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

Stefan Mitsch       André Platzer

Computer Science Department, Carnegie Mellon University

RV, Sept. 24, 2014

Simplex for Hybrid System Models
Formal Verification in CPS Development

Real CPS

Proof

Reachability Analysis

Verification Results

safe

Challenge

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

Verifiably correct runtime model validation

Stefan Mitsch, André Platzer—ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models
Formal Verification in CPS Development

Real CPS

Model $\alpha^*$

Control $\alpha_{\text{ctrl}}$

\[ v := v + 1 \]

Plant $\alpha_{\text{plant}}$

\[ x' = v \]

abstract

Verification Results

Proof

Reachability Analysis

safe

Challenge

Verification results about models only apply if CPS fits to the model $\Rightarrow$ Verifiably correct runtime model validation.
Formal Verification in CPS Development

Real CPS

Model

Challenge

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

⇝ Verifiably correct runtime model validation
ModelPlex ensures that verification results about models apply to CPS implementations.

ModelPlex \( \alpha \) ensures that verification results about models apply to CPS implementations.
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?
ModelPlex at Runtime

“Simplex for Models”

Controller

Sensors

Actuators

Stefan Mitsch, André Platzer—ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System 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

Challenge What conditions do the monitors need to check to be safe?
Is current CPS behavior included in the behavior of the model?

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

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

\[ \text{CPS observed through sensors} \]
\[ \text{Model describes behavior of CPS between states} \]

**Challenge**

Model describes behavior,
but at runtime we get sampled observations

\[ \rightsquigarrow \text{Transform model into observation-monitor} \]

Detect non-compliance as soon as possible to initiate safe fallback actions
When are two states linked through a run of model $\alpha$?
When are two states linked through a run of model $\alpha$?

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

reachability relation of $\alpha$
When are two states linked through a run of model $\alpha$?

- A prior state characterized by $x^-$
- A posterior state characterized by $x^+$

**Offline**

**Semantical:** $(x^-, x^+) \in \rho(\alpha) 
\Leftrightarrow \text{Theorem}

**Logic (dL):** $(x = x^-) \rightarrow \langle \alpha(x) \rangle (x = x^+)$

Starting at $x = x^-$ exists a run of $\alpha$ to a state where $x = x^+$
When are two states linked through a run of model $\alpha$?

**Semantical:**

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

**Logic ($d\mathcal{L}$):**

$$(x = x^-) \Rightarrow \langle \alpha(x) \rangle (x = x^+)$$

**Real arithmetic:**

$$F(x^-, x^+)$$

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

check at runtime (efficient)
Monitor Characterization

When are two states linked through a run of model $\alpha$?

- A prior state characterized by $x^-$
- A posterior state characterized by $x^+$

Offline

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

Logic ($\mathcal{dL}$): $(x = x^-) \rightarrow \langle \alpha(x) \rangle (x = x^+)$

Real arithmetic:

- $F(x^-, x^+)$
- Check at runtime (efficient)

Theorem:

Starting at $x = x^-$ exists a run of $\alpha$ to a state where $x = x^+$
Proof calculus of $\mathcal{dL}$ executes models symbolically

- Proof attempt:
  \[(x = x^-) \rightarrow \langle \alpha(x) \rangle (x = x^+)\]
Proof calculus of $d\mathcal{L}$ executes models symbolically

Proof attempt:

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

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

Model Monitor

Immediate detection of model violation $\Rightarrow$ Mitigates safety issues with safe fallback action
Proof calculus of $d\mathcal{L}$ executes models symbolically

Proof attempt:

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

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

Model $\alpha$

prior state $x^-$

$i-1$

climb

descend

$i$

posterior state $x^+$

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

Model Monitor

Immediate detection of model violation $\Rightarrow$ Mitigates safety issues with safe fallback action
Proof calculus of $d\mathcal{L}$ executes models symbolically

