A Timed Approximation based Compositional Approach towards Formally Verified Aircraft Control Protocols

Pavithra Prabhakar

IMDEA Software Institute & Kansas State University

Safe and Secure Systems and Software Symposium
Motivation

Aircraft control protocols are multi-agent hybrid systems

- Aircraft landing protocols
- Aircraft collision avoidance protocols
Aircraft control protocols are multi-agent hybrid systems

- Aircraft landing protocols
- Aircraft collision avoidance protocols

Current approaches in hybrid systems theory for safety analysis

- Reachable set computation based on fixpoint iterations [SpaceEx, d/dt, Flow*]
- Finite state abstraction based approaches [Predicate abstraction, CEGAR]
Motivation

Aircraft control protocols are multi-agent hybrid systems

- Aircraft landing protocols
- Aircraft collision avoidance protocols

Current approaches in hybrid systems theory for safety analysis

- Reachable set computation based on fixpoint iterations [SpaceEx, d/dt, Flow*]
- Finite state abstraction based approaches [Predicate abstraction, CEGAR]

Shortcomings

- Monolithic view for analysis doesn’t work — leads to the state-space explosion problem
- Symbolic and abstraction based approaches don’t suffice — don’t have enough information for compositional analysis
Motivation

Aircraft control protocols are multi-agent hybrid systems

- Aircraft landing protocols
- Aircraft collision avoidance protocols

Current approaches in hybrid systems theory for safety analysis

- Reachable set computation based on fixpoint iterations [SpaceEx, d/dt, Flow*]
- Finite state abstraction based approaches [Predicate abstraction, CEGAR]

Shortcomings

- Monolithic view for analysis doesn’t work — leads to the state-space explosion problem
- Symbolic and abstraction based approaches don’t suffice — don’t have enough information for compositional analysis

Time is crucial!
Agenda

Safety verification of networks of hybrid systems
Agenda

Safety verification of networks of hybrid systems

Develop a compositional approach
Safety verification of networks of hybrid systems

Develop a compositional approach

Timed Approximations

- Bounded Error Approximations based Verification (BEAVER)
- Hybridization based CEGAR (HARE: Hybrid Abstraction Refinement Engine)
Bounded Error Approximations
Air traffic collision avoidance protocol

\[ \mathbf{x} = (x_1, x_2): \text{ position of the airplane} \]
\[ \mathbf{d} = (d_1, d_2): \text{ velocity of the airplane} \]
Air traffic collision avoidance protocol

The aircraft maintain a minimum distance between them always.

Minimum separation

$x = (x_1, x_2)$: position of the airplane

d = $(d_1, d_2)$: velocity of the airplane
Air traffic collision avoidance protocol

Minimum separation

The aircraft maintain a minimum distance between them always

\[ \mathbf{x} = (x_1, x_2): \text{position of the airplane} \]
\[ \mathbf{d} = (d_1, d_2): \text{velocity of the airplane} \]
Air traffic collision avoidance protocol

Minimum separation

The aircraft maintain a minimum distance between them always

\[ x = (x_1, x_2): \text{position of the airplane} \]
\[ d = (d_1, d_2): \text{velocity of the airplane} \]
Air traffic collision avoidance protocol

Minimum separation

The aircraft maintain a minimum distance between them always

\[ x = (x_1, x_2): \text{position of the airplane} \]
\[ d = (d_1, d_2): \text{velocity of the airplane} \]
Air traffic collision avoidance protocol

\[ \mathbf{x} = (x_1, x_2) : \text{position of the airplane} \]
\[ \mathbf{d} = (d_1, d_2) : \text{velocity of the airplane} \]

Minimum separation

The aircraft maintain a minimum distance between them always
Air traffic collision avoidance protocol

The aircraft maintain a minimum distance between them always.

\[ x = (x_1, x_2): \text{position of the airplane} \]
\[ d = (d_1, d_2): \text{velocity of the airplane} \]
Air traffic collision avoidance protocol

Minimum separation

The aircraft maintain a minimum distance between them always

\[ x = (x_1, x_2): \text{position of the airplane} \]
\[ d = (d_1, d_2): \text{velocity of the airplane} \]
Air traffic collision avoidance protocol

The aircraft maintain a minimum distance between them always.

\[ \mathbf{x} = (x_1, x_2): \text{position of the airplane} \]
\[ \mathbf{d} = (d_1, d_2): \text{velocity of the airplane} \]
Air traffic collision avoidance protocol

Minimum separation

The aircraft maintain a minimum distance between them always

\[
\mathbf{x} = (x_1, x_2): \text{ position of the airplane}
\]

\[
\mathbf{d} = (d_1, d_2): \text{ velocity of the airplane}
\]

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{d}_1 \\
\dot{d}_2
\end{bmatrix}
= 
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -\omega & 0 \\
0 & 0 & 0 & -\omega
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
d_1 \\
d_2
\end{bmatrix}
\]

\(\omega: \text{ the angular velocity}\)
Air traffic collision avoidance protocol

\[ \mathbf{x} = (x_1, x_2): \text{position of the airplane} \]
\[ \mathbf{d} = (d_1, d_2): \text{velocity of the airplane} \]

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{d}_1 \\
\dot{d}_2 \\
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -\omega & 0 \\
0 & 0 & \omega & 0 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
d_1 \\
d_2 \\
\end{bmatrix}
\]

\( \omega: \text{the angular velocity} \)

