High-Level Modeling and Analysis of TCAS

Carl Livadas

Theory of Distributed Systems Group
Laboratory for Computer Science
MIT

Joint work with: John Lygeros and Nancy A. Lynch
Outline of Talk

I. Introduction

II. Hybrid Input/Output Automata (HIOA)

III. TCAS Model

IV. Results So Far

V. Further Research

VI. Conclusions
I. Introduction

**Goal:** Model, analyze, and design hybrid systems.

- Develop abstract system models.
- Prove high-level theorems about system safety.
- Refine models.
- Use refinement to infer lower-level theorems.

**Framework:** Input/Output Automaton models

- Precise notion of behavior at component interfaces.
- Invariant assertion proofs.
- Compositional reasoning.
- Levels of abstraction (simulation proofs).

- Asynchronous [Lynch, Tuttle]
  - Timed [Lynch, Vaandrager]
  - Hybrid [Lynch, Segala, Vaandrager, Weinberg]
Hybrid System Case Studies

Hybrid System Benchmark Problems:
- Railroad Crossing Problem [Heitmeyer, Lynch 94]
- Steam Boiler Controller [Leeb, Lynch 95]

Personal Rapid Transit Systems:
- Raytheon’s PRT 2000™
  [Weinberg, Lynch 95], [Weinberg, Lynch, Delisle 96], [Weinberg 96]
  [Livadas 97], [Livadas, Lynch 98]

Automated Highway Systems:
- California PATH Project
  [Branicky, Dolginova, Lynch 96], [Dolginova, Lynch 97], [Dolginova 98]

Air Traffic Management Systems:
- Traffic Alert and Collision Avoidance System (TCAS)
  [Lygeros, Lynch 97], [Livadas, Lynch, Lygeros 99]
Traffic Alert and Collision Avoidance System (TCAS)

- On board conflict detection/resolution system
- Safety critical but very complex
- Humans in the loop
- Developed through a series of revisions:
  - TCAS I: Traffic Advisories (TAs) (implemented, 1983)
  - TCAS II-6: TAs and vertical Resolution Advisories (RAs) (implemented, 1990)
  - TCAS II-7: TAs and vertical RAs with reversals (developed, 1997)
  - TCAS III: TAs and horizontal/vertical RAs (abandoned)
  - TCAS IV: TAs and horizontal/vertical RAs using GPS (conceptual)
More about TCAS II-7

TCAS II-7 uses:

- Range and range rate data (radar).
- Own and intruder altitude data (Mode_S transponder).
- Simple aircraft dynamics ($\ddot{x} = u$).
- Simple intruder intent assumptions ($u = 0$).

To:

- Alert pilot to potential collision threats.
- Determine maneuver to resolve conflict.
  - Sufficient vertical separation at closest approach.
  - “Climb” or “Descend” only.
- Issue resolution advisories to pilot.
  - Consistent for TCAS-equipped aircraft.
  - Can reverse resolution advisories (once).
Validation for TCAS

Extensive simulation:
- MIT Lincoln Labs [Drumm 97–98]
  - Provides no absolute guarantees.
  - Number of test cases very large (approx. 2 million).
  - Changes involve extensive regression testing.
  - Hard to provide conditional guarantees.

Software specification methods:
- StateCharts [Harel 97]
- RSML [Leveson, Heimdahl, Hildreth, Reese 94, 96]
- SpecTRM-RL [Leveson, Heimdahl, Reese, Brown 98, 99]
- Intent Specifications [Leveson 98]

Probabilistic modeling:
- [Kuchar 95]

Drawbacks of these approaches:
- Primarily software (not hybrid).
- Complete, detailed, low-level.
- Validation: Static checking, e.g., all cases covered.
What Else is Needed?

Better models for TCAS systems:
- Include continuous and discrete system components.
- Model and analyze systems at higher level of abstraction.
  - Capture intuitions of software designers, application users.
  - Suitable for informal communication/documentation.
  - Suitable for formal reasoning.
- Allow nondeterminism, incorporate uncertainty.
- Use formal notions of composition and refinement.
  - Support successive model refinement down to code level.
  - Connect with lower-level verification approaches.

Short-Term Goals for TCAS:
- Safety guarantees (Conditional but absolute).
- Qualitative evaluation of system limitations.
- Propose suggestions for the design of TCAS IV.

