ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models

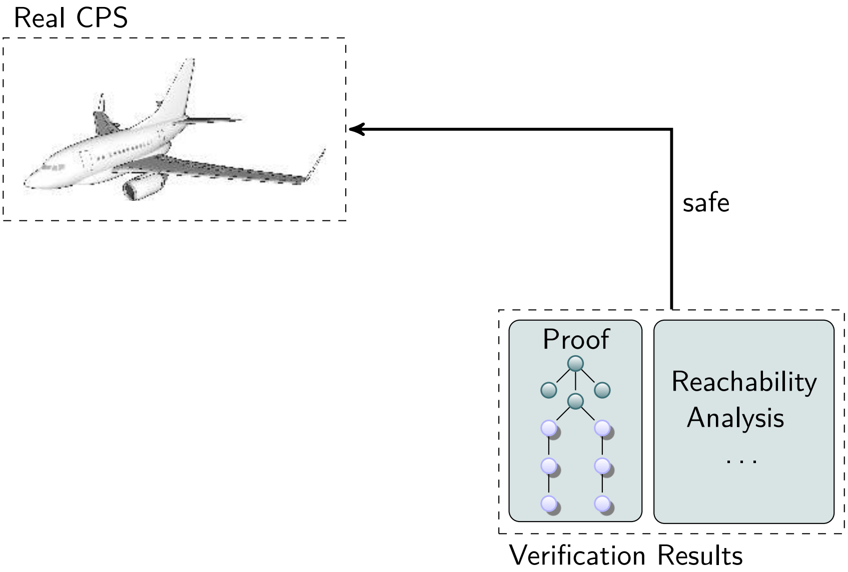
Stefan Mitsch André Platzer

Computer Science Department, Carnegie Mellon University

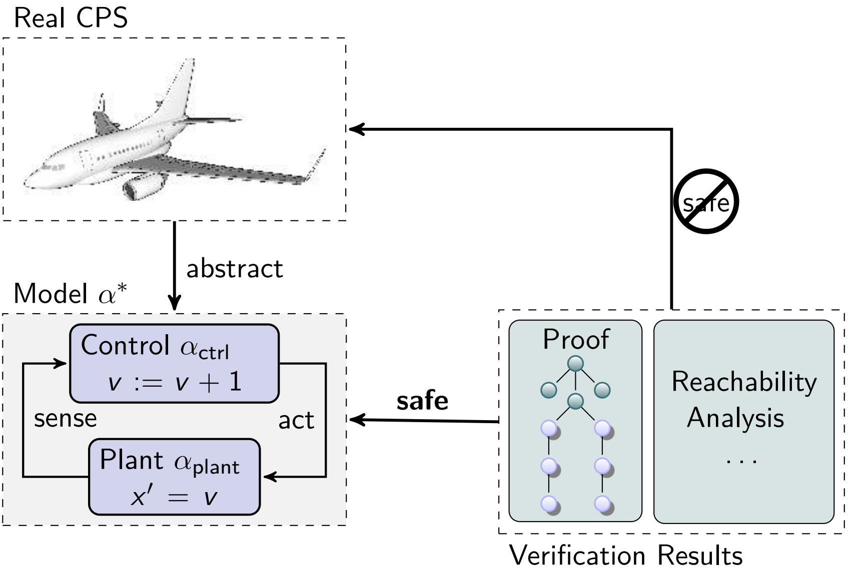
RV'14, Sept. 24, 2014

Simplex for Hybrid System Models

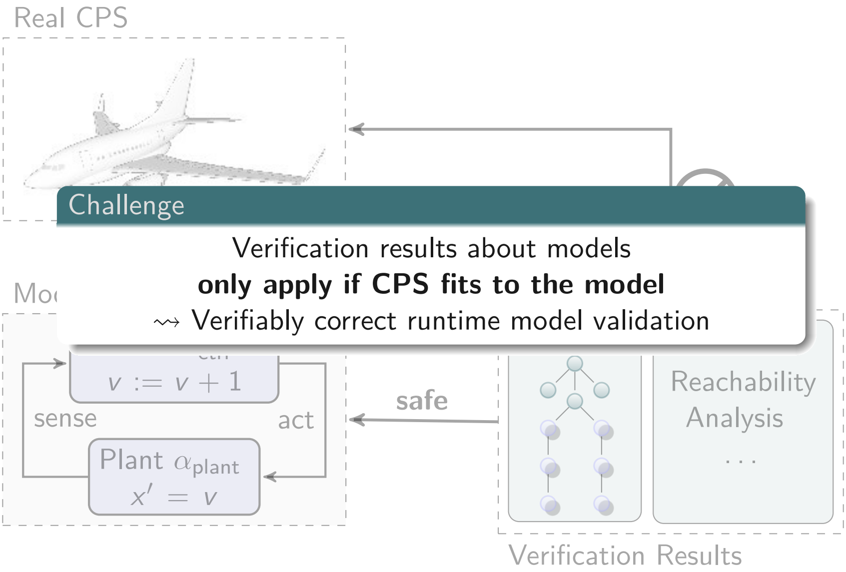
Formal Verification in CPS Development



Formal Verification in CPS Development



Formal Verification in CPS Development

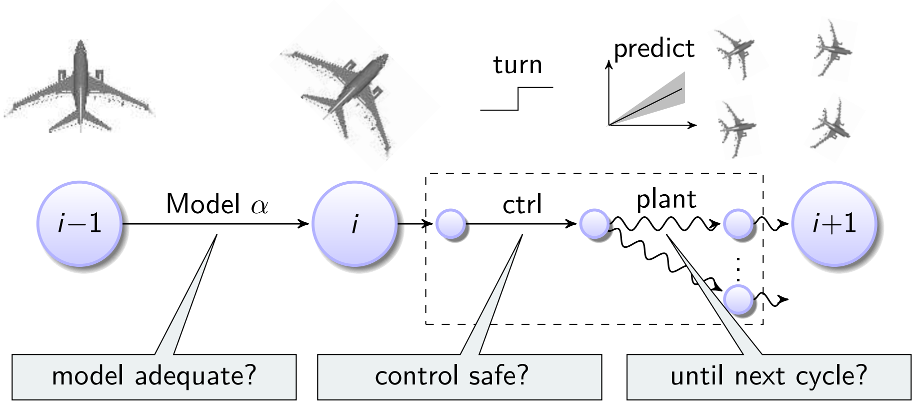


Challenge

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

