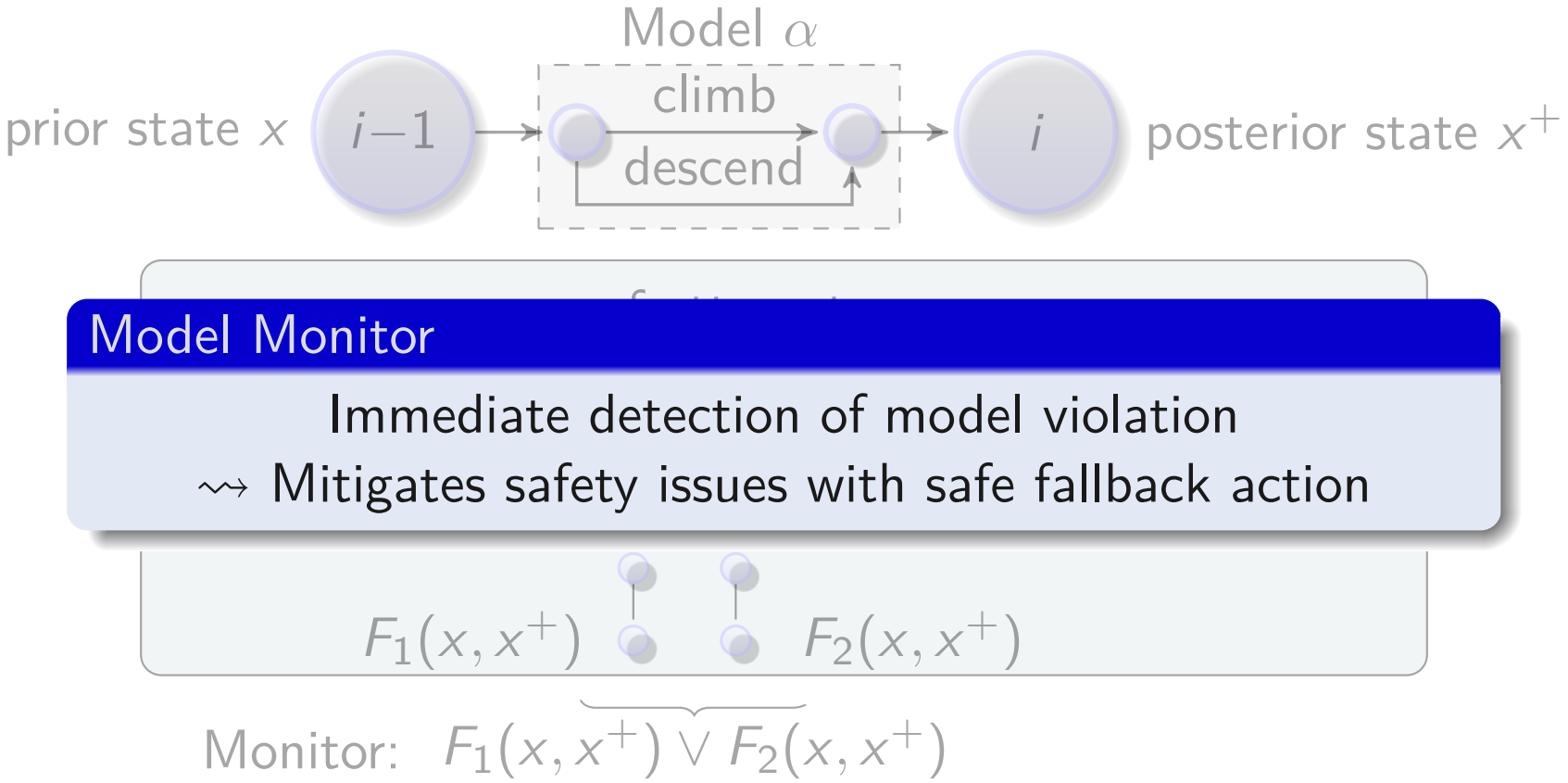
Monitor: $F_1(x, x^+) \vee F_2(x, x^+)$

- 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



- 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

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

Model Monitor Specification Conjecture

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

Water Tank Example: Nondeterministic Assignment

Proof Rules

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

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

$$WR \ \frac{\Gamma \vdash \Delta}{\Gamma \vdash P, \Delta}$$

Sequent Deduction

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

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

Water Tank Example: Differential Equations

Proof Rules

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

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

Sequent Deduction

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash F = f^+ \wedge x^+ = x + Ft^+ \wedge t^+ \geq 0 \wedge x \geq 0 \wedge \varepsilon \geq t^+ \geq 0 \wedge Ft^+ + x \geq 0}{A \vdash \forall 0 \leq \tilde{t} \leq T \ (x + f^+ \tilde{t} \geq 0 \wedge \tilde{t} \leq \varepsilon) \wedge F = f^+ \wedge x^+ = x + Ft^+ \wedge t^+ = t^+}\,\text{QE}}{A \vdash \exists T \geq 0 ((\forall 0 \leq \tilde{t} \leq T \ (x + f^+ \tilde{t} \geq 0 \wedge \tilde{t} \leq \varepsilon)) \wedge F = f^+ \wedge (x^+ = x + FT \wedge t^+ = T))}\,\exists\text{R,WR}}{A \vdash \langle f := F; t := 0 \rangle \langle \{x' = f, t' = 1 \ \&\ x \geq 0 \wedge t \leq \varepsilon\} \rangle \Upsilon^+}\,\langle' \rangle$$