Long-Term Goals for TDS:
- Develop practical verification approach for hybrid systems:
  - Improve the hybrid system model and proof methods.
  - Develop practical tools for designing hybrid systems.
II. Hybrid Input/Output Automata
[Lynch, Segala, Vaandrager, Weinberg 96, 99]

\[ A = (U, X, Y, \Sigma^{in}, \Sigma^{int}, \Sigma^{out}, \Theta, D, W) : \]

- **Variables** \( V \)
  (Input \( U \)/Internal \( X \)/Output \( Y \); External \( E = U \cup Y \))
- **Discrete Actions** \( \Sigma \)
  (Input \( \Sigma^{in} \)/Internal \( \Sigma^{int} \)/Output \( \Sigma^{out} \))
- **Initial States** \( \Theta \)
- **Discrete Transitions** \( D \) (discrete behavior)
- **Trajectories** \( W \) (continuous behavior)

**Axioms:** input-enabling, closure of trajectories, *etc.*

**Execution**
- \( \alpha = w_0a_1w_1a_2w_2 \cdots \), where \( \alpha.fstate \in \Theta, w_i \in W, \) and \( a_i \in \Sigma \)
- **Admissible:** \( \alpha.ltime = \infty \)

**Traces** (external behavior)
- \( \gamma = (w_0 \downarrow E)vis(a_1)(w_1 \downarrow E)vis(a_2)(w_2 \downarrow E) \cdots \)

**HIOA model admits:**
- Parallel composition
- Successive refinement (levels of abstraction)
Composition of HIOAs

Let $A_1$ and $A_2$ be two compatible HIOA. The composition of $A_1$ and $A_2$ is defined as $A_1 \times A_2 = (U, X, Y, \Sigma^{in}, \Sigma^{int}, \Sigma^{out}, \Theta, \mathcal{D}, \mathcal{W})$, where:

- $X = X_1 \cup X_2$
  $Y = Y_1 \cup Y_2$
  $U = (U_1 \cup U_2) - (Y_1 \cup Y_2)$

- $\Sigma^{int} = \Sigma_1^{int} \cup \Sigma_2^{int}$
  $\Sigma^{out} = \Sigma_1^{out} \cup \Sigma_2^{out}$
  $\Sigma^{in} = (\Sigma_1^{in} \cup \Sigma_2^{in}) - (\Sigma_1^{out} \cup \Sigma_2^{out})$

- $s \in \Theta$ iff $s[V_1 \in \Theta_1 \land s[V_2 \in \Theta_2$

- $(s, a, s') \in \mathcal{D}$ iff $s[V_1 \xrightarrow{\pi_{A_1}(a)} V_1] \land s[V_2 \xrightarrow{\pi_{A_2}(a)} V_2]$

- $w \in \mathcal{W}$ iff $w \downarrow V_1 \in W_1 \land w \downarrow V_2 \in W_2$
Levels of Abstraction/Simulation Relations

Simulation relation from $A$ to $B$ (same external signature):

1. If $r \in \Theta_A$, then $B$ has start state $s$ with $r R s$.

2. If $r R s$ and $r \xrightarrow{a} r'$ then $B$ has a finite execution fragment $\alpha$ from $s$ to $s'$, with the same trace, and with $r' R s'$.

3. If $r R s$ and $w$ is a trajectory of $A$ from $r$ to $r'$ then $B$ has finite execution fragment $\alpha$ from $s$ to $s'$, with the same trace, and with $r' R s'$.

Theorem.

If there is a simulation from $A$ to $B$, then $\text{traces}(A) \subseteq \text{traces}(B)$. 
III. TCAS Model

The diagram illustrates the TCAS Model, which involves aircraft, pilots, sensors, conflict detection, conflict resolution, channels, and advisories. The model shows the interaction between aircraft $i$ and $j$, pilots $i$ and $j$, sensors $i$ and $j$, conflict detection and resolution processes, and communication channels $i,j$ and $j,i$. The advisories are exchanged between the aircraft to mitigate potential conflicts.
TCAS Components

Aircraft Automaton $A_i$
- Variables representing transponder number and equipment.
- Simple aircraft dynamics.

Sensor Automaton $S_i$
- Senses available signals and schedules conflict resolutions.

Conflict Detection Automaton $D_i$
- Senses available information and declares/undeclares threats.

Conflict Resolution Automaton $R_i$
- Resolves threats.

Communication Channel Automaton $C_{ij}$
- Sends messages between aircraft.