Minimum separation

The aircraft maintain a minimum distance between them always

\[ \| \mathbf{x} - \mathbf{y} \| \leq p \]
\[ c = x + \lambda d = y + \lambda e \]
\[ \| \mathbf{x} - c \| = \sqrt{3}r \]

\[ (r\omega)^2 = \| \mathbf{d} \|^2 \]
\[ x^0 := x, \quad d^0 := d \]

\( \omega := * \)

collision detection & negotiation

parallel to its initial direction

reach inner circle

\( \omega := -\omega \)

\[ \omega := 0 \]
\[ x + \lambda_2 \mathbf{d} = x^0 + \lambda_1 d^0 \]
Reach set computation based Safety verification

Post computation problem:

\[
\begin{align*}
\dot{x} &= f(x) \\
\Phi_f(x_0, t) &= \ \text{\(\Phi_f(x_0, t)\)}
\end{align*}
\]
Reach set computation based Safety verification

Post computation problem:

<table>
<thead>
<tr>
<th>\dot{x} = f(x)</th>
<th>\Phi f(x_0, t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $f(x) = c$, $\Phi(x_0, t) = x_0 + ct$</td>
<td></td>
</tr>
<tr>
<td>If $f(x) = Ax$, $\Phi(x_0, t) = e^{At}x_0$</td>
<td></td>
</tr>
</tbody>
</table>
Reach set computation based Safety verification

Post computation problem:

\[
\begin{align*}
\dot{x} &= f(x) \\
\Phi_f(x_0, t) &= 
\begin{cases}
  f(x) = c, & \Phi(x_0, t) = x_0 + ct \\
  f(x) = Ax, & \Phi(x_0, t) = e^{At}x_0
\end{cases}
\end{align*}
\]

\[Post_f(X_0, T) = \{ x \mid x_0 \in X_0, \Phi_f(x_0, t) \} \]
Reach set computation based Safety verification

Post computation problem:

\[ \dot{x} = f(x) \]
\[ \Phi_f(x_0, t) \]

If \( f(x) = c \), \( \Phi(x_0, t) = x_0 + ct \)

If \( f(x) = Ax \), \( \Phi(x_0, t) = e^{At}x_0 \)

\[ Post_f(X_0, T) = \{ x \mid x_0 \in X_0, \Phi_f(x_0, t) \} \]

Safety verification

- State-space exploration based on discrete and continuous post operation
- Intersection with guards, emptiness checking
Reach set computation based Safety verification

Post computation problem:

\[
\dot{x} = f(x) \quad \Phi_f(x_0, t)
\]

If \( f(x) = c \), \( \Phi(x_0, t) = x_0 + ct \)

If \( f(x) = Ax \), \( \Phi(x_0, t) = e^{At}x_0 \)

\[
Post_f(X_0, T) = \{ x \mid x_0 \in X_0, \Phi_f(x_0, t) \}
\]

Safety verification

- State-space exploration based on discrete and continuous post operation
- Intersection with guards, emptiness checking
Reach set computation based Safety verification

Post computation problem:

\[ \dot{x} = f(x) \]
\[ \Phi_f(x_0, t) \]

If \( f(x) = c \), \( \Phi(x_0, t) = x_0 + ct \)

If \( f(x) = Ax \), \( \Phi(x_0, t) = e^{At}x_0 \)

\[ Post_f(X_0, T) = \{ x \mid x_0 \in X_0, \Phi_f(x_0, t) \} \]

Safety verification

- State-space exploration based on discrete and continuous post operation
- Intersection with guards, emptiness checking

Class of Continuous Dynamics

Complexity of Verification

FSM

TIMED

\[ \dot{x} = 1 \]
Reach set computation based Safety verification

Post computation problem:

\[
\begin{align*}
\dot{x} &= f(x) \\
\Phi_f(x_0, t) &= \begin{cases} 
& x_0 + ct \quad \text{if } f(x) = c, \\
& e^{At}x_0 \quad \text{if } f(x) = Ax,
\end{cases}
\end{align*}
\]

Post\_f(X_0, T) = \{ x | x_0 \in X_0, \Phi_f(x_0, t) \}

Safety verification
- State-space exploration based on discrete and continuous post operation
- Intersection with guards, emptiness checking
Reach set computation based Safety verification

Post computation problem:

\[
\dot{x} = f(x) \\
\Phi_f(x_0, t)
\]

If \( f(x) = c \), \( \Phi(x_0, t) = x_0 + ct \)

If \( f(x) = Ax \), \( \Phi(x_0, t) = e^{At}x_0 \)

\[\text{Post}_f(X_0, T) = \{x \mid x_0 \in X_0, \Phi_f(x_0, t)\}\]

Safety verification

- State-space exploration based on discrete and continuous post operation
- Intersection with guards, emptiness checking

Complexity of Verification

Class of Continuous Dynamics

Exponential \( \dot{x} = 1 \)  RECTANGULAR

FSM  TIMED
Reach set computation based Safety verification

Post computation problem:

\[
\begin{align*}
\dot{x} &= f(x) \\
\Phi_f(x_0, t) &= \begin{cases} 
  x_0 + ct & \text{if } f(x) = c \\
  e^{At} x_0 & \text{if } f(x) = Ax
\end{cases}
\end{align*}
\]

\[Post_f(X_0, T) = \{ x \mid x_0 \in X_0, \Phi_f(x_0, t) \}\]

Safety verification

- State-space exploration based on discrete and continuous post operation
- Intersection with guards, emptiness checking
Reach set computation based Safety verification

Post computation problem:

\[
\dot{x} = f(x) \\
\Phi_f(x_0, t)
\]

If \( f(x) = c \), \( \Phi(x_0, t) = x_0 + ct \)