Water Tank Example: Synthesized Model Monitor

Input: Model and Safety Property

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

Output: Synthesized Model Monitor

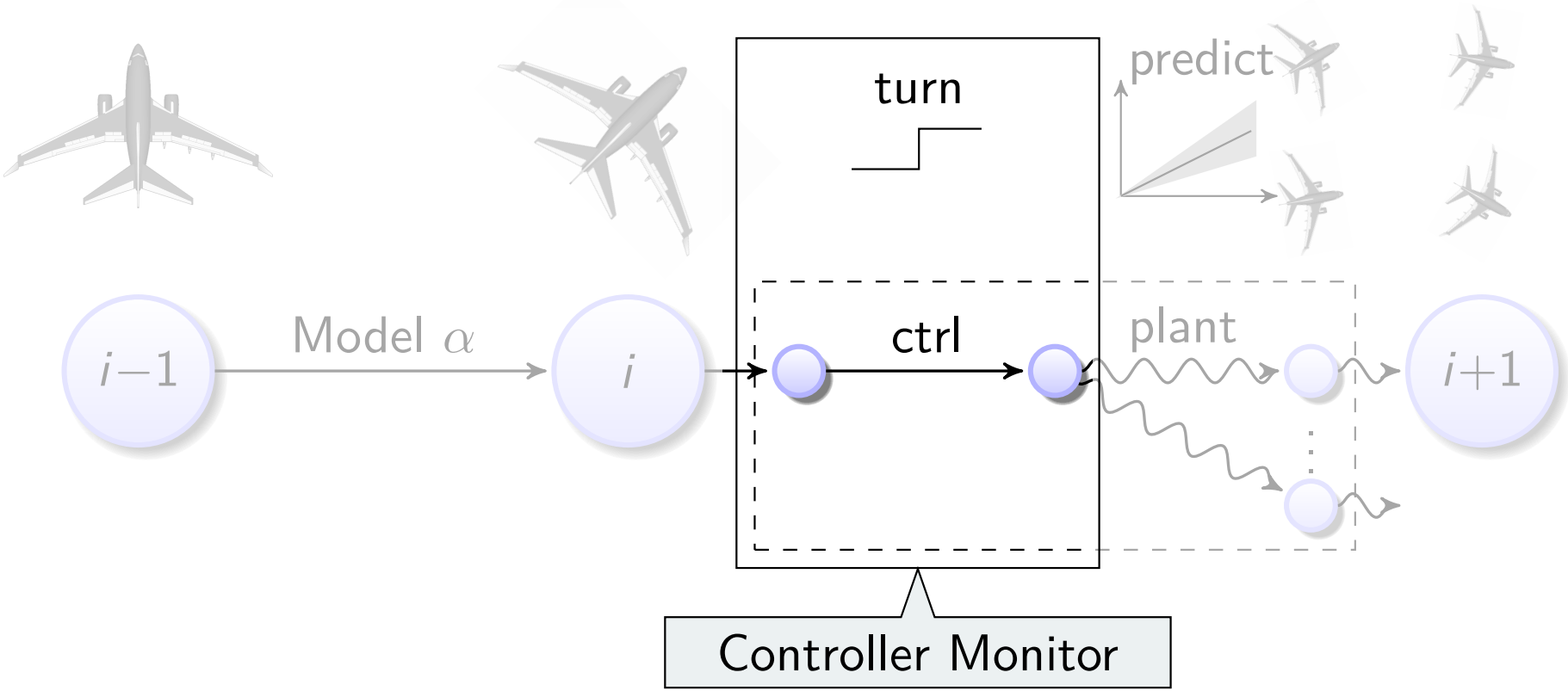
$$-1 \leq f^+ \leq \frac{m-x}{\varepsilon} \wedge x^+ = x + f^+ t^+ \wedge x \geq 0 \wedge x + f^+ t^+ \geq 0 \wedge \varepsilon \geq t^+ \geq 0$$

Proof (Generated by ModelPlex tactic).

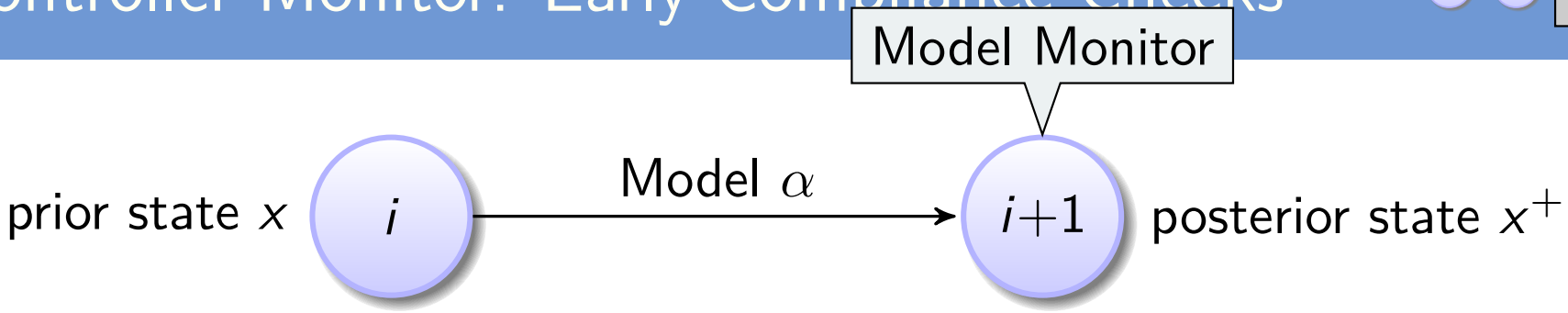
A proof of correctness of the synthesized model monitor. □

Outline

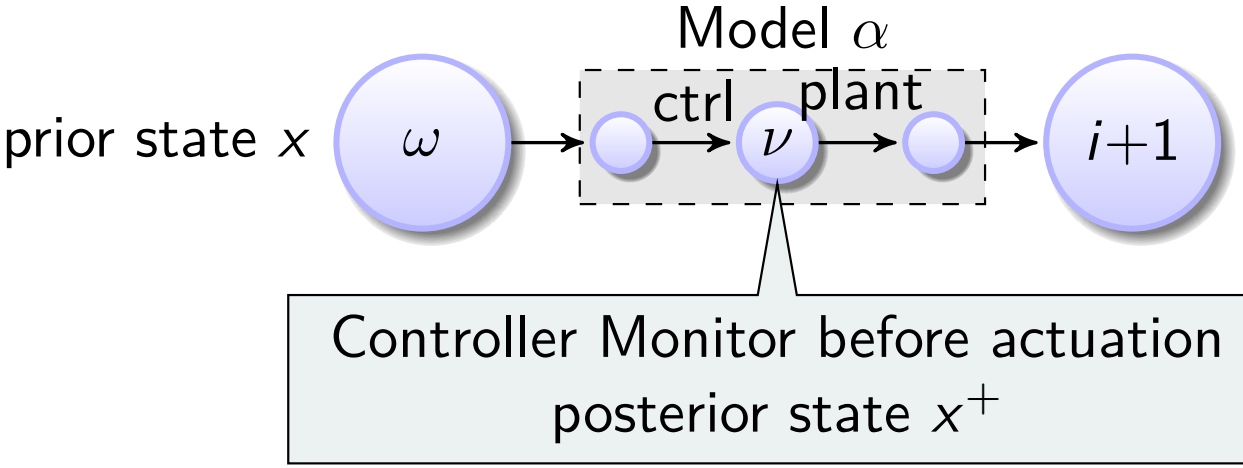
For typical models ctrl; plant we can check earlier