Pilot Automaton $P_i$
- Maintains velocity and acceleration within acceptable bounds.
- Follows/ignores advisories within allotted time.
Aircraft Automaton $A_i$

Variables:
- **Input:** $a_i \in \mathbb{R}^3$, aircraft acceleration
- **Output:** $Mode_S_i \in \mathbb{N}$, unique transponder number
  - $Equipment_i \in \{\text{None, Report, TCAS}\}$
  - $p_i \in \mathbb{R}^3$, position
  - $v_i \in \mathbb{R}^3$, velocity

Discrete Actions:
- None.

Trajectories:
- Input variables arbitrary.
- $Mode_S_i$, $Equipment_i$ constant.

\[
\begin{bmatrix}
\dot{p}_i(t) \\
\dot{v}_i(t)
\end{bmatrix} = \begin{bmatrix}
v_i(t) \\
a_i(t)
\end{bmatrix}
\]
Sensor Automaton $S_i$

Variables:
- **Input:** $p_j, v_j$, for all $j$
- **Output:** $R_{ij} \in \mathbb{R}^+, \dot{R}_{ij} \in \mathbb{R}$, for all other $j$, range and range rate
  $h_{ij} \in \mathbb{R}^+, \dot{h}_{ij} \in \mathbb{R}$, for all $j$, relative altitude and vertical velocity
- **Internal:** A timer.

Actions:
- **Output:** $Sample_i$

Trajectories:
- Input variables arbitrary.
- Output variables constant, calculated from inputs.
- Trajectories stop when $Sample_i$ action enabled (after time $T$).

Discrete Transitions:
- $Sample_i$ occurs every $T$ seconds.
**Conflict Detection Automaton** $D_i$

**Variables:**

- **Input:** $R_{ij}, \dot{R}_{ij}$, for all other $j$
  $h_{ij}, \dot{h}_{ij}$, for all $j$

- **Derived:** $\text{Range\_Test}_{ij} \in \text{Bool}$, for all other $j$
  $\text{Altitude\_Test}_{ij} \in \text{Bool}$, for all other $j$

**Actions:**

- **Output:** $\text{Declare}_{ij}$, for all other $j$
  $\text{Undeclare}_{ij}$, for all other $j$

**Trajectories:**

- Input variables arbitrary.
- Trajectories stop when any output action enabled.

**Discrete Transitions:**

- $\text{Declare}_{ij}$ occurs when aircraft $j$ poses a threat (satisfies range, altitude tests).
- $\text{Undeclare}_{ij}$ occurs when aircraft $j$ moves out of range (fails range test).

Details of range, altitude tests extracted from TCAS specifications.
Conflict Resolution Automaton $R_i$

Variables:
- **Input:** $h_{ij}, \dot{h}_{ij}, R_{ij}, \dot{R}_{ij}, \text{Mode}_S_j, \text{Equipment}_j$
- **Output:** $\text{Sense}_i \in \{\text{Climb, } \perp, \text{Descend}\}$, initially $\perp$
  - $\text{Strength}_i \in \{2500, 1500, 0, -500, -1000, -2000\}$ (units of $ft/min$)

Actions:
- **Input:** $\text{Sample}_i$
  - $\text{Declare}_{ij}, \text{Undeclare}_{ij}$
  - $\text{Receive}_{ji}(\text{dir})$
- **Output:** $\text{Send}_{ij}(\text{dir})$

Discrete Transitions:
- Sense selected initially (upon $\text{Declare}_{ij}$ or $\text{Receive}_{ij}(\text{dir})$), may be reversed later (upon $\text{Receive}_{ij}(\text{dir})$ or $\text{Sample}_i$).
- Sense changes communicated to intruder ($\text{Send}_{ij}(\text{dir})$).
- After $\text{Declare}_{ij}$, strength selected upon every $\text{Sample}_i$.
- $\text{Sense}_i, \text{Strength}_i$ told to the pilot.
- Conflict ends for $j$ upon $\text{Undeclare}_{ij}$.
Sense Selection \([N = 2]\)

- Initial selection upon \(Declare_{ij}\):
  - If only one sense OK, choose it.
  - If both senses OK, choose non-crossing.
  - If neither sense OK, choose the one that leads to greater separation.

- Initial selection upon \(Receive_{ij}(\text{dir})\):
  - If \(Mode_{S_i} > Mode_{S_j}\) comply.
  - If \(Mode_{S_i} < Mode_{S_j}\) \(\land\) dir non-crossing, comply.
  - If \(Mode_{S_i} < Mode_{S_j}\) \(\land\) dir crossing, choose as above.