If \( f(x) = Ax \), \( \Phi(x_0, t) = e^{At}x_0 \)

\[
Post_f(X_0, T) = \{x \mid x_0 \in X_0, \Phi_f(x_0, t)\}
\]

Complexity of Verification

Safety verification

- State-space exploration based on discrete and continuous post operation
- Intersection with guards, emptiness checking
Reach set computation based Safety verification

Post computation problem:

\[
\begin{align*}
\dot{x} &= f(x) \\
\Phi_f(x_0, t) &= \\
\text{If } f(x) = c, \Phi(x_0, t) &= x_0 + ct \\
\text{If } f(x) = Ax, \Phi(x_0, t) &= e^{At}x_0
\end{align*}
\]

\[
Post_f(X_0, T) = \{ x | x_0 \in X_0, \Phi_f(x_0, t) \}
\]

Complexity of Verification

Class of Continuous Dynamics

Safety verification

- State-space exploration based on discrete and continuous post operation
- Intersection with guards, emptiness checking
Reach set computation based Safety verification

Post computation problem:

\[ \dot{x} = f(x) \]
\[ \Phi_f(x_0, t) \]
If \( f(x) = c \), \( \Phi(x_0, t) = x_0 + ct \)
If \( f(x) = Ax \), \( \Phi(x_0, t) = e^{At}x_0 \)

\[ Post_f(X_0, T) = \{ x \mid x_0 \in X_0, \Phi_f(x_0, t) \} \]

Safety verification:
- State-space exploration based on discrete and continuous post operation
- Intersection with guards, emptiness checking

Approximation is a must:
Satisfiability of the theory of reals with exponentiation is an open problem
Approximations for linear dynamical systems

Approximate reach set computation

\[ \Phi(x_0, t) \]

Compute an \( \epsilon \) over-approximation \( R \)

\[ \text{Post}_f(X_0, T) \subseteq R \]
\[ \subseteq B_\epsilon(\text{Post}_f(X_0, T)) \]
Approximations for linear dynamical systems

Approximate reach set computation

Compute an $\epsilon$ over-approximation $R$

$$Post_f(X_0, T) \subseteq R \subseteq B_\epsilon(Post_f(X_0, T))$$
Approximations for linear dynamical systems

Approximate reach set computation

Compute an $\epsilon$ over-approximation $R$

$$Post_f(X_0, T) \subseteq R \subseteq B_\epsilon(Post_f(X_0, T))$$

Approximations for linear dynamical systems

Approximate reach set computation

Compute an $\epsilon$ over-approximation $R$

\[ \text{Post}_f(X_0, T) \subseteq R \subseteq B_\epsilon(\text{Post}_f(X_0, T)) \]
Approximations for linear dynamical systems

Approximate reach set computation

\[ \Phi(x_0, t) \]

\[ R \]

\[ x_0 \]

\[ X_0 \]

Compute an \( \epsilon \) over-approximation \( R \)

\[ \text{Post}_f(X_0, T) \subseteq R \]

\[ \subseteq B_\epsilon(\text{Post}_f(X_0, T)) \]

Approximations for linear dynamical systems

Approximate reach set computation

Compute an $\epsilon$ over-approximation $R$

$$Post_f(X_0, T) \subseteq R \subseteq B_\epsilon(\text{Post}_f(X_0, T))$$
Approximations for linear dynamical systems

Approximate reach set computation

Compute an $\epsilon$ over-approximation $R$

$$Post_f(X_0, T) \subseteq R \subseteq B_\epsilon(Post_f(X_0, T))$$

Approximations for linear dynamical systems

Approximate reach set computation

Compute an $\epsilon$ over-approximation $R$

\[
Post_f(X_0, T) \subseteq R \subseteq B_\epsilon(Post_f(X_0, T))
\]

- Data structure investigated — Polyhedra [Dang, Maler], [Chutinan, Krogh], Ellipsoids [Kurzhanski, Varaiya], Zonotopes, Support functions [Girard, Guernic]

A dynamic algorithm for approximate flow computations. Pavithra Prabhakar and Mahesh Viswanathan.
Parameterized linear systems

Parameterized linear system

\[ \dot{x} = Ax \]

\[ x_0 \in X_0, t \in [0, T] \]

\[ A \in \Omega \]
Parameterized linear systems

Parameterized linear system

\[ \dot{x} = Ax \]
\[ x_0 \in X_0, \, t \in [0, T] \]
\[ A \in \Omega \]

Related work:
Approximate the state transition matrices [Althoff et al]:
\[ \mathcal{M}(\delta) = \{ e^{A\delta} \mid A \in \Omega \} \]
Not straightforward to compute the sampling interval for a given error tolerance
Parameterized linear systems

\[ \dot{x} = Ax \]
\[ x_0 \in X_0, \ t \in [0, T] \]
\[ A \in \Omega \]

Main idea:

- Sample both the parameter space and the time domain
- Construct a piecewise bilinear function interpolating the values at the sample points

Related work:

Approximate the state transition matrices [Althoff et al]:

\[ \mathcal{M}(\delta) = \{ e^{A\delta} \mid A \in \Omega \} \]

Not straightforward to compute the sampling interval for a given error tolerance
Parameterized linear systems

Parameterized linear system

\[ \dot{x} = Ax \]
\[ x_0 \in X_0, \ t \in [0, T] \]
\[ A \in \Omega \]

Related work:

Approximate the state transition matrices [Althoff et al]:

\[ M(\delta) = \{ e^{A\delta} | A \in \Omega \} \]

Not straightforward to compute the sampling interval for a given error tolerance

Main idea:

- Sample both the parameter space and the time domain
- Construct a piecewise bilinear function interpolating the values at the sample points
Parameterized linear systems

Main idea:
- Sample both the parameter space and the time domain
- Construct a piecewise bilinear function interpolating the values at the sample points

For $\omega \in [\omega_1, \omega_2]$ and $t \in [t_1, t_2]$,

$$\hat{\Phi}(x_0, \omega, t) = \left[ \beta \{ \alpha e^{\omega_1 t_1} + (1 - \alpha) e^{\omega_1 t_2} \} + (1 - \beta) \{ \alpha e^{\omega_2 t_1} + (1 - \alpha) e^{\omega_2 t_2} \} \right] x_0$$

where $\alpha = \frac{t - t_2}{t_1 - t_2}$ and $\beta = \frac{\omega - \omega_1}{\omega_1 - \omega_2}$

$$\Phi(x_0, \omega, t) = e^{\omega t} x_0$$
Parameterized linear systems

Main idea:

- Sample both the parameter space and the time domain
- Construct a piecewise bilinear function interpolating the values at the sample points

For $\omega \in [\omega_1, \omega_2]$ and $t \in [t_1, t_2]$, 

$$\hat{\Phi}(x_0, \omega, t) = \left[ \beta \{ \alpha e^{\omega_1 t_1} + (1 - \alpha) e^{\omega_1 t_2} \} + (1 - \beta) \{ \alpha e^{\omega_2 t_1} + (1 - \alpha) e^{\omega_2 t_2} \} \right] x_0$$

where $\alpha = \frac{t - t_2}{t_1 - t_2}$ and $\beta = \frac{\omega - \omega_1}{\omega_1 - \omega_2}$

$$\Phi(x_0, \omega, t) = e^{\omega t} x_0$$
Parameterized linear systems

Main idea:
- Sample both the parameter space and the time domain
- Construct a piecewise bilinear function interpolating the values at the sample points

For $\omega \in [\omega_1, \omega_2]$ and $t \in [t_1, t_2]$,

$$\hat{\Phi}(x_0, \omega, t) = \left[ \beta \{ \alpha e^{\omega_1 t_1} + (1 - \alpha) e^{\omega_1 t_2} \} + (1 - \beta) \{ \alpha e^{\omega_2 t_1} + (1 - \alpha) e^{\omega_2 t_2} \} \right] x_0$$

where $\alpha = \frac{t - t_2}{t_1 - t_2}$ and $\beta = \frac{\omega - \omega_1}{\omega_1 - \omega_2}$
Parameterized linear systems

Main idea:
- Sample both the parameter space and the time domain
- Construct a piecewise bilinear function interpolating the values at the sample points

For $\omega \in [\omega_1, \omega_2]$ and $t \in [t_1, t_2]$,

$$\hat{\Phi}(x_0, \omega, t) = \left[ \beta \{ \alpha e^{\omega_1 t_1} + (1 - \alpha)e^{\omega_1 t_2} \} + (1 - \beta) \{ \alpha e^{\omega_2 t_1} + (1 - \alpha)e^{\omega_2 t_2} \} \right] x_0$$

where $\alpha = \frac{t-t_2}{t_1-t_2}$ and $\beta = \frac{\omega-\omega_1}{\omega_1-\omega_2}$

Bound the precision of approximation:
- Finding the $\delta$ corresponding to an $\epsilon$

$$\max \{ \delta \| \Omega \| e^{\delta \| \Omega \| T}, \delta T e^{\delta T} \} \leq \frac{\epsilon}{4e \| \Omega \| T}$$
BEAVER: Bounded Error Approximation based VERification

Parameterized Linear Hybrid Automaton

Bounded error approximation

Bilinear expressions

SMT formula construction

SMT formula

Safety property

SMT formula verification

Yes/No
**BEAVER**: Bounded Error Approximation based VERification

Parameterized Linear Hybrid Automaton

- Bounded error approximation
- Bilinear expressions
- SMT formula construction
- SMT formula
- SMT formula verification

Safety property

Yes/No
\[ \varphi_{\text{exec}}^{i,\varepsilon}(x_i, t_i) = \varphi_{\text{free}}^{i,\varepsilon} \land \varphi_{\text{entry}}^{i,\varepsilon} \land \varphi_{\text{circ}}^{i,\varepsilon} \land \varphi_{\text{exit}}^{i,\varepsilon} \]
BEAVER: Bounded Error Approximation based VERification

Parameterized Linear Hybrid Automaton

Bounded error approximation

Bilinear expressions

SMT formula construction

SMT formula

SMT formula verification

Yes/No

Parameterized Linear Hybrid Automaton

Bounded error approximation

Bilinear expressions

SMT formula construction

SMT formula

SMT formula verification

Yes/No

\[ \varphi_{\text{exec}}^{i,\epsilon}(x_i, t_i) = \varphi_{\text{free}}^{i,\epsilon} \land \varphi_{\text{entry}}^{i,\epsilon} \land \varphi_{\text{circ}}^{i,\epsilon} \land \varphi_{\text{exit}}^{i,\epsilon} \]

\[ \varphi_{\text{safe}}^{\epsilon} = \neg \exists t [ \varphi_{\text{exec}}^{1,\epsilon}(x_1, t) \land \varphi_{\text{exec}}^{2,\epsilon}(x_2, t) \land \|x_1 - x_2\| \leq d_{\text{sep}} + 2\epsilon ] \]
**BEAVER**: Bounded Error Approximation based VERification