Controller Monitor: Early Compliance Checks

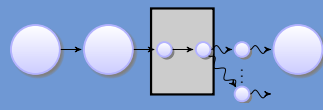


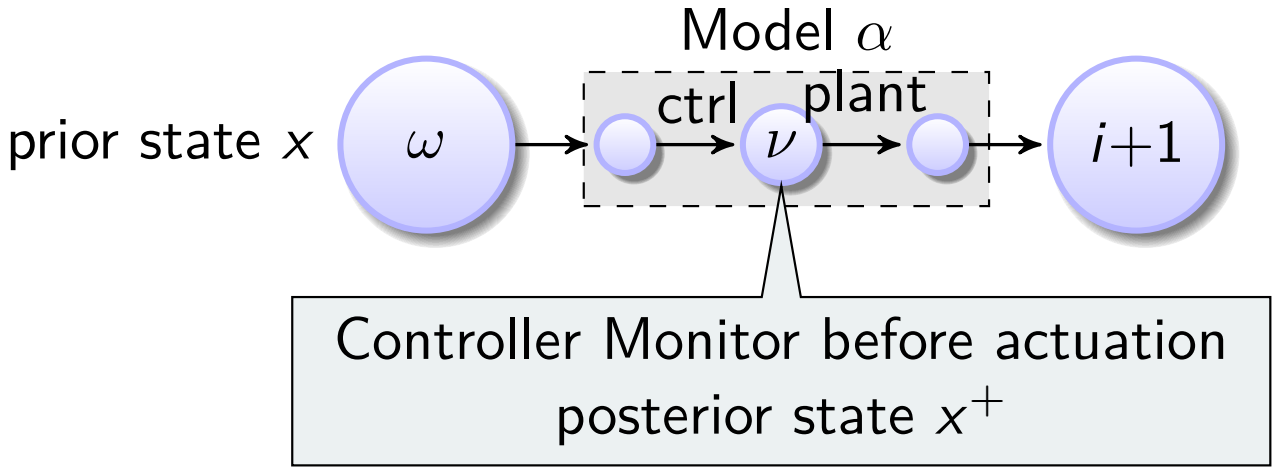
Controller Monitor: Early Compliance Checks



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

Controller Monitor: Early Compliance Checks





Offline

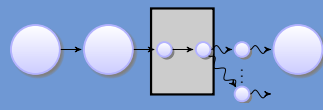
Semantical: $(\omega, \nu) \in [\![\text{ctrl}]\!]$

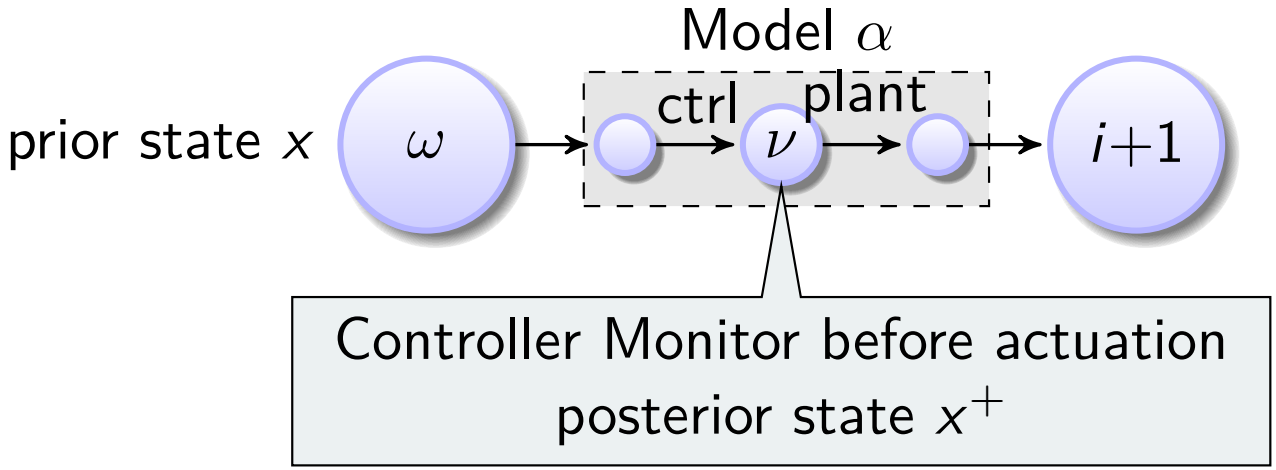
$\Updownarrow$ Theorem

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

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

Controller Monitor: Early Compliance Checks





Offline

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

$\Updownarrow$ Theorem

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

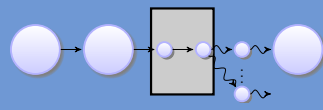
exists a run of ctrl to a state where $x = x^+$

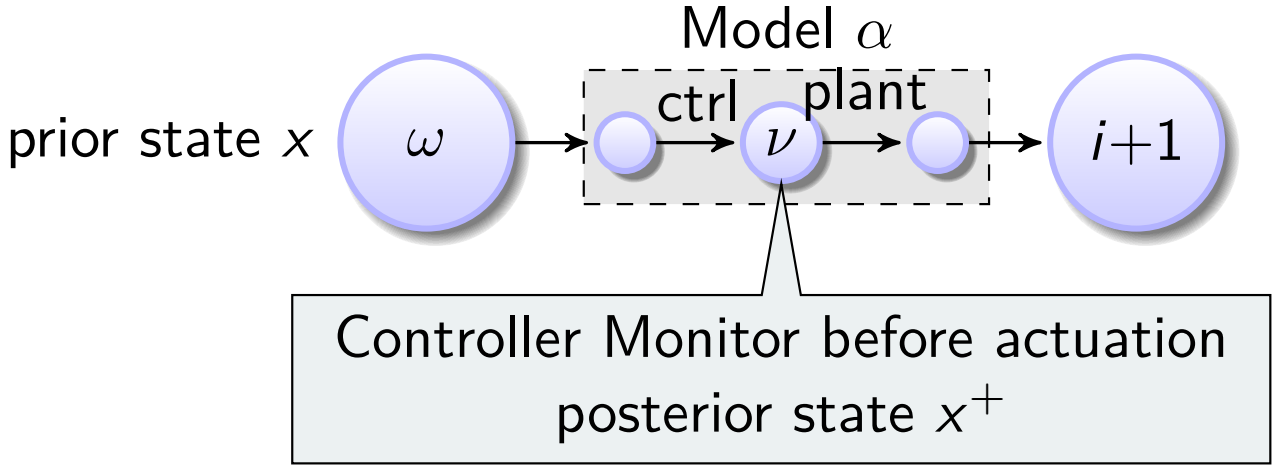
$\Uparrow$ d$\mathcal{L}$ proof

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

check at runtime (efficient)

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

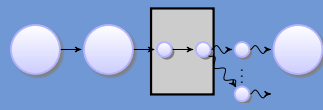
Arithmetical: $(\omega, \nu) \models F(x, x^+)$

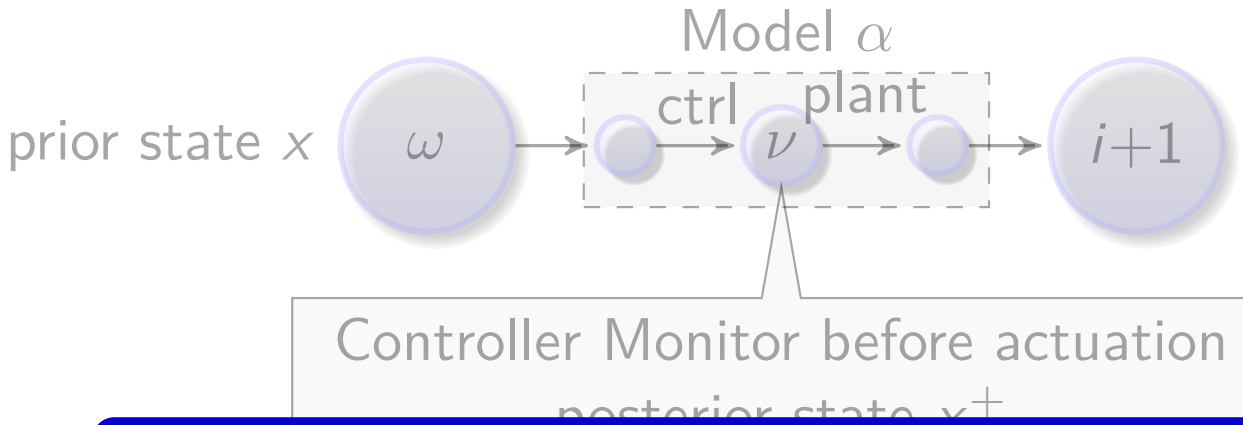
check at runtime (efficient)