- Subsequent sense selection upon \(Receive_{ij}(\text{dir})\):
  - If \(Mode_{S_i} > Mode_{S_j}\) comply.
  - If \(Mode_{S_i} < Mode_{S_j}\) ignore.

- Subsequent sense selection upon \(Sample_i\):
  - If \(Mode_{S_i} > Mode_{S_j}\) do nothing.
  - If \(Mode_{S_i} < Mode_{S_j}\) reverse if current sense not OK and reversed sense OK and non-crossing.
  - Reverse at most once.
Communication Channel and Pilot Automata

Communication Channel Automaton $C_{ij}$: Messages from $i$ to $j$
- Input Action: $Send_{ij}$
- Output Action: $Receive_{ji}$
- All messages delivered, in order, within time $d_c$.

Pilot Automaton $P_i$:
- Generates $a_i \in [a_x, \bar{a}_x] \times [a_y, \bar{a}_y] \times [a_z, \bar{a}_z]$.
- Maintains vertical velocity in $[\bar{v}_z, \bar{v}_z]$.
- May follow or ignore an advisory ($Follow_i$).
- If $Follow_i = \text{False}$, may apply any allowable $a_i$.
- If $Follow_i = \text{True}$, responds within time $d_p$ by applying $a_{zi} = \sigma a$ until $\sigma \dot{h}_{ii} \geq \text{Strength}_i$, where $\sigma = 1$, if $Sense_i = \text{Climb}$, and $\sigma = -1$, otherwise. Before then, apply any allowable $a_i$. 
IV. Results So Far

Verification Approach:

- Start with simple/idealized cases.
  - Only two aircraft, both TCAS equipped, limited uncertainty,...
- Consider various execution assumptions.
  - Both pilots follow advisories, within certain time.
  - One follows, one keeps going as before,...
- Obtain conditional claims, conditioned on execution assumptions.
- Gradually add complexity.

Verification Goal:

Theorem. *TCAS ensures that aircraft maintain sufficient separation.*
Proof Outline

Proof Approach:

1. Define “well-behaved aircraft system”, WBS.
2. Classify WBS executions into three categories.
   - Non-crossing
   - Crossing
   - Reversing
3. Prove safety for each execution category.
4. Combine per-category safety results to obtain overall safety results.

Proof Goal:

**Theorem.** If safety condition $P$ holds when a conflict is declared then TCAS ensures that “well-behaved” aircraft maintain sufficient separation.
Well-Behaved Aircraft System (WBS)

- Only two aircraft.
- Aircraft 1 has higher priority.
- Communication channels $C_{12}$ and $C_{21}$ are FIFO.
- Both aircraft TCAS equipped.
- Both pilots follow RAs.
- No sensor uncertainty.
- Use nominal strength only.
- Constant horizontal velocity.
- Pilots do not oppose RAs.
- Sufficient time to implement RAs.
- When threat is first detected, at least one sense appears to be ok.
Well-Behaved Aircraft System (WBS)

- Only two aircraft.
- Aircraft 1 has higher priority: $\text{Mode}_1 < \text{Mode}_2$.
- Communication channels $C_{12}$ and $C_{21}$ are FIFO.
- Both aircraft TCAS equipped: $\text{Equipment}_1 = \text{Equipment}_2 = \text{TCAS}$.
- Both pilots follow RAs: $\text{Follow}_1 = \text{Follow}_2 = \text{True}$.
- No sensor uncertainty.
- Use nominal strength only: $\text{Strength}_1 = \text{Strength}_2 = 1500$.
- Constant horizontal velocity: $a_x = \bar{a}_x = a_y = \bar{a}_y = 0$.
- Pilots do not oppose RAs: $\text{Sense}_i \neq \bot \implies \sigma_i a_{zi} \geq 0$, where $\sigma_i = 1$, if $\text{Sense}_i = \text{Climb}$, and $\sigma_i = -1$, otherwise.
Well-Behaved Aircraft System (WBS), cont’d

- Sufficient time to implement RA: Initially, \( T > 2d_c + d_p \).
- When threat is first detected, at least one sense appears OK: Initially \( SEP_{ij}(\sigma) \geq ALIM \), for some \( \sigma \in \{-1, 1\} \).