Proof attempt

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

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

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

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

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

Model $\alpha$

prior state $x^-$

$i-1$

climb
descend

$i$

posterior state $x^+$

Stefan Mitsch, André Platzer—ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models 8 of 15
Provable Correct Synthesis of Monitors

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

![Diagram showing state transitions with climb and descend actions](image)

- Model $\alpha$

  - Prior state $x^-$
  - Posterior state $x^+$

  - Proof attempt
    
    \[(x = x^-) \implies \langle\text{climb} \cup \text{descend}\rangle (x = x^+)\]

    \[\langle\text{climb}\rangle (x = x^+) \lor \langle\text{descend}\rangle (x = x^+)\]

    \[F_1(x^-, x^+) \lor F_2(x^-, x^+)\]

- 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 $\rightsquigarrow$ execute at runtime
Proof calculus of $dL$ executes models symbolically

Model Monitor

Immediate detection of model violation

Mitigates safety issues with safe fallback action

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 $\leadsto$ execute at runtime
For typical models $\text{ctrl; plant}$ we can check earlier.
Early Compliance Checks for Controllers

Model Monitor

prior state $x^{-}$  \rightarrow  Model $\alpha$  \rightarrow  posterior state $x^{+}$

Stefan Mitsch, André Platzer—ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models
Early Compliance Checks for Controllers

Model $\alpha$

prior state $x^-$

$i-1$ -> ctrl -> plant -> $i$

Controller Monitor before actuation
posterior state $x^+$

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

Theorem $(x^- = x^-) \rightarrow \langle \text{ctrl}(x^-) \rangle (x^- = x^+)$

Logic (dL): starting at $x^- = x^-$ exists a run of ctrl to a state where $x^- = x^+$

Real arithmetic:

Controller Monitor
Immediate detection of unsafe control before actuation
$\Rightarrow$ Safe execution of unverified implementations in perfect environments
Early Compliance Checks for Controllers

\begin{align*}
\text{prior state } x^- & \quad i-1 \quad \text{Controller Monitor before actuation} \\
\text{posterior state } x^+ & \quad i \\
\end{align*}

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

Logic \((dL):(x = x^-) \rightarrow \langle \text{ctrl}_{(x)} \rangle (x = x^+)\)

starting at \(x = x^-\) exists a run of \(\text{ctrl}\) to a state where \(x = x^+\)


\textbf{Theorem}
Early Compliance Checks for Controllers

\[ Model \alpha \]

\[ i-1 \quad ctrl \quad plant \quad i \]

Controller Monitor before actuation

posterior state \( x^+ \)

prior state \( x^- \)

Semantical:

\[ (x^-, x^+) \in \rho(\text{ctrl}) \]

Logic (\( d\mathcal{L} \)):

\[ (x = x^-) \rightarrow \langle \text{ctrl}_{(x)} \rangle (x = x^+) \]

Real arithmetic:

\[ F(x^-, x^+) \]

starting at \( x = x^- \)
exists a run of \( \text{ctrl} \) to
a state where \( x = x^+ \)

Offline
Early Compliance Checks for Controllers

Controller Monitor

Immediate detection of unsafe control before actuation
⇒ Safe execution of unverified implementations
in perfect environments

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

Logic (dL): \((x = x^-) \rightarrow \langle \text{ctrl}(x) \rangle (x = x^+)\)

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

Theorem

Starting at \(x = x^-\) exists a run of ctrl to
a state where \(x = x^+\)
Safe despite evolution with disturbance?
Compliance Checks despite Disturbance

prior state $x^-$ $\rightarrow i-1$ $\rightarrow$ Model $\alpha$ ctrl $\rightarrow$ plant $\rightarrow$ $i$

posterior state $x^+$

Invariant state $\varphi$ implies safety (known from safety proof)

Logic ($dL$):

$\Rightarrow dL$ proof

$F(x^-, x^+)$

Real arithmetic:

Prediction Monitor with Disturbance

Proactive detection of unsafe control before actuation

$\Rightarrow$ Safety in realistic environments
Compliance Checks despite Disturbance

plant of the form \( x' = \theta \land H \)

prior state \( x^{-} \)

Model \( \alpha \)

\( i - 1 \) → ctrl → plant → \( i \)

Prediction Monitor before actuation

posterior state \( x^{+} \)

offline \((x = x^{-}) \rightarrow \langle ctrl(x) \rangle (x = x^{+} \land \left[ plant(x) \right] \phi)\)

Invariant state \( \phi \) implies safety (known from safety proof)

Logic (dL):

\[ \uparrow dL \text{proof} \]

F \( (x^{-}, x^{+}) \)

Real arithmetic:

Prediction Monitor with Disturbance
Proactive detection of unsafe control before actuation despite disturbance

⇒ Safety in realistic environments
Compliance Checks despite Disturbance

\[
\text{time bound } t := 0; \left(x' = \theta, \ t' = 1 \& H \land t \leq \varepsilon \right)
\]

prior state \( x^{-} \)

Prediction Monitor
before actuation
posterior state \( x^{+} \)

states reachable
within \( \varepsilon \) time
Compliance Checks despite Disturbance

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

Prior state $x^{-}$

Model $\alpha$

Ctrl

Plant

Posterior state $x^{+}$

Prediction Monitor before actuation

States reachable within $\varepsilon$ time

Invariant state $\phi$ implies safety (known from safety proof)

Logic ($\mathcal{L}$):

$\uparrow \mathcal{L}/proof \ F(x^{-}, x^{+})$

Real arithmetic:
Compliance Checks despite Disturbance

\[
\text{disturbance } t := 0; \left( \theta - \delta \leq x' \leq \theta + \delta, \ t' = 1 \land H \land t \leq \varepsilon \right)
\]

\[ \text{prior state } x^{-} \xrightarrow{i-1} \text{ctrl} \xrightarrow{\text{plant}} \text{posterior state } x^{+} \]

\[
\text{Logic (dL): } (x = x^{-}) \rightarrow \langle \text{ctrl}(x) \rangle \left( x = x^{+} \land [\text{plant}(x)] \varphi \right)
\]

\[
\text{Real arithmetic: } F\left(x^{-}, x^{+}\right)
\]

\[\uparrow \text{dL proof} \]

\[\text{Invariant state } \varphi \text{ implies safety (known from safety proof)} \]

\[\text{Prediction Monitor before actuation} \]

\[\text{states reachable within } \varepsilon \text{ time} \]
Compliance Checks despite Disturbance

Disturbance $t := 0; \left( \theta - \delta \leq x' \leq \theta + \delta, \; t' = 1 & H \land t \leq \varepsilon \right)$

Model $\alpha$

Prior state $x_i-1$

Prediction Monitor with Disturbance

Proactive detection of unsafe control before actuation despite disturbance

$\leadsto$ Safety in realistic environments

Offline

Logic ($dL$): $(x = x^-) \rightarrow \langle ctrl(x) \rangle \left( x = x^+ \land [plant(x)]\varphi \right)$

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

Invariant state $\varphi$ implies safety (known from safety proof)
Evaluation

- Evaluated on hybrid system case studies

Water tank  |  Cruise control  |  Traffic control  |  Ground robots  |  Train control

- 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
Theorems

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

\( \mathcal{V} \) set of variables whose state we want to recall

\[ \Upsilon^{-}_V \equiv \wedge_{x \in \mathcal{V}} x = x^{-} \] characterizes a state prior to a run of \( \alpha \) (fresh variables \( x^{-} \) occur solely in \( \Upsilon^{-}_V \) and recall this state)

\[ \Upsilon^{+}_V \equiv \wedge_{x \in \mathcal{V}} x = x^{+} \] characterizes the posterior states (fresh \( x^{+} \))