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

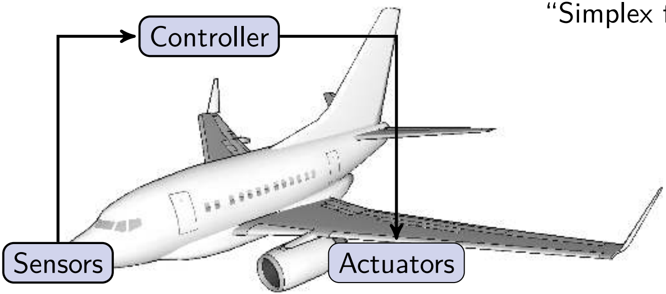
- Verification results transfer to CPS when validating model compliance
- Compliance with model is characterizable in logic
- Compliance formula transformed by proof to executable monitor

model adequate?

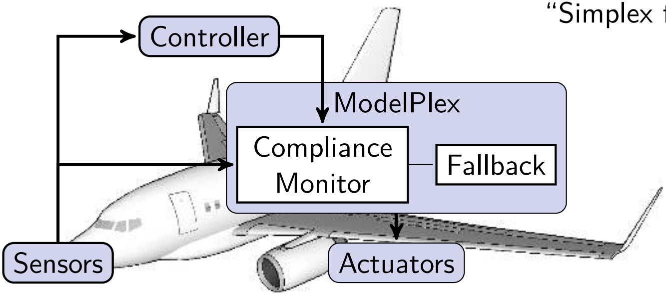
control safe?

until next cycle?

"Simplex for Models"



"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?

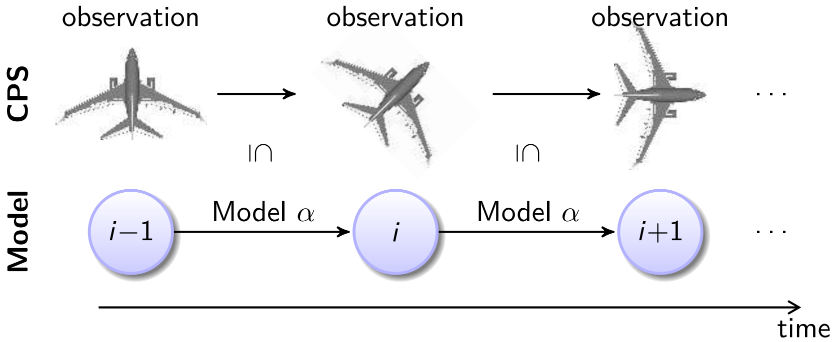
Fallback Safe action, executed when monitor is not satisfied