\[ T, \text{ Time to closest horizontal approach:} \quad T = -\frac{\Delta x \Delta v_x + \Delta y \Delta v_y}{\Delta v_x^2 + \Delta v_y^2} \]

\[ \tau, \text{ Estimated time to closest approach:} \quad \tau = -\frac{R_{ij}}{\min\{\dot{R}_{ij}, -10\}} \]

\( SEP_{ij}, \text{ Estimated vertical separation at closest approach:} \)

\[ SEP_{ij}(\sigma) = \begin{cases} 
\sigma [\Delta z_{ij} + \Delta v_{zij}\tau], & \text{if } (\tau \leq d) \lor (\sigma v_{zi} \geq 1500), \\
\sigma [\Delta z_{ij} + \Delta v_{zij}d + (1500\sigma - v_{zj})(\tau - d)], & \text{else.} 
\end{cases} \]

where \( \sigma \in \{-1, 1\} \), sense of aircraft \( i \) and \( \dot{R}_{ij} \in \mathbb{R}^+ \), RA implementation delay assumed by TCAS.
**WBS Execution Categories**

**Non_Crossing_Execs**
- TCAS protocol invoked
- Non-crossing RA issued initially
- Non-crossing RA maintained throughout conflict

**Crossing_Execs**
- TCAS protocol invoked
- Crossing RA issued initially
- Crossing RA maintained throughout conflict

**Reversing_Execs**
- TCAS protocol invoked
- Crossing RA issued initially
- Non-crossing RA issued subsequently
- Non-crossing RA maintained thereafter throughout conflict
**Per-Category Results**

**Definition.** *SafeExecs* is the set of executions of WBS in which the aircraft are sufficiently separated in altitude (ALIM) at closest horizontal approach.

**Lemma 1.**
If $\alpha \in \text{Non\_Crossing\_Execs}$ and $P_{NC}$ holds initially, then $\alpha \in \text{Safe\_Execs}$.

**Lemma 2.**
If $\alpha \in \text{Crossing\_Execs}$ and $P_{C}$ holds initially, then $\alpha \in \text{Safe\_Execs}$.

**Lemma 3.**
If $\alpha \in \text{Reversing\_Execs}$ and $P_{R}$ holds initially, then $\alpha \in \text{Safe\_Execs}$.
Lemma 1: Safety of Non-Crossing Executions

\( P_{NC} \):
\( S \geq ALIM \), where \( S \) = projected smallest altitude separation at time \( T \) (time of closest horizontal approach), based on non-crossing RA, nominal strength.

Calculating \( S \):
- Initial altitude separation, minus
- Potential loss in altitude separation due to RA implementation delay \( D_{NC} \), plus
- Altitude separation gained by following nominal strength non-crossing RA from when it is implemented until \( T \).

\[
S = |\Delta z| + V_{NC}D_{NC} + 3000(T - D_{NC}),
\]
where
\[
\sigma = \text{sign}(z_1 - z_2),
\]
\[
D_{NC} = 2d_c + d_p, \quad \text{and}
\]
\[
V_{NC} = \max(v_z, \sigma v_{z1} - \bar{a}_zd_c) + \max(v_z, -\sigma v_{z2} - 2\bar{a}_zd_c).
\]
Lemma 2: Safety of Crossing Executions

$P_C$:
$S \geq \text{ALIM}$, where $S = \text{projected smallest altitude separation at time } T$, based on crossing RA, nominal strength.

Calculating $S$:
- Initial altitude separation, minus
- Potential loss in altitude separation due to RA implementation delay $D_C$, plus
- Altitude separation gained by following nominal strength crossing RA from when it is implemented until $T$.

$S = -|\Delta z| + V_C D_C + 3000(T - D_C)$, where
$\sigma = \text{sign}(z_1 - z_2)$,
$D_C = 2d_c + d_p$, and
$V_C = \max(v_z, -\sigma v_{z1} - \bar{a}_z d_c) + \max(v_z, \sigma v_{z2} - 2\bar{a}_z d_c)$.
Lemma 3: Safety of Reversing Executions, $T_R \leq D_C$

$P_R$:
$S \geq ALIM$, where $S =$ projected smallest altitude separation at $T$, based on reversing just prior to implementing a crossing advisory, then following non-crossing RA, nominal strength.

Calculating $S$:
- Initial altitude separation, minus
- Potential loss in altitude separation due to reversed RA implementation delay $D_C + D_R$, plus
- Altitude separation gained by following nominal strength non-crossing RA from the time of implementing the reversal until $T$.