**Programs** hybrid program \( \alpha, \alpha^{*} \) repeats \( \alpha \) 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

\[ \Upsilon^{i}_{V} \equiv \wedge_{x \in \mathcal{V}} x = x^{i} \text{ and } \Upsilon^{i-1}_{V} \] are plausible w.r.t. the model \( \alpha \), i.e., \( | = \wedge_{1 \leq i \leq n} \left( \Upsilon^{i-1}_{V} \rightarrow \langle \alpha \rangle \Upsilon^{i}_{V} \right) \) with \( \Upsilon^{-}_V = \Upsilon^{0}_V \text{ and } \Upsilon^{+}_V = \Upsilon^{n}_V. \)

**Then** the sequence of states originates from an \( \alpha^{*} \) execution from \( \Upsilon^{0}_V \) to \( \Upsilon^{n}_V \), i.e., \( | = \Upsilon^{-}_V \rightarrow \langle \alpha^{*} \rangle \Upsilon^{+}_V. \)
\[ \models \phi \rightarrow [\alpha^*]\psi \quad \alpha^* \text{ is provably safe} \]

**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**

\( (\nu, \nu_{i+1}) \models \chi_m \) as \( \chi_m \) evaluated in the state resulting from \( \nu \) by interpreting \( x^+ \) as \( \nu_{i+1}(x) \) for all \( x \in V_m \), i.e.,

\[ \nu_{x^+}^{\nu_{i+1}} \models \chi_m \]

**Correctness**

If \( (\nu_i, \nu_{i+1}) \models \chi_m \) for all \( i < n \) then we have \( \nu_n \models \psi \) where

\[ \chi_m \equiv \left( \phi|_{\text{const}} \rightarrow \langle \alpha \rangle \Upsilon^+_V \right) \]

and \( \phi|_{\text{const}} \) denotes the conditions of \( \phi \) that involve only constants that do not change in \( \alpha \), i.e.,

\[ FV(\phi|_{\text{const}}) \cap BV(\alpha) = \emptyset. \]
Controller Monitor Correctness

\[ \models \phi \rightarrow [\alpha^*] \psi \]  \( \alpha^* \) is provably safe with invariant \( \varphi \)

**Definitions** Let \( \alpha \) of the canonical form \( \alpha_{\text{ctrl}}; \alpha_{\text{plant}} \); let \( \nu \models \phi |_{\text{const}} \land \varphi \), as checked by \( \chi_m \); let \( \tilde{\nu} \) be a post-controller state.

**Controller Monitor** \( (\nu, \tilde{\nu}) \models \chi_c \) as \( \chi_c \) evaluated in the state resulting from \( \nu \) by interpreting \( x^+ \) as \( \tilde{\nu}(x) \) for all \( x \in V_c \), i.e., \( \nu_{\tilde{\nu}(x)} |_{x^+} \models \chi_c \)

**Correctness** If \( (\nu, \tilde{\nu}) \models \chi_c \) where

\[
\chi_c \equiv \phi |_{\text{const}} \rightarrow \langle \alpha_{\text{ctrl}} \rangle \Upsilon^+_V
\]

then we have that \( (\nu, \tilde{\nu}) \in \rho(\alpha_{\text{ctrl}}) \) and \( \tilde{\nu} \models \varphi \).
\[ \models \phi \rightarrow [\alpha^*] \psi \quad \alpha^* \text{ is provably safe with invariant } \varphi \]

**Definitions** Let \( V_p = BV(\alpha) \cup FV([\alpha] \varphi) \). Let \( \nu \models \phi \mid_{\text{const}} \land \varphi \), as checked by \( \chi_m \). Further assume \( \tilde{\nu} \) such that \( (\nu, \tilde{\nu}) \in \rho(\alpha_{\text{ctrl}}) \), as checked by \( \chi_c \).