Parameterized Linear Hybrid Automaton → Bounded error approximation → SMT formula construction

Bilinear expressions → SMT formula construction

SMT formula → SMT formula verification → Yes/No

**BEAVER**

\[
\varphi_{\text{exec}}^{i,\epsilon}(x_i, t_i) = \varphi_{\text{free}}^{i,\epsilon} \land \varphi_{\text{entry}}^{i,\epsilon} \land \varphi_{\text{circ}}^{i,\epsilon} \land \varphi_{\text{exit}}^{i,\epsilon}
\]

\[
\varphi_{\text{safety}}^{\epsilon} = \neg \exists t \left[ \varphi_{\text{exec}}^{1,\epsilon}(x_1, t) \land \varphi_{\text{exec}}^{2,\epsilon}(x_2, t) \land \|x_1 - x_2\| \leq d_{\text{sep}} + 2\epsilon \right]
\]

Main highlight of BEAVER — can perform compositional verification
## Analysis results

<table>
<thead>
<tr>
<th>#Aircraft</th>
<th>#locations</th>
<th>epsilon</th>
<th>Time Approx</th>
<th>Time Create SMT</th>
<th>Time Verify (in sec)</th>
<th>Total Time (in seconds)</th>
<th>SMT result</th>
<th>KeYmaera</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>16</td>
<td>15</td>
<td>0.66</td>
<td>1.53</td>
<td>2.73</td>
<td>4.92</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>8</td>
<td>1.30</td>
<td>1.55</td>
<td>10.35</td>
<td>13.20</td>
<td>Unsat</td>
<td>10.4</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>15</td>
<td>0.67</td>
<td>1.54</td>
<td>11.44</td>
<td>13.65</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>8</td>
<td>1.29</td>
<td>1.52</td>
<td>11.69</td>
<td>14.50</td>
<td>Unsat</td>
<td>253.6</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>15</td>
<td>0.82</td>
<td>1.56</td>
<td>12.36</td>
<td>14.74</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>8</td>
<td>1.68</td>
<td>1.55</td>
<td>14.42</td>
<td>17.65</td>
<td>Unsat</td>
<td>382.9</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
<td>15</td>
<td>0.95</td>
<td>1.50</td>
<td>40.81</td>
<td>43.26</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
<td>8</td>
<td>2.05</td>
<td>1.57</td>
<td>35.23</td>
<td>38.85</td>
<td>Unsat</td>
<td>1882.9</td>
</tr>
<tr>
<td>7</td>
<td>16384</td>
<td>15</td>
<td>1.24</td>
<td>1.52</td>
<td>49.66</td>
<td>52.42</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>16384</td>
<td>8</td>
<td>2.70</td>
<td>1.55</td>
<td>51.33</td>
<td>55.58</td>
<td>Unsat</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>1000576</td>
<td>15</td>
<td>1.68</td>
<td>1.56</td>
<td>115.52</td>
<td>118.76</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>1000576</td>
<td>8</td>
<td>3.80</td>
<td>1.61</td>
<td>96.18</td>
<td>101.59</td>
<td>Unsat</td>
<td>—</td>
</tr>
</tbody>
</table>
## Analysis results

<table>
<thead>
<tr>
<th>#Aircraft</th>
<th>#locations</th>
<th>epsilon</th>
<th>Time Approx</th>
<th>Time Create SMT</th>
<th>Time Verify (in sec)</th>
<th>Total Time (in seconds)</th>
<th>SMT result</th>
<th>KeYmaera</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>16</td>
<td>15</td>
<td>0.66</td>
<td>1.53</td>
<td>2.73</td>
<td>4.92</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>8</td>
<td>1.30</td>
<td>1.55</td>
<td>10.35</td>
<td>13.20</td>
<td>Unsat</td>
<td>10.4</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>15</td>
<td>0.67</td>
<td>1.54</td>
<td>11.44</td>
<td>13.65</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>8</td>
<td>1.29</td>
<td>1.52</td>
<td>11.69</td>
<td>14.50</td>
<td>Unsat</td>
<td>253.6</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>15</td>
<td>0.82</td>
<td>1.56</td>
<td>12.36</td>
<td>14.74</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>8</td>
<td>1.68</td>
<td>1.55</td>
<td>14.42</td>
<td>17.65</td>
<td>Unsat</td>
<td>382.9</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
<td>15</td>
<td>0.95</td>
<td>1.50</td>
<td>40.81</td>
<td>43.26</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
<td>8</td>
<td>2.05</td>
<td>1.57</td>
<td>35.23</td>
<td>38.85</td>
<td>Unsat</td>
<td>1882.9</td>
</tr>
<tr>
<td>7</td>
<td>16384</td>
<td>15</td>
<td>1.24</td>
<td>1.52</td>
<td>49.66</td>
<td>52.42</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>16384</td>
<td>8</td>
<td>2.70</td>
<td>1.55</td>
<td>51.33</td>
<td>55.58</td>
<td>Unsat</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>1000576</td>
<td>15</td>
<td>1.68</td>
<td>1.56</td>
<td>115.52</td>
<td>118.76</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>1000576</td>
<td>8</td>
<td>3.80</td>
<td>1.61</td>
<td>96.18</td>
<td>101.59</td>
<td>Unsat</td>
<td>—</td>
</tr>
</tbody>
</table>
### Analysis results