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

ModelPlex Approach

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

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

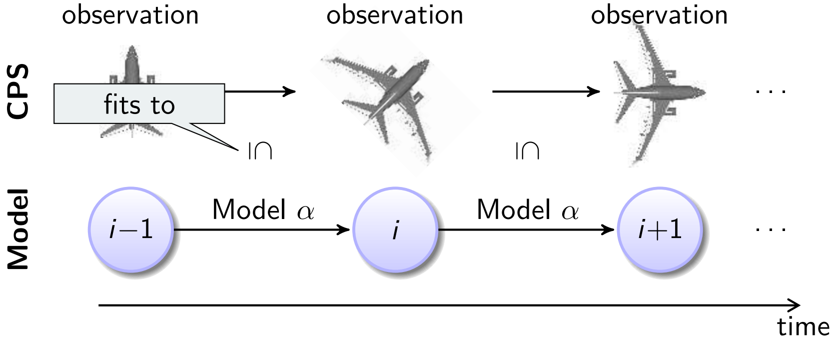


Detect non-compliance as soon as possible to initiate safe fallback actions

ModelPlex Approach

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

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

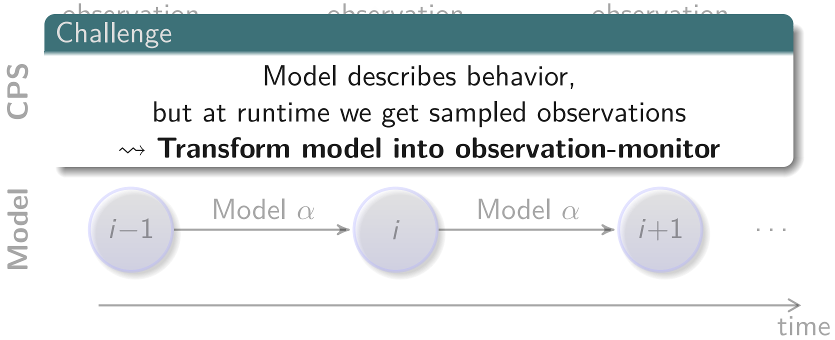


Detect non-compliance as soon as possible to initiate safe fallback actions

ModelPlex Approach

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

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

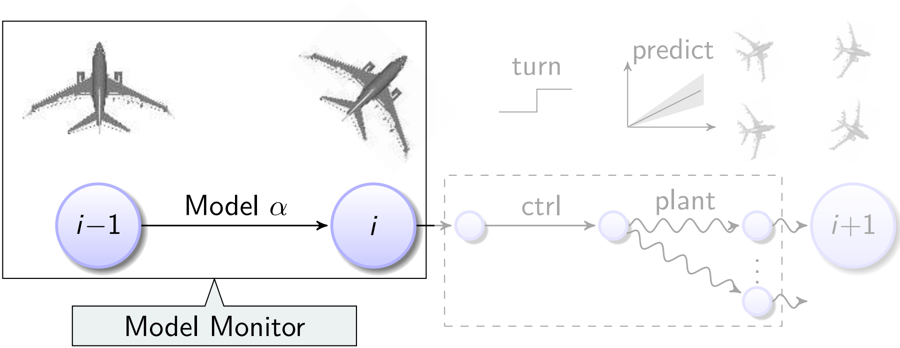


Challenge

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

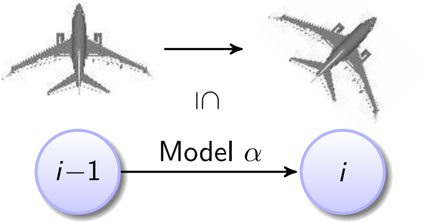
Detect non-compliance as soon as possible to initiate safe fallback actions

Outline



Monitor Characterization

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



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

$\longrightarrow$

$\mid\cap$

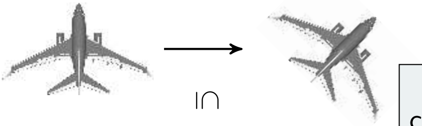
a prior state characterized by x^-

$i{-}1 \xrightarrow{\text{Model } \alpha} i$

a posterior state characterized by x^+

Semantical: $(x^-, x^+) \in \rho(\alpha)$ — reachability relation of α

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



a prior state characterized by x^-

$\cap\!|$

$(i{-}1) \xrightarrow{\text{Model } \alpha} (i)$

a posterior state characterized by x^+

Offline

Semantical: $(x^-, x^+) \in \rho(\alpha)$

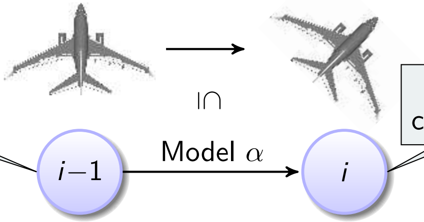
$\Updownarrow$ Theorem

Logic (d$\mathcal{L}$): $(x = x^-) \rightarrow \langle \alpha_{(x)} \rangle (x = x^+)$

starting at $x = x^-$ 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: $(x^-, x^+) \in \rho(\alpha)$

$\Updownarrow$ Theorem

Logic (d$\mathcal{L}$): $(x = x^-) \rightarrow \langle \alpha_{(x)} \rangle (x = x^+)$

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

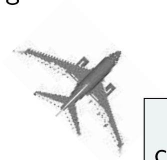
$\Updownarrow$ d$\mathcal{L}$ proof

Real arithmetic: $F(x^-, x^+)$

check at runtime (efficient)

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

$\longrightarrow$



$\rotatebox{90}{$\subseteq$}$

a prior state characterized by x^-

$i{-}1 \xrightarrow{\text{Model } \alpha} i$

a posterior state characterized by x^+

Offline

Semantical: $(x^-, x^+) \in \rho(\alpha)$

$\Updownarrow$ Theorem

Logic (d$\mathcal{L}$): $(x = x^-) \rightarrow \langle \alpha_{(x)} \rangle (x = x^+)$

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

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

Real arithmetic: $F(x^-, x^+)$

check at runtime (efficient)

Provably Correct Synthesis of Monitors

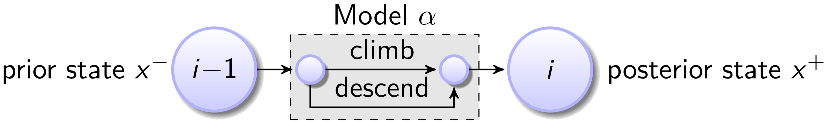
- Proof calculus of d$\mathcal{L}$ executes models symbolically

prior state x^- $\quad (i{-}1) \xrightarrow{\text{Model } \alpha} (i) \quad$ posterior state x^+

proof attempt

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

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



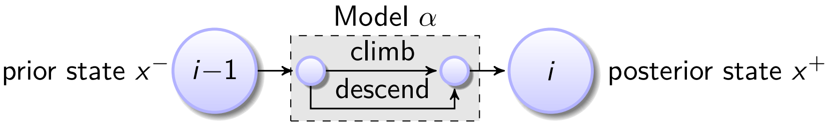
proof attempt

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

$$\langle\cup\rangle \frac{\langle \text{climb} \rangle \phi \vee \langle \text{descend} \rangle \phi}{\langle \text{climb} \cup \text{descend} \rangle \phi}$$