Theorem (Controller Monitor Correctness) (FMSD'16)

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

Controller Monitor: Early Compliance Checks





Controller Monitor

Immediate detection of unsafe control before actuation

$\rightsquigarrow$ Safe execution of unverified implementations in perfect environments

Offline

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

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

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

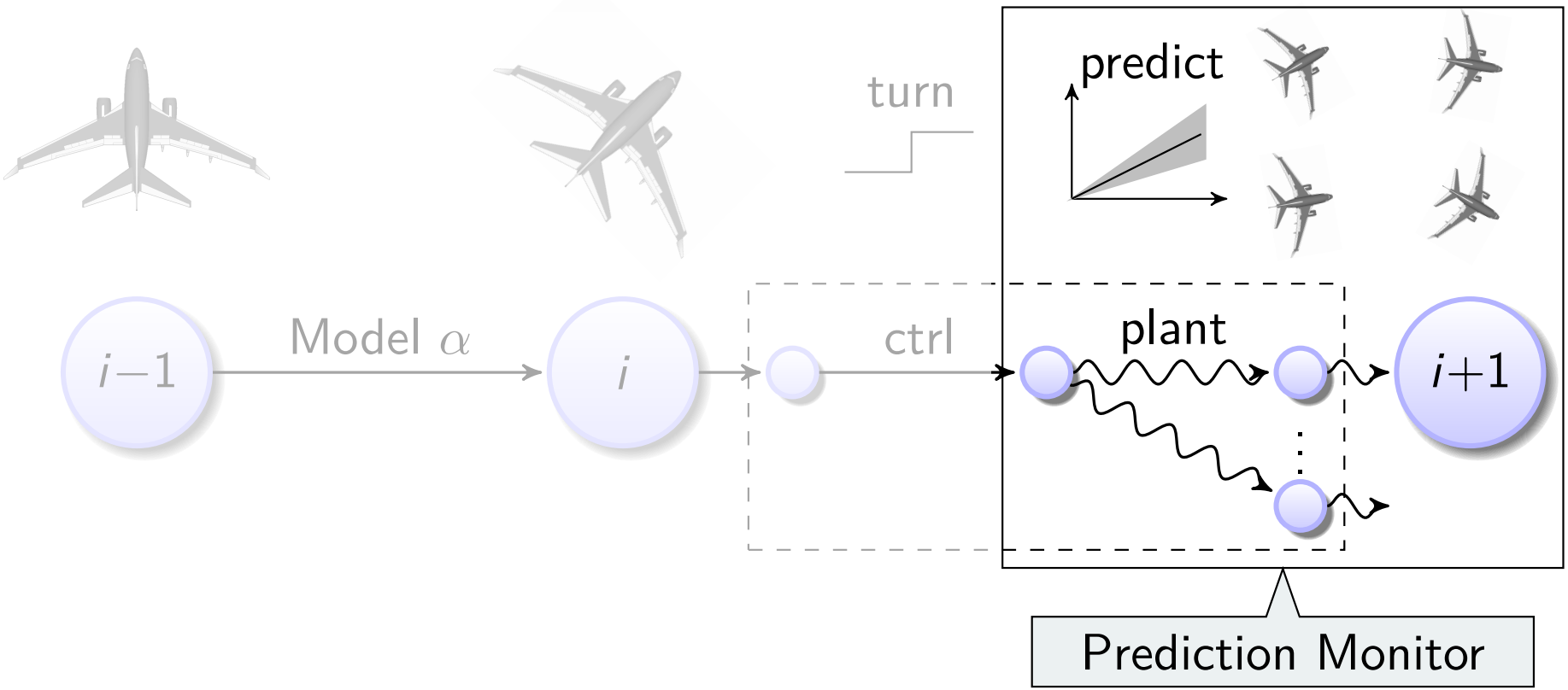
check at runtime (efficient)

Theorem (Controller Monitor Correctness) (FMSD'16)

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

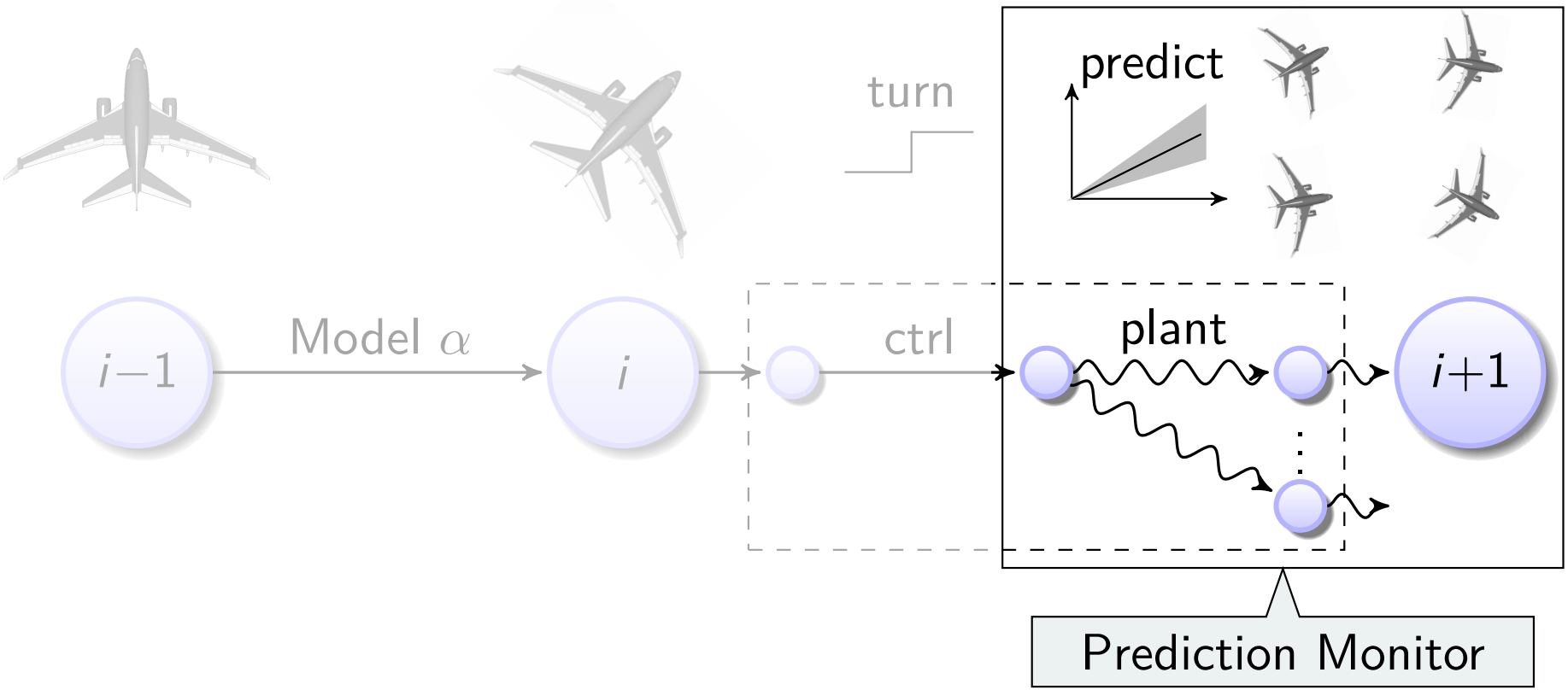
Outline

Safe despite evolution with disturbance?