<table>
<thead>
<tr>
<th>#Aircraft</th>
<th>#locations</th>
<th>epsilon</th>
<th>Time Approx</th>
<th>Time Create SMT</th>
<th>Time Verify (in sec)</th>
<th>Total Time (in seconds)</th>
<th>SMT result</th>
<th>KeYmaera</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>16</td>
<td>15</td>
<td>0.66</td>
<td>1.53</td>
<td>2.73</td>
<td>4.92</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>8</td>
<td>1.30</td>
<td>1.55</td>
<td>10.35</td>
<td>13.20</td>
<td>Unsat</td>
<td>10.4</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>15</td>
<td>0.67</td>
<td>1.54</td>
<td>11.44</td>
<td>13.65</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>8</td>
<td>1.29</td>
<td>1.52</td>
<td>11.69</td>
<td>14.50</td>
<td>Unsat</td>
<td>253.6</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>15</td>
<td>0.82</td>
<td>1.56</td>
<td>12.36</td>
<td>14.74</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>8</td>
<td>1.68</td>
<td>1.55</td>
<td>14.42</td>
<td>17.65</td>
<td>Unsat</td>
<td>382.9</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
<td>15</td>
<td>0.95</td>
<td>1.50</td>
<td>40.81</td>
<td>43.26</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
<td>8</td>
<td>2.05</td>
<td>1.57</td>
<td>35.23</td>
<td>38.85</td>
<td>Unsat</td>
<td>1882.9</td>
</tr>
<tr>
<td>7</td>
<td>16384</td>
<td>15</td>
<td>1.24</td>
<td>1.52</td>
<td>49.66</td>
<td>52.42</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>16384</td>
<td>8</td>
<td>2.70</td>
<td>1.55</td>
<td>51.33</td>
<td>55.58</td>
<td>Unsat</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>1000576</td>
<td>15</td>
<td>1.68</td>
<td>1.56</td>
<td>115.52</td>
<td>118.76</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>1000576</td>
<td>8</td>
<td>3.80</td>
<td>1.61</td>
<td>96.18</td>
<td>101.59</td>
<td>Unsat</td>
<td>—</td>
</tr>
</tbody>
</table>

- Our approach scales to more than 10 aircraft
Analysis results

<table>
<thead>
<tr>
<th>#Aircraft</th>
<th>#locations</th>
<th>epsilon</th>
<th>Time Approx</th>
<th>Time Create</th>
<th>Time Verify (in sec)</th>
<th>Total Time (in seconds)</th>
<th>SMT result</th>
<th>KeYmaera</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>16</td>
<td>15</td>
<td>0.66</td>
<td>1.53</td>
<td>2.73</td>
<td>4.92</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>8</td>
<td>1.30</td>
<td>1.55</td>
<td>10.35</td>
<td>13.20</td>
<td>Unsat</td>
<td>10.4</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>15</td>
<td>0.67</td>
<td>1.54</td>
<td>11.44</td>
<td>13.65</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>8</td>
<td>1.29</td>
<td>1.52</td>
<td>11.69</td>
<td>14.50</td>
<td>Unsat</td>
<td>253.6</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>15</td>
<td>0.82</td>
<td>1.56</td>
<td>12.36</td>
<td>14.74</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>8</td>
<td>1.68</td>
<td>1.55</td>
<td>14.42</td>
<td>17.65</td>
<td>Unsat</td>
<td>382.9</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
<td>15</td>
<td>0.95</td>
<td>1.50</td>
<td>40.81</td>
<td>43.26</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
<td>8</td>
<td>2.05</td>
<td>1.57</td>
<td>35.23</td>
<td>38.85</td>
<td>Unsat</td>
<td>1882.9</td>
</tr>
<tr>
<td>7</td>
<td>16384</td>
<td>15</td>
<td>1.24</td>
<td>1.52</td>
<td>49.66</td>
<td>52.42</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>16384</td>
<td>8</td>
<td>2.70</td>
<td>1.55</td>
<td>51.33</td>
<td>55.58</td>
<td>Unsat</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>1000576</td>
<td>15</td>
<td>1.68</td>
<td>1.56</td>
<td>115.52</td>
<td>118.76</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>1000576</td>
<td>8</td>
<td>3.80</td>
<td>1.61</td>
<td>96.18</td>
<td>101.59</td>
<td>Unsat</td>
<td>—</td>
</tr>
</tbody>
</table>

- Our approach scales to more than 10 aircraft
- Scales better than existing approaches — Beaver (3 times) vs KeYmaera (7 times)
<table>
<thead>
<tr>
<th>#Aircraft</th>
<th>#locations</th>
<th>epsilon</th>
<th>Time Approx</th>
<th>Time Create SMT</th>
<th>Time Verify (in sec)</th>
<th>Total Time (in seconds)</th>
<th>SMT result</th>
<th>KeYmaera</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>16</td>
<td>15</td>
<td>0.66</td>
<td>1.53</td>
<td>2.73</td>
<td>4.92</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>8</td>
<td>1.30</td>
<td>1.55</td>
<td>10.35</td>
<td>13.20</td>
<td>Unsat</td>
<td>10.4</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>15</td>
<td>0.67</td>
<td>1.54</td>
<td>11.44</td>
<td>13.65</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>8</td>
<td>1.29</td>
<td>1.52</td>
<td>11.69</td>
<td>14.50</td>
<td>Unsat</td>
<td>253.6</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>15</td>
<td>0.82</td>
<td>1.56</td>
<td>12.36</td>
<td>14.74</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>8</td>
<td>1.68</td>
<td>1.55</td>
<td>14.42</td>
<td>17.65</td>
<td>Unsat</td>
<td>382.9</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
<td>15</td>
<td>0.95</td>
<td>1.50</td>
<td>40.81</td>
<td>43.26</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
<td>8</td>
<td>2.05</td>
<td>1.57</td>
<td>35.23</td>
<td>38.85</td>
<td>Unsat</td>
<td>1882.9</td>
</tr>
<tr>
<td>7</td>
<td>16384</td>
<td>15</td>
<td>1.24</td>
<td>1.52</td>
<td>49.66</td>
<td>52.42</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>16384</td>
<td>8</td>
<td>2.70</td>
<td>1.55</td>
<td>51.33</td>
<td>55.58</td>
<td>Unsat</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>1000576</td>
<td>15</td>
<td>1.68</td>
<td>1.56</td>
<td>115.52</td>
<td>118.76</td>
<td>Sat</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>1000576</td>
<td>8</td>
<td>3.80</td>
<td>1.61</td>
<td>96.18</td>
<td>101.59</td>
<td>Unsat</td>
<td>—</td>
</tr>
</tbody>
</table>