Provably Correct Synthesis of Monitors

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



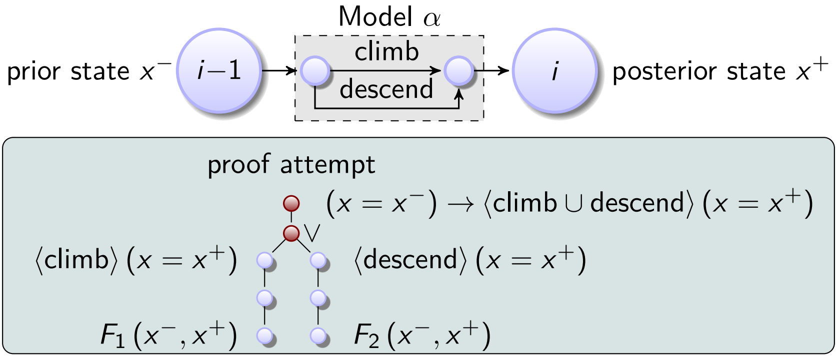
proof attempt

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

$\vee$

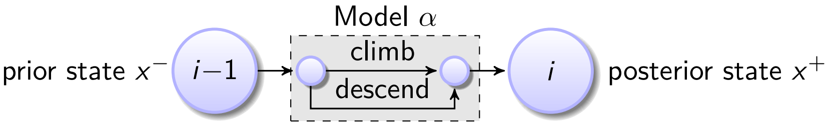
$\langle \text{climb} \rangle \, (x = x^+)$ $\qquad$ $\langle \text{descend} \rangle \, (x = x^+)$

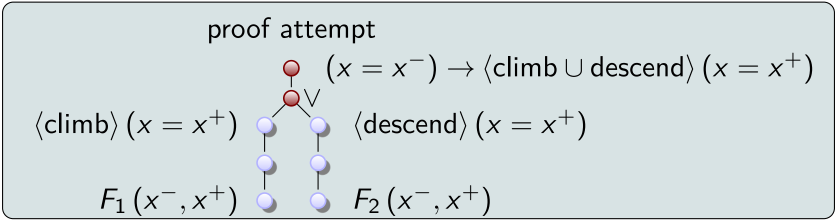
- Proof calculus of d$\mathcal{L}$ executes models symbolically



Provably Correct Synthesis of Monitors

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



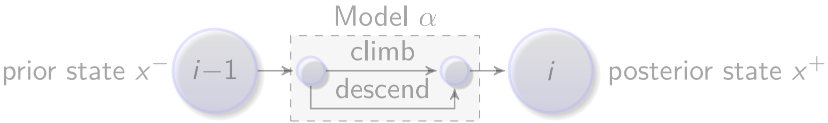


Monitor: $\overbrace{F_1(x^-, x^+) \vee F_2(x^-, x^+)}$

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

Provably Correct Synthesis of Monitors

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



Model Monitor

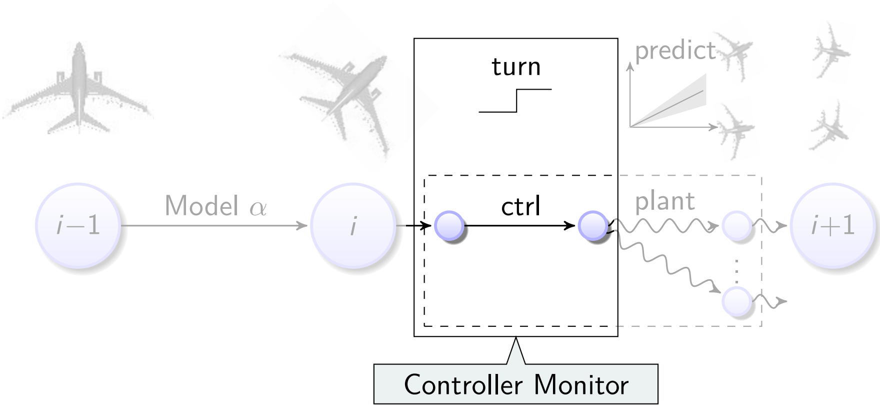
Immediate detection of model violation
$\rightsquigarrow$ Mitigates safety issues with safe fallback action

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

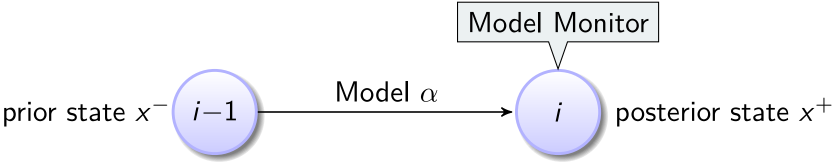
Monitor: $\underbrace{F_1(x^-, x^+) \vee F_2(x^-, x^+)}$

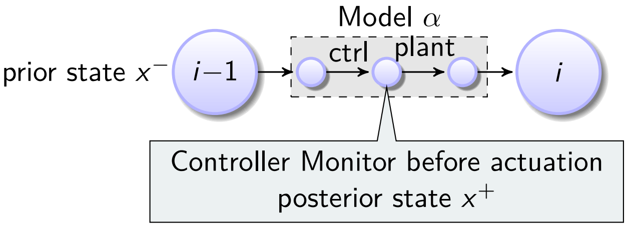
- The subgoals that cannot be proved express all the conditions on the relations of variables imposed by the model $\rightsquigarrow$ execute at runtime

For typical models ctrl; plant we can check earlier



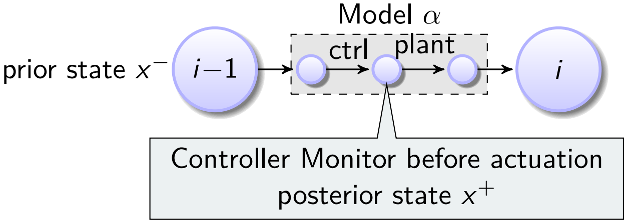
Early Compliance Checks for Controllers





Semantical: $(x^-, x^+) \in \rho(\text{ctrl})$ reachability relation of ctrl

Early Compliance Checks for Controllers



Offline

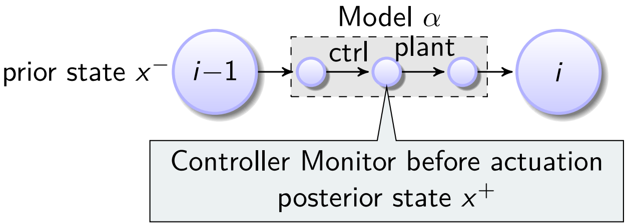
Semantical: $(x^-, x^+) \in \rho(\text{ctrl})$