**Prediction Monitor** \( (\nu, \tilde{\nu}) \models \chi_p \) as \( \chi_p \) evaluated in the state resulting from \( \nu \) by interpreting \( x^+ \) as \( \tilde{\nu}(x) \) for all \( x \in V_p \), i.e., \( \nu_{x^+} \tilde{\nu}(x) \models \chi_p \)

**Correctness** If \( (\nu, \tilde{\nu}) \models \chi_p \) where

\[
\chi_p \equiv (\phi \mid_{\text{const}} \land \varphi) \rightarrow \langle \alpha_{\text{ctrl}} \rangle (\Upsilon_{V_p}^+ \land [\alpha_{\delta_{\text{plant}}}] \varphi)
\]

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 $\alpha_{\text{plant}}$
- disturbed plants $\alpha_{\delta_{\text{plant}}}$ with constant additive disturbance $\delta$

Decidability

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

\[ \chi_m \rightarrow \langle \alpha \rangle (\bigwedge V \land [\alpha_{\delta_{\text{plant}}}]) \]

are decidable

Computability

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

\[ \text{synth}_m : \langle \alpha \rangle (\bigwedge V) \mapsto \chi_m \]
\[ \text{synth}_c : \langle \alpha_{\text{ctrl}} \rangle (\bigwedge V) \mapsto \chi_c \]
\[ \text{synth}_p : \langle \alpha \rangle (\bigwedge V \land [\alpha_{\delta_{\text{plant}}}]) \mapsto \chi_p \]

are computable
Water Tank Example: Monitor Conjecture

Variables

\[\begin{align*}
    x & \text{ current level} \\
    m & \text{ maximum level} \\
    \varepsilon & \text{ control cycle} \\
    f & \text{ flow}
\end{align*}\]

Model and Safety Property

\[
0 \leq x \leq m \land \varepsilon > 0 \rightarrow \begin{cases} 
    (f := \ast; \ ?(-1 \leq f \leq \frac{m-x}{\varepsilon}); \\
    t := 0; (x' = f, \ t' = 1 \land x \geq 0 \land t \leq \varepsilon))^* \end{cases}
\]

Model Monitor Specification Conjecture

\[
\begin{align*}
    \varepsilon > 0 \rightarrow \bigg( f := \ast; \ ?(-1 \leq f \leq \frac{m-x}{\varepsilon}); \\
    t := 0; (x' = f, \ t' = 1 \land x \geq 0 \land t \leq \varepsilon) \bigg)a_r^+_{\psi_{V_m}} (x = x^+ \land f = f^+ \land t = t^+)
\end{align*}
\]
Water Tank Example: Nondeterministic Assignment

Proof Rules

\[
(\langle*\rangle) \quad \frac{\exists X \langle x := X \rangle \phi}{\langle x := * \rangle \phi} \quad 1 \\
(\exists r) \quad \frac{\Gamma \vdash \phi(\theta), \exists x \phi(x), \Delta}{\Gamma \vdash \exists x \phi(x), \Delta} \quad 2 \\
(\text{Wr}) \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \phi, \Delta}
\]

1. \(X\) is a new logical variable
2. \(\theta\) is an arbitrary term, often a new (existential) logical variable \(X\).

Sequent Deduction

\[
\phi \vdash \langle f := F \rangle \langle ?-1 \leq f \leq \frac{m-x}{\epsilon} \rangle \langle \text{plant} \rangle \gamma^+ \\
\phi \vdash \exists F \langle f := F \rangle \langle ?-1 \leq f \leq \frac{m-x}{\epsilon} \rangle \langle \text{plant} \rangle \gamma^+ \\
\langle*\rangle \quad \frac{\phi \vdash \langle f := * \rangle \langle ?-1 \leq f \leq \frac{m-x}{\epsilon} \rangle \langle \text{plant} \rangle \gamma^+}{\exists r, \text{Wr} \ldots}
\]

with Opt. 1 (anticipate \(f = f^+\) from \(\gamma^+\))
Water Tank Example: Differential Equations

Proof Rules

\[
\begin{align*}
\exists T \geq 0 \left( (\forall 0 \leq \tilde{t} \leq T \langle x := y(\tilde{t}) \rangle H) \land \langle x := y(T) \rangle \phi \right) & \quad \text{(QE)} \\
\langle x' = \theta \land H \rangle \phi & \quad \text{(QE)} \phi
\end{align*}
\]

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 \( \phi \equiv \text{QE}(\phi) \), \( \phi \) is a first-order real arithmetic formula, \( \text{QE}(\phi) \) is an equivalent quantifier-free formula.