- Our approach scales to more than 10 aircraft
- Scales better than existing approaches — Beaver (3 times) vs KeYmaera (7 times)
- We ignore the error in computation of the value of the solution at the sample points
Hybridization based CEGAR
Abstraction

Safety Analysis
Abstraction

Safety Analysis
Abstraction

Safety Analysis
Abstraction

Safety Analysis
## Abstraction

### Safety Analysis

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Diagram:

1 → 2 → 3
4 → 5 → 6
7 → 8 → 9
Abstraction

Safety Analysis
Abstraction

Safety Analysis
Safety Analysis

- Every trajectory corresponds to a path in the graph
Safety Analysis

+ Every trajectory corresponds to a path in the graph
+ Absence of a path from green to red node implies safety
Abstraction

Safety Analysis

- Every trajectory corresponds to a path in the graph
- Absence of a path from green to red node implies safety
Abstraction

Safety Analysis

- Every trajectory corresponds to a path in the graph
- Absence of a path from green to red node implies safety

The above system is safe
Abstraction

Safety Analysis

- Every trajectory corresponds to a path in the graph
- Absence of a path from green to red node implies safety

- The above system is safe
- The abstract graph has a counter-example
Abstraction

Safety Analysis

- Every trajectory corresponds to a path in the graph
- Absence of a path from green to red node implies safety

- The above system is safe
- The abstract graph has a counter-example
- Right abstractions are hard to find!
Refinement

Safety Analysis
- Every trajectory corresponds to a path in the graph
- Absence of a path from green to red node implies safety
- Refine by analyzing the abstract counter-example

- The above system is safe
- The abstract graph has a counter-example
- Right abstractions are hard to find!
Safety Analysis

- Every trajectory corresponds to a path in the graph
- Absence of a path from green to red node implies safety
- Refine by analyzing the abstract counter-example

The above system is safe
- The abstract graph has a counter-example
- Right abstractions are hard to find!
Refinement

Safety Analysis

- Every trajectory corresponds to a path in the graph
- Absence of a path from green to red node implies safety
- Refine by analyzing the abstract counter-example

- The above system is safe
- The abstract graph has a counter-example
- Right abstractions are hard to find!
Counter-example guided abstraction refinement

- **CEGAR for discrete systems** [Kurshan et al. 93, Clarke et al. 00, Ball et al. 02]
- **CEGAR for hybrid systems by discrete abstractions** [Alur et al. 03, Clarke et al. 03]
Finite State Abstractions

- **Main challenges**
  - Constructing abstractions — requires reachable set computation
  - The abstractions are too coarse for compositional verification
Finite State Abstractions

- **Main challenges**
  - Constructing abstractions — requires reachable set computation
  - The abstractions are too coarse for compositional verification
Finite State Abstractions

Main challenges

- Constructing abstractions — requires reachable set computation
- The abstractions are too coarse for compositional verification
Finite State Abstractions

- **Main challenges**
  - Constructing abstractions — requires reachable set computation
  - The abstractions are too coarse for compositional verification
Hybridization based CEGAR

Abstract a hybrid system by another hybrid system
Hybridization based CEGAR

Abstract a hybrid system by another hybrid system

Linear dynamics:

\[
\begin{pmatrix}
\dot{x} \\
\dot{y}
\end{pmatrix} = \begin{pmatrix}
a & b \\
c & d
\end{pmatrix} \begin{pmatrix}
x \\
y
\end{pmatrix}
\]

Overview:

- Divide the state-space into finite number of regions
- Approximate the dynamics in each of the regions by simpler dynamics
Hybridization based CEGAR

Abstract a hybrid system by another hybrid system

Linear dynamics:

\[
\begin{pmatrix}
\dot{x} \\
\dot{y}
\end{pmatrix} = \begin{pmatrix}
a & b \\
c & d
\end{pmatrix} \begin{pmatrix}
x \\
y
\end{pmatrix}
\]

Overview:

- Divide the state-space into finite number of regions
- Approximate the dynamics in each of the regions by simpler dynamics

Rectangular dynamics:

\[
\begin{array}{c}
(r_2, w_2) \\
(r_1, w_1)
\end{array}
\begin{array}{c}
\dot{x} \in [l_x, u_x] \\
\dot{y} \in [l_y, u_y]
\end{array}
\]

\[\max \ ax + by, \ x \in [r_1, r_2], \ y \in [w_1, w_2]\]
Hybridization based CEGAR

Abstract a hybrid system by another hybrid system

Linear dynamics:
\[
\begin{pmatrix}
\dot{x} \\
\dot{y}
\end{pmatrix} =
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix}
\]