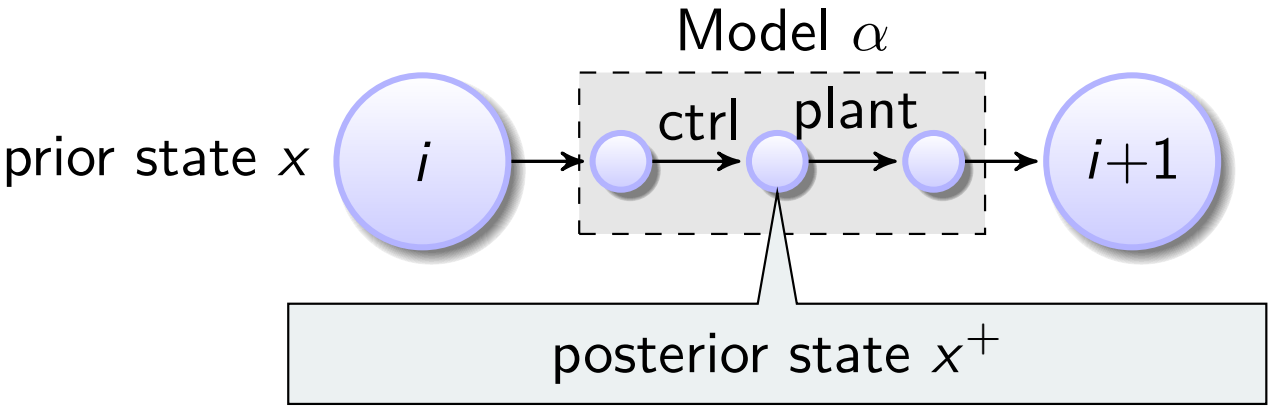
Outline

Safe despite evolution with disturbance?

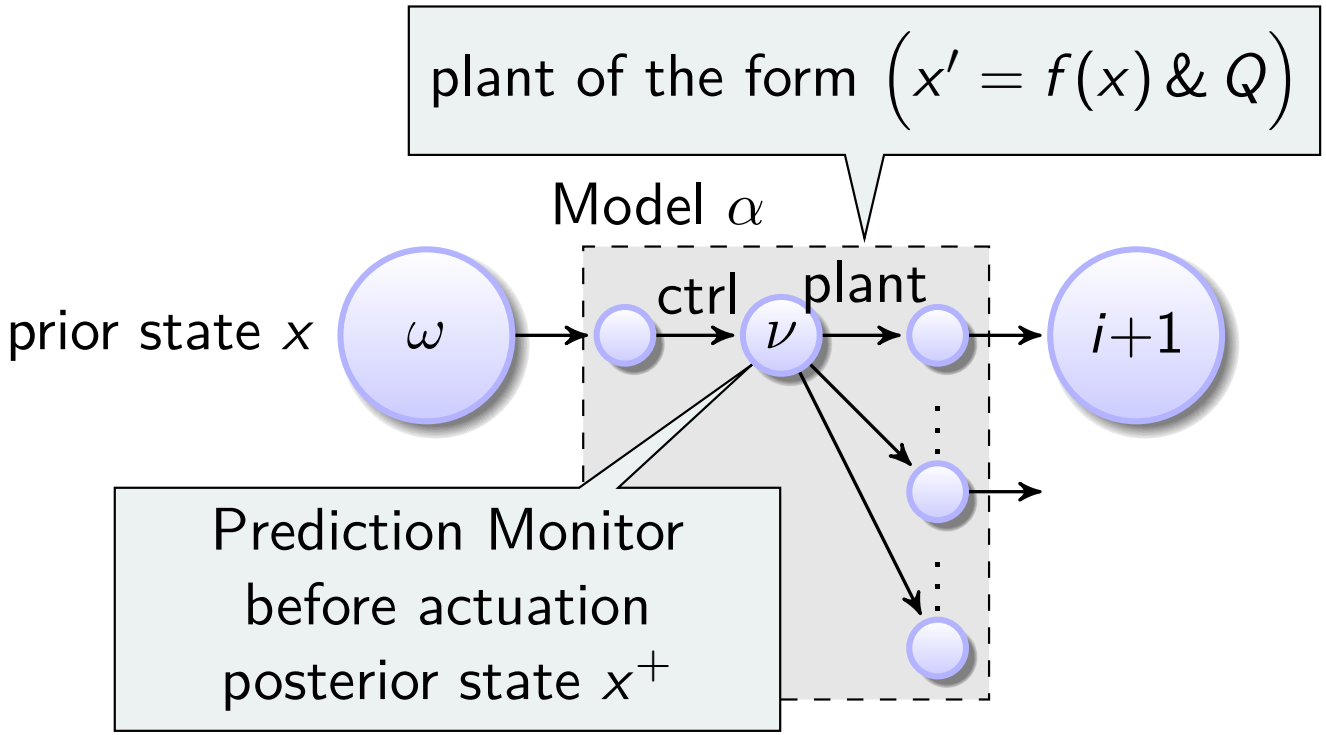


"Prediction is very difficult, especially if it's about the future." [Nils Bohr]