$\Updownarrow$ Theorem

Logic ($\mathsf{d}\mathcal{L}$): $(x = x^-) \rightarrow \langle \text{ctrl}_{(x)} \rangle\, (x = x^+)$

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

Early Compliance Checks for Controllers



Offline

Semantical: $(x^-, x^+) \in \rho(\text{ctrl})$

$\Updownarrow$ Theorem

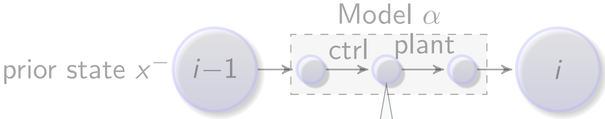
Logic ($\mathsf{d}\mathcal{L}$): $(x = x^-) \rightarrow \langle \text{ctrl}_{(x)} \rangle \, (x = x^+)$

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

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

Real arithmetic: $F\,(x^-, x^+)$

Early Compliance Checks for Controllers



Controller Monitor

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

Offlin

Semantical: $(x^-, x^+) \in \rho(\text{ctrl})$

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

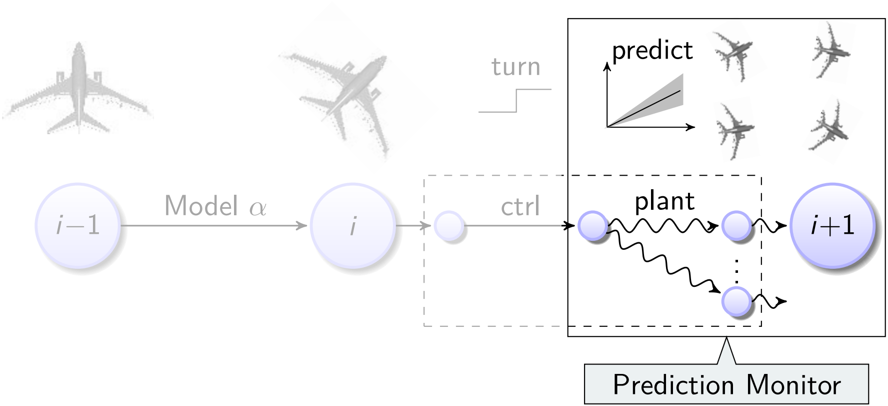
$\Updownarrow$ Theorem

Logic ($\mathsf{d}\mathcal{L}$):$(x = x^-) \rightarrow \langle \text{ctrl}_{(x)} \rangle (x = x^+)$

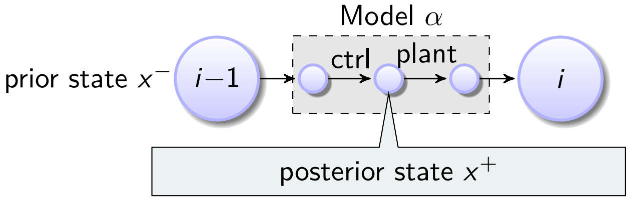
$\Uparrow$ $\mathsf{d}\mathcal{L}$ proof

Real arithmetic: $F(x^-, x^+)$

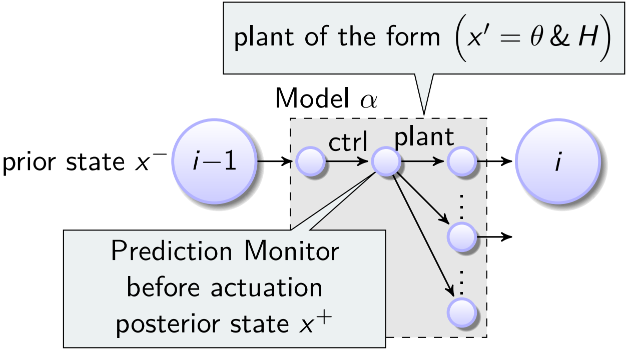
Safe despite evolution with disturbance?