Sequent Deduction

\[
\phi \vdash F = f^+ \land x^+ = x + Ft^+ \land t^+ \geq 0 \land x \geq 0 \land \epsilon \geq t^+ \geq 0 \land Ft^+ + x \geq 0
\]

\[
\phi \vdash \forall 0 \leq \tilde{t} \leq T (x + f^+\tilde{t} \geq 0 \land \tilde{t} \leq \epsilon) \land F = f^+ \land x^+ = x + Ft^+ \land t^+ = t^+
\]

\[
\exists r, W \phi \vdash \exists T \geq 0 ((\forall 0 \leq \tilde{t} \leq T (x + f^+\tilde{t} \geq 0 \land \tilde{t} \leq \epsilon)) \land F = f^+ \land (x^+ = x + FT \land t^+ = T))
\]

\[
\phi \vdash \langle f := F; t := 0 \rangle \langle \{x' = f, t' = 1 \land x \geq 0 \land t \leq \epsilon\} \rangle \Upsilon^+
\]
## Evaluation

<table>
<thead>
<tr>
<th>Case Study</th>
<th>Model</th>
<th>Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dim.</td>
<td>dim.</td>
</tr>
<tr>
<td></td>
<td>proof</td>
<td>steps (open seq.)</td>
</tr>
<tr>
<td></td>
<td>size</td>
<td>w/ Opt. 1 auto</td>
</tr>
<tr>
<td></td>
<td>(branches)</td>
<td></td>
</tr>
<tr>
<td>Water tank</td>
<td>5</td>
<td>3 16 (2)</td>
</tr>
<tr>
<td>Cruise control</td>
<td>11</td>
<td>7 127 (13)</td>
</tr>
<tr>
<td>Speed limit</td>
<td>9</td>
<td>6 487 (32)</td>
</tr>
<tr>
<td>Water tank</td>
<td>5</td>
<td>1 12 (2)</td>
</tr>
<tr>
<td>Cruise control</td>
<td>11</td>
<td>7 83 (13)</td>
</tr>
<tr>
<td>Ground robot</td>
<td>14</td>
<td>11 94 (10)</td>
</tr>
<tr>
<td>ETCS safety</td>
<td>16</td>
<td>13 162 (13)</td>
</tr>
<tr>
<td>Water tank</td>
<td>8</td>
<td>1 135 (4)</td>
</tr>
</tbody>
</table>

- **Theorem:** ModelPlex is decidable and monitor synthesis can be automated in important classes.
Algorithm 1: ModelPlex monitor synthesis

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

output: A monitor $\chi_m$ such that $|\vdash \chi_m \equiv \phi|_{\text{const}} \rightarrow \langle \alpha \rangle \gamma^+$. 

begin

$S \leftarrow \emptyset$
$\gamma^+ \leftarrow \bigwedge_{x \in \mathcal{V}} x = x^+$ with fresh variables $x_i^+$ // Monitor conjecture

$G \leftarrow \{ \vdash \phi|_{\text{const}} \rightarrow \langle \alpha \rangle \gamma^+ \}$

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

foreach $g \in G$ do

$G \leftarrow G - \{ g \}$

if $g$ is first-order then

\begin{align*}
&\text{if } \not\vdash g \text{ then } S \leftarrow S \cup \{ g \} \\
&\text{else}
\end{align*}

$\tilde{g} \leftarrow \text{apply dL proof rule to } g$

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

\end{align*}

\end{verbatim}

$\chi_m \leftarrow \bigwedge_{s \in S} s$ // Collect open sequents