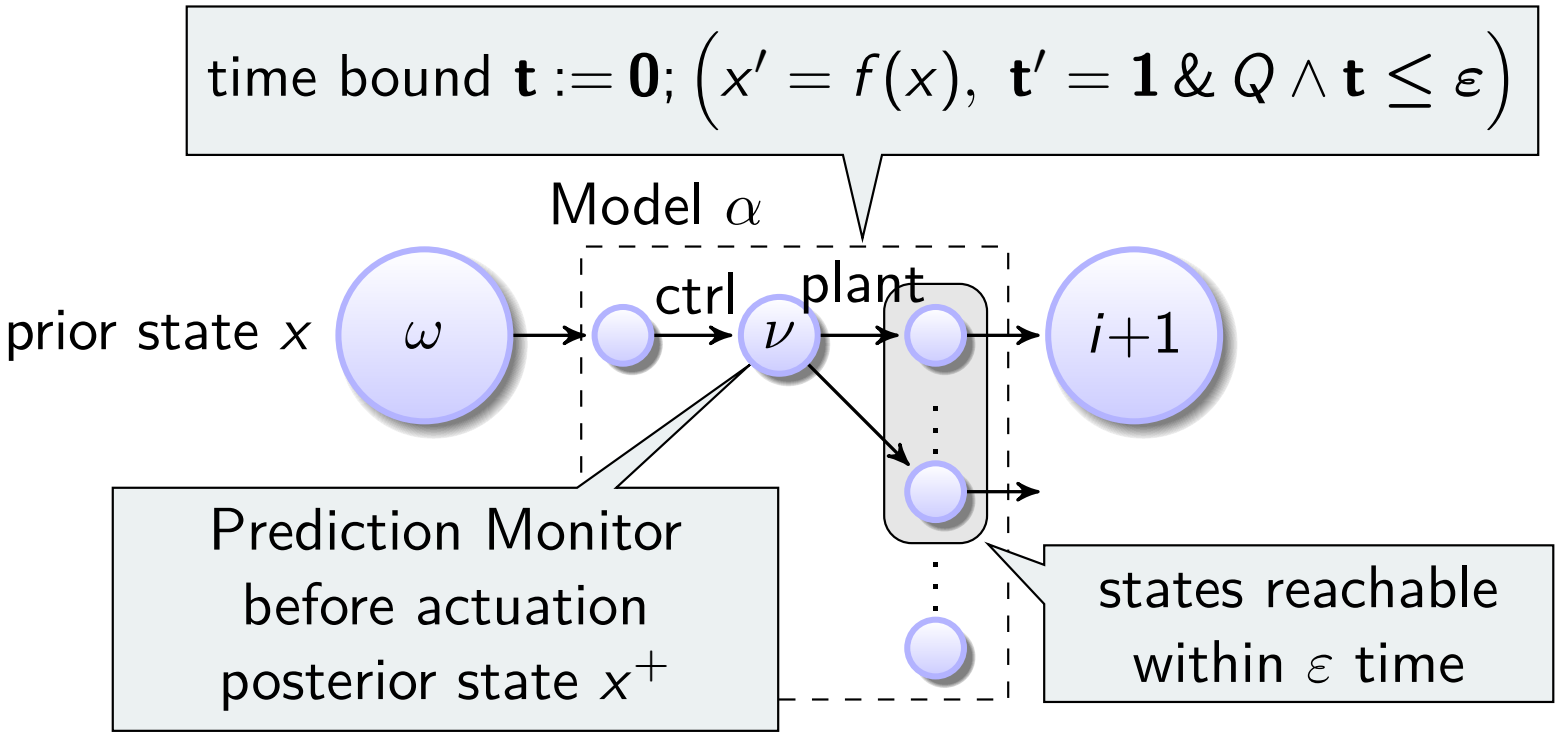
Prediction Monitor: Compliance with Disturbance



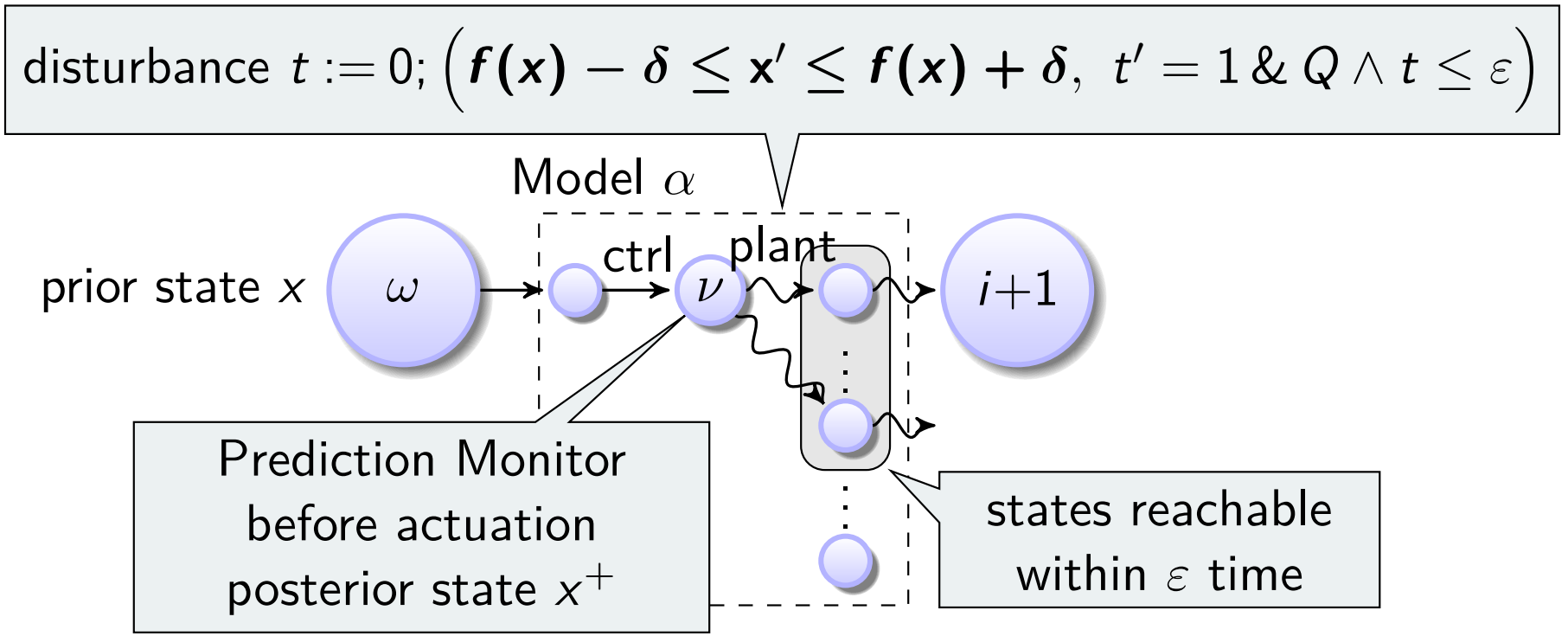
Prediction Monitor: Compliance with Disturbance