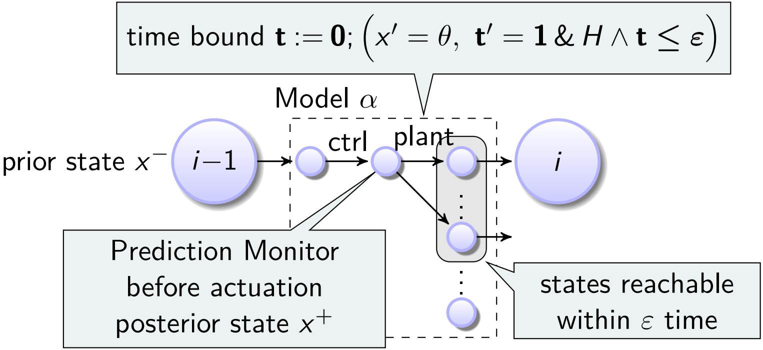


Compliance Checks despite Disturbance

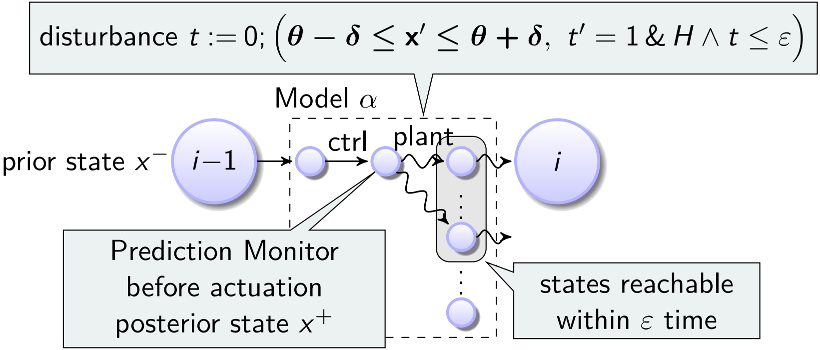


Compliance Checks despite Disturbance

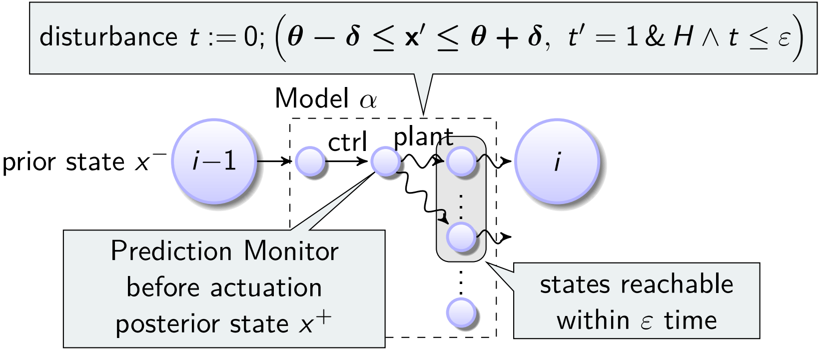



time bound $\mathbf{t} := \mathbf{0}; \left(x' = \theta,\ \mathbf{t}' = \mathbf{1} \,\&\, H \wedge \mathbf{t} \leq \varepsilon\right)$
Model α
prior state x^-
$i-1$
ctrl
plant
i
Prediction Monitor
before actuation
posterior state x^+
states reachable
within ε time

Compliance Checks despite Disturbance



Compliance Checks despite Disturbance



Offline

Logic (d$\mathcal{L}$): $(x = x^-) \rightarrow \langle \mathrm{ctrl}_{(x)} \rangle \left(x = x^+ \wedge [\mathrm{plant}_{(x)}]\varphi\right)$

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

Real arithmetic: $F(x^-, x^+)$

Invariant state φ implies safety
(known from safety proof)

disturbance $t := 0; \left(\theta - \delta \le x' \le \theta + \delta,\ t' = 1 \,\&\, H \wedge t \le \varepsilon\right)$

Model α

prior state x^- $i{-}1$ ctrl plant i

Prediction Monitor with Disturbance

Proactive detection of unsafe control before actuation despite disturbance

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

Offline

Logic (d$\mathcal{L}$): $(x = x^-) \rightarrow \langle \mathrm{ctrl}_{(x)} \rangle \left(x = x^+ \wedge [\mathrm{plant}_{(x)}]\varphi\right)$

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

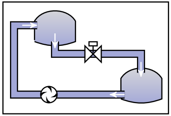
Real arithmetic: $F(x^-, x^+)$

Invariant state φ implies safety (known from safety proof)

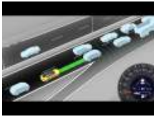
Evaluation

- Evaluated on hybrid system case studies

Water tank



Cruise control



© Volvo

Traffic control

© ASFINAG

Ground robots

© Black-I Robotics

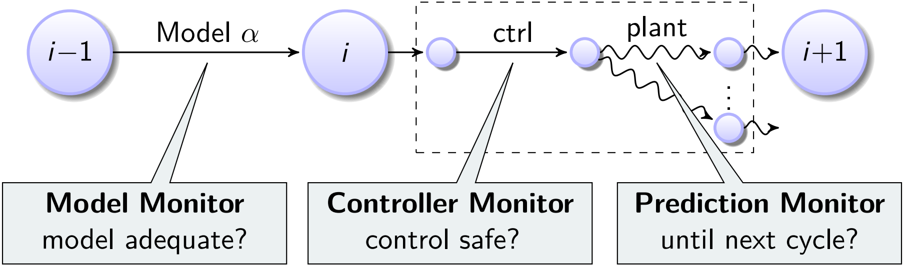
Train control

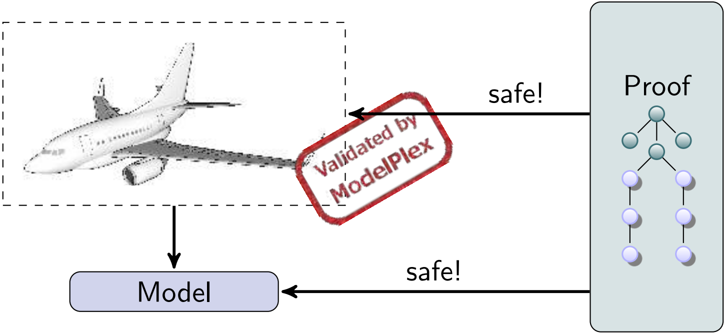
© Harald Eisenberger

- Model sizes: 5–16 variables
- Monitor sizes: 20–150 operations
  - with automated simplification to remove redundant checks
  - improvement potential: simplification for any monitor
- Theorem: ModelPlex is decidable and monitor synthesis fully automated in important classes

ModelPlex ensures that proofs apply to real CPS

- Validate model compliance
- Characterize compliance with model in logic
- Prover transforms compliance formula to executable monitor





Stefan Mitsch
smitsch@cs.cmu.edu
www.cs.cmu.edu/∼smitsch

Theorems

- State Recall (Online Monitoring)
- Model Monitor Correctness
- Controller Monitor Correctness
- Prediction Monitor Correctness
- Decidability and Computability

State Recall

V set of variables whose state we want to recall

$\Upsilon_V^- \equiv \bigwedge_{x\in V} x = x^-$ characterizes a state prior to a run of α (fresh variables x^- occur solely in Υ_V^- and recall this state)

$\Upsilon_V^+ \equiv \bigwedge_{x\in V} x = x^+$ characterizes the posterior states (fresh x^+)

Programs hybrid program α, α^* repeats α arbitrarily many times

Assume all consecutive pairs of states $(\nu_{i-1}, \nu_i) \in \rho(\alpha)$ of $n \in \mathbb{N}^+$ executions, whose valuations are recalled with $\Upsilon_V^i \equiv \bigwedge_{x\in V} x = x^i$ and Υ_V^{i-1} are plausible w.r.t. the model α, i. e., $\models \bigwedge_{1\leq i\leq n} \left(\Upsilon_V^{i-1} \rightarrow \langle\alpha\rangle\Upsilon_V^i\right)$ with $\Upsilon_V^- = \Upsilon_V^0$ and $\Upsilon_V^+ = \Upsilon_V^n$.