Rectangular dynamics:
\[
(r_2, w_2) \\
\dot{x} \in [l_x, u_x] \\
\dot{y} \in [l_y, u_y] \\
(r_1, w_1)
\]

\[
\max \ ax + by, x \in [r_1, r_2], y \in [w_1, w_2]
\]

Overview:
- Divide the state-space into finite number of regions
- Approximate the dynamics in each of the regions by simpler dynamics

Features:
- Construction of abstraction simpler
- Model-checking is more involved
- Our refinement algorithm splits a region of the state-space partition in the abstraction
HARE: Hybrid Abstraction Refinement Engine

**HARE: Hybrid Abstraction Refinement Engine**

- The algorithm is sound — when the model-checker says yes, the system is safe
- Validation can be performed only approximately

Experimental results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Safety</th>
<th>Vars.</th>
<th>Locs.</th>
<th>Time</th>
<th>Safe</th>
<th>HARE</th>
<th>Time</th>
<th>Safe</th>
<th>PHAVer</th>
<th>HSolver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tank 1</td>
<td>Safe</td>
<td>3</td>
<td>3</td>
<td>&lt; 1</td>
<td>Yes</td>
<td>721</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Tank 2</td>
<td>Safe</td>
<td>3</td>
<td>3</td>
<td>&lt; 1</td>
<td>Yes</td>
<td>763</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Satellite 1</td>
<td>Unsafe</td>
<td>4</td>
<td>64</td>
<td>88</td>
<td>No</td>
<td>16</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Satellite 2</td>
<td>Safe</td>
<td>4</td>
<td>100</td>
<td>&lt; 1</td>
<td>Yes</td>
<td>270</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Satellite 3</td>
<td>Safe</td>
<td>4</td>
<td>576</td>
<td>&lt; 1</td>
<td>Yes</td>
<td>267</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Satellite 4</td>
<td>Safe</td>
<td>4</td>
<td>1296</td>
<td>&lt; 1</td>
<td>Yes</td>
<td>482</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Heater 01</td>
<td>Unsafe</td>
<td>3</td>
<td>4</td>
<td>&lt; 1</td>
<td>No</td>
<td>11</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Heater 02</td>
<td>Unsafe</td>
<td>3</td>
<td>4</td>
<td>&lt; 1</td>
<td>No</td>
<td>21</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Nav 01</td>
<td>Safe</td>
<td>4</td>
<td>25</td>
<td>9</td>
<td>Yes</td>
<td>&lt; 1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Nav 02</td>
<td>Safe</td>
<td>4</td>
<td>16</td>
<td>7</td>
<td>Yes</td>
<td>52</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Nav 03</td>
<td>Safe</td>
<td>4</td>
<td>9</td>
<td>7</td>
<td>Yes</td>
<td>58</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Nav 04</td>
<td>Safe</td>
<td>4</td>
<td>33</td>
<td>31</td>
<td>Yes</td>
<td>58</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Nav 05</td>
<td>Safe</td>
<td>4</td>
<td>9</td>
<td>8</td>
<td>Yes</td>
<td>&lt; 1</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- HARE performs much better than SpaceEx, but slightly worse than PHAVer in terms of running time; HSolver always does worse than all other tools
- HARE proves safety on many more instances than SpaceEx, PHAVer
Compositional Analysis Using HARE

SATS: Small Airport Transportation Systems

- A new concepts where pilots interact with an automated centralized Airport Management Module (AMM) without ground controller
- Increase access to small airports with multiple landings and departures at the same time.
- Zones - holding, base, lateral entry, runway ...
- Flight rules - entry rules (vertical/lateral), descend, approach, landing ...
- AMM provides entry clearances, missed approach holding fixes, leader aircraft, ...

Safety concern: Maintain minimum separation
Compositional Analysis Using HARE

SATS: Small Airport Transportation Systems

- Each aircraft modelled as a rectangular hybrid automaton
- Abstractions involved variable dropping and scaling, location merging
- Abstract each hybrid automaton into a simpler one
- Decompose the counter-example and refine each of the abstractions
Compositional Analysis Using HARE

SATS: Small Airport Transportation Systems

- Each aircraft modelled as a rectangular hybrid automaton
- Abstractions involved variable dropping and scaling, location merging
- Abstract each hybrid automaton into a simpler one
- Decompose the counter-example and refine each of the abstractions

<table>
<thead>
<tr>
<th>Benchmark/Model</th>
<th>Concrete size (modes, variables)</th>
<th>Abstract size (modes, variables)</th>
<th>Iterations</th>
<th>Time taken (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SATS</td>
<td>(100000, 5)</td>
<td>(3000, 5)</td>
<td>3</td>
<td>155.32</td>
</tr>
</tbody>
</table>
Conclusion & Future Work
Conclusion

- Time approximations are crucial for compositional approximation
- Bounded error approximation
- Software Tool: BEAVER (Bounded Error Approximation based VERification)
- Hybridization based Counter-example guided abstraction refinement
- Software Tool: HARE (Hybrid Abstraction Refinement Engine)
Conclusion & Future Directions

**Conclusion**

- Time approximations are crucial for compositional approximation
- Bounded error approximation
- Software Tool: BEAVER (Bounded Error Approximation based VERification)
- Hybridization based Counter-example guided abstraction refinement
- Software Tool: HARE (Hybrid Abstraction Refinement Engine)

**Current and Future Work**

- Extension to hybrid systems with more complex dynamics
- Non-linear hybrid systems, more complex interactions
- **Compositional Synthesis**
  - Generate multi-robot path plans compositionally from specifications
- **Compositional approaches for verifying robustness properties**
- Stability verification