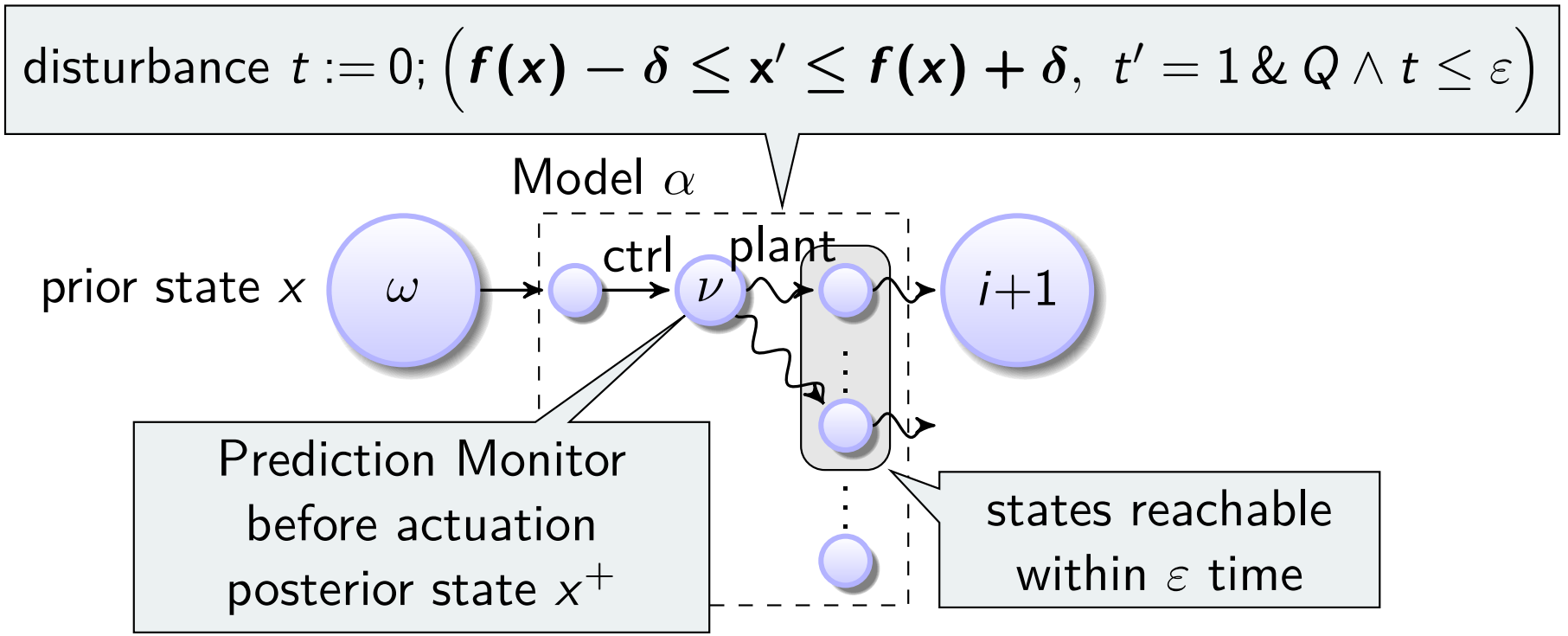
Prediction Monitor: Compliance with Disturbance



Prediction Monitor: Compliance with Disturbance



Prediction Monitor: Compliance with Disturbance



Offline

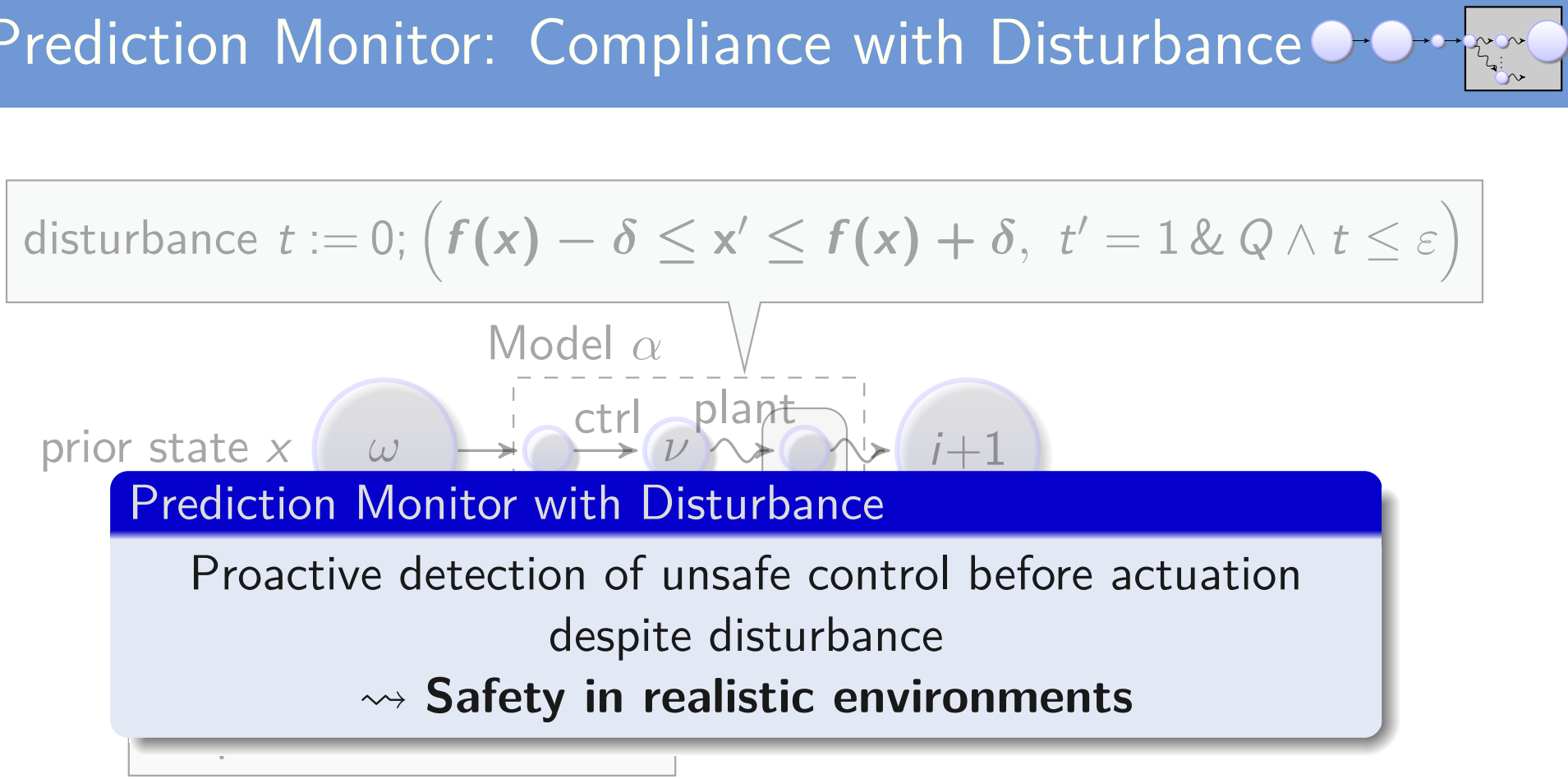
Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle \text{ctrl} \rangle (x = x^+ \wedge [\text{plant}]\varphi)$

$\Uparrow$ 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



Offline

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

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

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

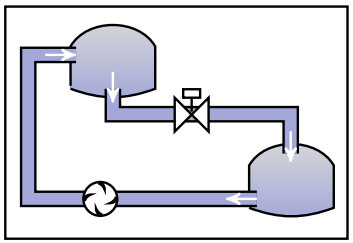
Invariant φ implies safety S
(known from safety proof)