Then the sequence of states originates from an α^* execution from Υ_V^0 to Υ_V^n, i. e., $\models \Upsilon_V^- \rightarrow \langle\alpha^*\rangle\Upsilon_V^+$.

Model Monitor Correctness

$\models \phi \to [\alpha^*]\psi$ α^* is provably safe

Definitions Let $V_m = BV(\alpha) \cup FV(\psi)$; let $\nu_0, \nu_1, \nu_2, \nu_3 \ldots \in \mathbb{R}^n$ be a sequence of states, with $\nu_0 \models \phi$ and that agree on $\Sigma \setminus V_m$, i. e., $\nu_0|_{\Sigma \setminus V_m} = \nu_k|_{\Sigma \setminus V_m}$ for all k.

Model Monitor $(\nu, \nu_{i+1}) \models \chi_\mathrm{m}$ as χ_m evaluated in the state resulting from ν by interpreting x^+ as $\nu_{i+1}(x)$ for all $x \in V_m$, i. e., $\nu_{x^+}^{\nu_{i+1}(x)} \models \chi_\mathrm{m}$

Correctness If $(\nu_i, \nu_{i+1}) \models \chi_\mathrm{m}$ for all $i < n$ then we have $\nu_n \models \psi$ where

$$\chi_\mathrm{m} \equiv \left(\phi|_\mathrm{const} \to \langle\alpha\rangle \Upsilon^+_{V_m}\right)$$

and $\phi|_\mathrm{const}$ denotes the conditions of ϕ that involve only constants that do not change in α, i. e., $FV(\phi|_\mathrm{const}) \cap BV(\alpha) = \emptyset$.

Controller Monitor Correctness

$\models \phi \to [\alpha^*]\psi$ α^* is provably safe with invariant φ

Definitions Let α of the canonical form $\alpha_{\mathsf{ctrl}}; \alpha_{\mathsf{plant}}$; let $\nu \models \phi|_{\mathsf{const}} \land \varphi$, as checked by χ_{m}; let $\tilde{\nu}$ be a post-controller state.

Controller Monitor $(\nu, \tilde{\nu}) \models \chi_{\mathsf{c}}$ as χ_{c} evaluated in the state resulting from ν by interpreting x^+ as $\tilde{\nu}(x)$ for all $x \in V_c$, i. e., $\nu_{x^+}^{\tilde{\nu}(x)} \models \chi_{\mathsf{c}}$

Correctness If $(\nu, \tilde{\nu}) \models \chi_{\mathsf{c}}$ where

$$\chi_{\mathsf{c}} \equiv \phi|_{\mathsf{const}} \to \langle \alpha_{\mathsf{ctrl}} \rangle \Upsilon_{V_c}^{+}$$

then we have that $(\nu, \tilde{\nu}) \in \rho(\alpha_{\mathsf{ctrl}})$ and $\tilde{\nu} \models \varphi$.

Prediction Monitor Correctness

$\models \phi \rightarrow [\alpha^*]\psi$ α^* is provably safe with invariant φ

Definitions Let $V_p = BV(\alpha) \cup FV([\alpha]\varphi)$. Let $\nu \models \phi|_{\text{const}} \wedge \varphi$, as checked by χ_{m}. Further assume $\tilde{\nu}$ such that $(\nu, \tilde{\nu}) \in \rho(\alpha_{\text{ctrl}})$, as checked by χ_{c}.

Prediction Monitor $(\nu, \tilde{\nu}) \models \chi_{\text{p}}$ as χ_{p} evaluated in the state resulting from ν by interpreting x^+ as $\tilde{\nu}(x)$ for all $x \in V_p$, i. e., $\nu_{x^+}^{\tilde{\nu}(x)} \models \chi_{\text{p}}$

Correctness If $(\nu, \tilde{\nu}) \models \chi_{\text{p}}$ where

$$\chi_{\text{p}} \equiv (\phi|_{\text{const}} \wedge \varphi) \rightarrow \langle \alpha_{\text{ctrl}} \rangle (\Upsilon_{V_p}^{+} \wedge [\alpha_{\delta\text{plant}}]\varphi)$$

then we have for all $(\tilde{\nu}, \omega) \in \rho(\alpha_{\delta\text{plant}})$ that $\omega \models \varphi$

Decidability and Computability

Assumptions
- canonical models $\alpha \equiv \alpha_{\text{ctrl}}; \alpha_{\text{plant}}$ without nested loops
- with solvable differential equations in α_{plant}
- disturbed plants $\alpha_{\delta\text{plant}}$ with constant additive disturbance δ