Outline

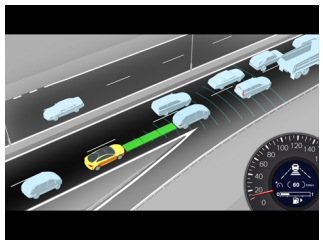
Evaluation

- Evaluated on hybrid system case studies

Water tank



Cruise control



© Volvo

Traffic control

© ASFINAG

Ground robots

© Black-I Robotics

Train control

© Harald Eisenberger

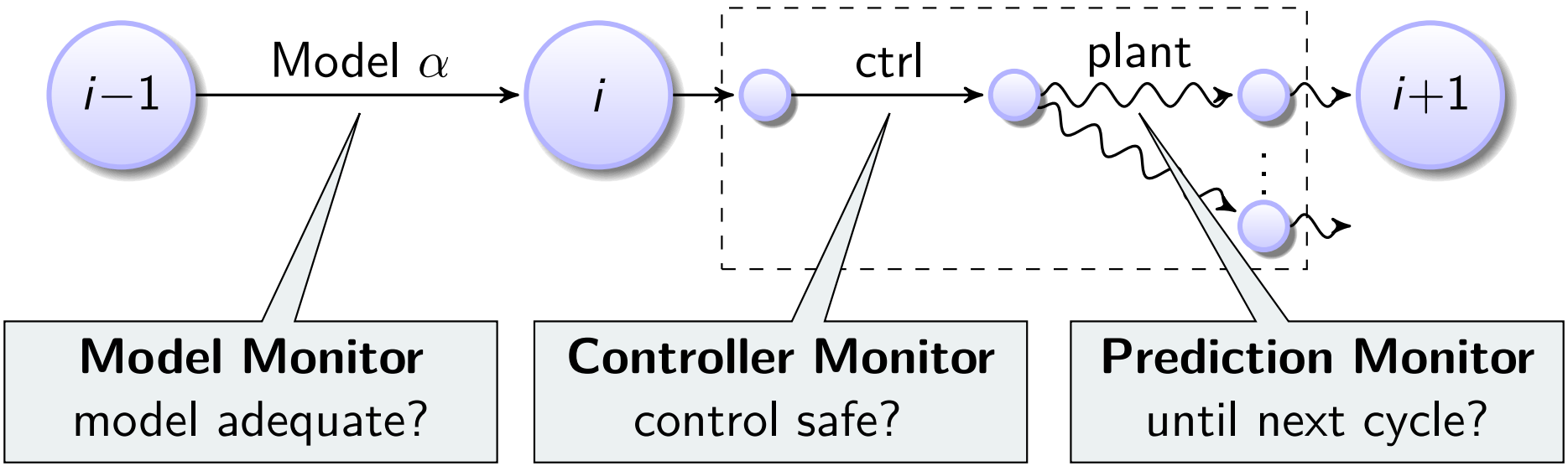
- Model sizes: 5–16 variables
- Monitor sizes: 20–150 operations
- Synthesis duration: 0.3–23 seconds (axiomatic) 6.2–211 (sequent)
- ModelPlex tactic produces correct-by-construction monitor in KeYmaera X
- Theorem: ModelPlex is decidable and monitor synthesis fully automated for controller monitor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

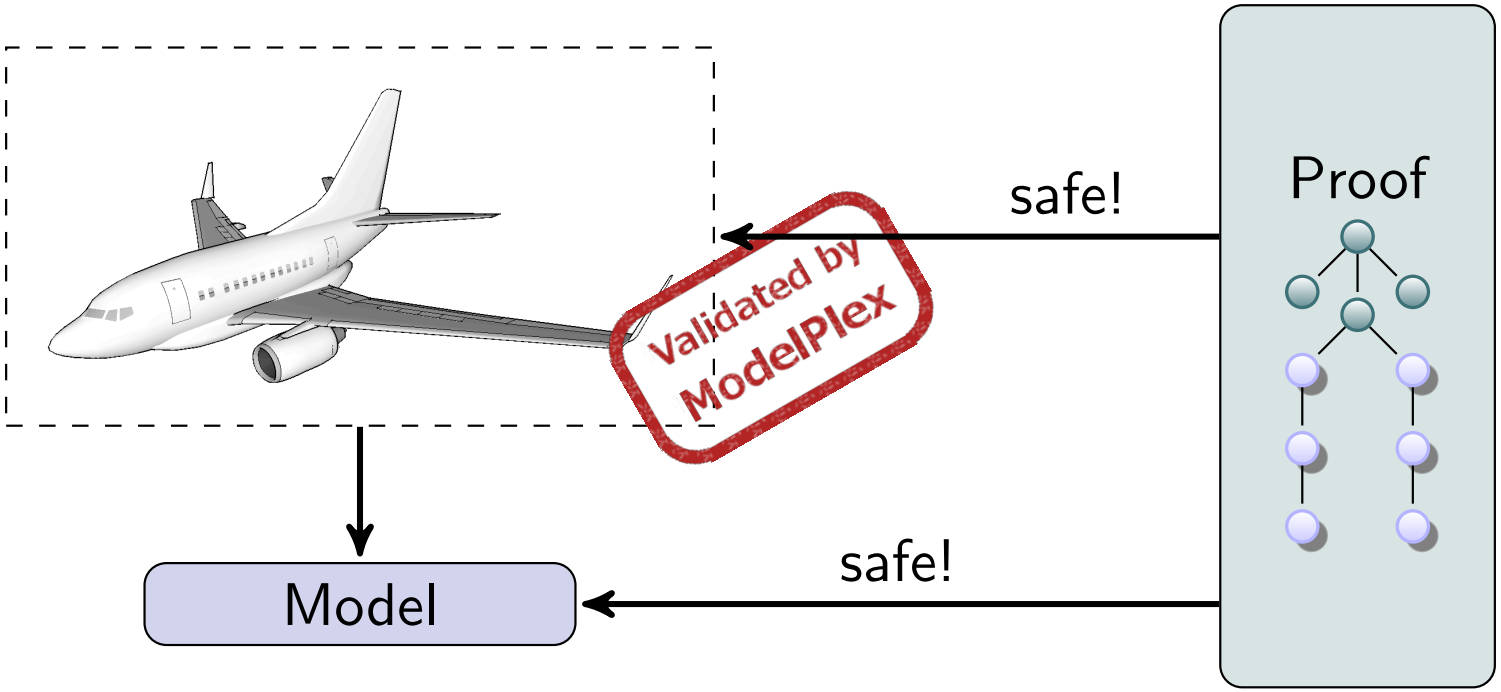
Outline

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.
Form. Methods Syst. Des., 49(1):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.
Logics of dynamical systems.
In *LICS*, pages 13–24. IEEE, 2012.
doi:10.1109/LICS.2012.13.


André Platzer.

A complete uniform substitution calculus for differential dynamic logic.

J. Autom. Reas., 2016.
doi:10.1007/s10817-016-9385-1.

Nathan Fulton and André Platzer.
A logic of proofs for differential dynamic logic: Toward independently checkable proof certificates for dynamic logics.
In Jeremy Avigad and Adam Chlipala, editors, *Proceedings of the 2016 Conference on Certified Programs and Proofs, CPP 2016, St. Petersburg, FL, USA, January 18-19, 2016*, pages 110–121. ACM, 2016.
doi:10.1145/2854065.2854078.

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.