Decidability Monitor correctness is decidable, i. e., the formulas

- $\chi_{\text{m}} \rightarrow \langle\alpha\rangle\Upsilon_V^+$
- $\chi_{\text{c}} \rightarrow \langle\alpha_{\text{ctrl}}\rangle\Upsilon_V^+$
- $\chi_{\text{p}} \rightarrow \langle\alpha\rangle(\Upsilon_V^+ \wedge [\alpha_{\delta\text{plant}}]\phi)$

are decidable

Computability Monitor synthesis is computable, i. e., the functions

- $\text{synth}_m : \langle\alpha\rangle\Upsilon_V^+ \mapsto \chi_{\text{m}}$
- $\text{synth}_c : \langle\alpha_{\text{ctrl}}\rangle\Upsilon_V^+ \mapsto \chi_{\text{c}}$
- $\text{synth}_p : \langle\alpha\rangle(\Upsilon_V^+ \wedge [\alpha_{\delta\text{plant}}]\phi) \mapsto \chi_{\text{p}}$

are computable

Variables

x	current level	ε	control cycle
m	maximum level	f	flow

Model and Safety Property

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

Model Monitor Specification Conjecture

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

Water Tank Example: Nondeterministic Assignment

Proof Rules

$$(\langle * \rangle)\frac{\exists X \langle x := X\rangle \phi}{\langle x := *\rangle \phi}\ ^1 \qquad (\exists r)\frac{\Gamma \vdash \phi(\theta), \exists x\, \phi(x), \Delta}{\Gamma \vdash \exists x\, \phi(x), \Delta}\ ^2 \qquad (Wr)\frac{\Gamma \vdash \Delta}{\Gamma \vdash \phi, \Delta}$$

[1] X is a new logical variable

[2] θ is an arbitrary term, often a new (existential) logical variable X.

Sequent Deduction

w/o Opt. 1

$$\cfrac{\cfrac{\phi \vdash \langle f := F\rangle \langle ?{-1} \le f \le \frac{m-x}{\varepsilon}\rangle \langle plant\rangle \Upsilon^+}{\phi \vdash \exists F \langle f := F\rangle \langle ?{-1} \le f \le \frac{m-x}{\varepsilon}\rangle \langle plant\rangle \Upsilon^+}\ \exists r, Wr}{\phi \vdash \langle f := *; ?{-1} \le f \le \frac{m-x}{\varepsilon}\rangle \langle plant\rangle \Upsilon^+}\ \langle * \rangle$$

$$\cfrac{\phi \vdash \langle f := f^+\rangle \langle ?{-1} \le f \le \frac{m-x}{\varepsilon}\rangle \langle plant\rangle \Upsilon^+}{\ldots}\ \exists r, Wr$$

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

Water Tank Example: Differential Equations

Proof Rules

$$(\langle ' \rangle)\frac{\exists T{\geq}0\ ((\forall 0{\leq}\tilde{t}{\leq}T\ \langle x := y(\tilde{t})\rangle H) \wedge \langle x := y(T)\rangle\phi)}{\langle x' = \theta\ \&\ H\rangle\phi}\ ^{1} \qquad (\mathrm{QE})\frac{\mathrm{QE}(\phi)}{\phi}\ ^{2}$$

[1] T and $\tilde{t}$ are fresh logical variables and $\langle x := y(T)\rangle$ is the discrete assignment belonging to the solution y of the differential equation with constant symbol x as symbolic initial value

[2] iff $\phi \equiv \mathrm{QE}(\phi)$, ϕ is a first-order real arithmetic formula, $\mathrm{QE}(\phi)$ is an equivalent quantifier-free formula

Sequent Deduction

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

Evaluation

	Case Study	Model dim.	Model proof size (branches)	Monitor dim.	Monitor steps (open seq.) w/ Opt. 1	Monitor steps (open seq.) auto	Monitor proof steps (branches)	Monitor size
χ_m	Water tank	5	38 (4)	3	16 (2)	20 (2)	64 (5)	32
	Cruise control	11	969 (124)	7	127 (13)	597 (21)	19514 (1058)	1111
	Speed limit	9	410 (30)	6	487 (32)	5016 (126)	64311 (2294)	19850
χ_c	Water tank	5	38 (4)	1	12 (2)	14 (2)	40 (3)	20
	Cruise control	11	969 (124)	7	83 (13)	518 (106)	5840 (676)	84
	Ground robot	14	3350 (225)	11	94 (10)	1210 (196)	26166 (2854)	121
	ETCS safety	16	193 (10)	13	162 (13)	359 (37)	16770 (869)	153
χ_p	Water tank	8	80 (6)	1	135 (4)	N/A	307 (12)	43

- Theorem: ModelPlex is decidable and monitor synthesis can be automated in important classes

Algorithm 1: ModelPlex monitor synthesis

input : A hybrid program α, a set of variables $\mathcal{V} \subseteq BV(\alpha)$, an initial condition ϕ such that $\models \phi \rightarrow [\alpha^*]\psi$.

output: A monitor χ_m such that $\models \chi_\mathsf{m} \equiv \phi|_\mathsf{const} \rightarrow \langle\alpha\rangle\Upsilon^+$.

begin

$S \longleftarrow \emptyset$

$\Upsilon^+ \longleftarrow \bigwedge_{x \in \mathcal{V}} x = x^+$ with fresh variables x_i^+ `// Monitor conjecture`

$G \longleftarrow \{\vdash \phi|_\mathsf{const} \rightarrow \langle\alpha\rangle\Upsilon^+\}$

while $G \neq \emptyset$ **do** `// Analyze monitor conjecture`

  foreach $g \in G$ **do**

    $G \longleftarrow G - \{g\}$

    if g *is first-order* **then**

      if $\not\models g$ **then** $S \longleftarrow S \cup \{g\}$

    else

      $\tilde{g} \longleftarrow$ apply d$\mathcal{L}$ proof rule to g

      $G \longleftarrow G \cup \{\tilde{g}\}$

$\chi_\mathsf{m} \longleftarrow \bigwedge_{s \in S} s$ `// Collect open sequents`