$S = |\Delta z| + V_R(D_C + D_R) + 3000(T - D_C - D_R)$, where
$\sigma = \text{sign}(z_1 - z_2)$,
$D_R = d_c + d_p$, and
$V_R = \max(v_z, \sigma v_{z1} - \bar{a}_z D_C) + \max(v_z, -\sigma v_{z2} - \bar{a}_z(D_C + d_c))$. 
Lemma 3: Safety of Reversing Executions, $T_R > D_C$

- At time of reversal, both aircraft have implemented the crossing RA, i.e., $-\sigma v_{z1} \geq 1500$ and $\sigma v_{z2} \geq 1500$.

- In order to reverse, crossing RA must not be OK and non-crossing RA must be OK, i.e., $\text{Sep}_{12}(\sigma) < \text{ALIM}$ and $\text{Sep}_{12}(\sigma) \geq \text{ALIM}$.

\[ \implies \sigma \Delta z \geq \text{ALIM} \]
Lemma 3: Safety of Reversing Executions, \( T_R > D_C \)

\( P_R: \)

\( S \geq ALIM, \) where \( S = \) projected smallest altitude separation at \( T, \) based on reversing just prior to the \( \sigma \Delta z = ALIM \) mark, then following non-crossing RA, nominal strength.

**Calculating \( S: \)**

- Initial altitude separation (\( \sigma \Delta z = ALIM \)), minus
- Potential loss in altitude separation due to RA implementation delay \( D_R, \) plus
- Altitude separation gained by following nominal strength non-crossing RA from the time of implementing the reversal until \( T. \)

\[
S = ALIM + V_R D_R + 3000(T - \overline{T}_R - D_R),
\]

where

\[
\sigma = \text{sign}(z_1 - z_2),
\]

\[
D_R = d_c + d_p,
\]

\[
V_R = \max(v_z, \sigma v_{z1} - \overline{a}_z \overline{T}_R) + \max(v_z, -\sigma v_{z2} - \overline{a}_z (\overline{T}_R + d_c)), \quad \text{and}
\]

\[
\overline{T}_R, \text{ upper bound on reversing time, given by:}
\]

\[
-|\Delta z| + V_C D_C + 3000(\overline{T}_R - D_C) = -ALIM.
\]
Overall Safety Results

**Goal**: Find condition $P$ such that:
For any execution $\alpha$ of $WBS$, if $P$ holds initially, then $\alpha \in Safe\_Execs$.

**Sample Results**:

1. **Conjunction of Safety Conditions**
2. **Ruling out Crossing RAs**
3. **Aircraft Close in Altitude**
Overall Safety Results, cont’d

Conjunction of Safety Conditions:

- \( P = P_{NC} \land P_C \land P_R \)
- Simple but conservative.

Ruling out Crossing RAs:

- \( P = P_{NC} \land Crossing\_Impossible \), where
  \[ \sigma = \text{sign}(z_1 - z_2) \], and
  \[ Crossing\_Impossible = (Sep_{12}(\sigma) \geq ALIM \lor Sep_{12}(-\sigma) < ALIM) \]
  \[ \land (Sep_{21}(-\sigma) \geq ALIM \lor Sep_{21}(\sigma) < ALIM) \].
Overall Safety Results, cont’d

**Aircraft Close in Altitude:**

\[ P = P(K) = (S \geq ALIM) \land |\Delta z| \leq K, \]

where \( S \) = projected smallest altitude separation at \( T \), based on reversing at the latest possible time \( T_R \), then following non-crossing RA, nominal strength.

**Calculating \( T_R \):** Latest possible time of reversal, dictated by \( |\Delta z| \leq K \) initially (possibly time of crossing in altitude).

**Calculating \( S \):** Similar to reversing execution case.
V. Future Research

- **Remove Restrictions:**
  1. Uncertainty in sensor values, in aircraft flight dynamics.
  2. Unrestricted horizontal aircraft dynamics.
  3. Variable resolution strengths, not just nominal.
  4. Different assumptions on pilots’ behavior, e.g.:
     - Follow RAs, but with larger delay.
     - One pilot doesn’t follow RAs.
  5. One of the planes not TCAS-equipped.
  6. Multiple aircraft, $N > 2$.

- **Challenge Problem 1:** *(For hybrid systems formal methods community)*
  Thoroughly analyze this problem, and others like it, using formal techniques.

- **Challenge Problem 2:** *(For ATM design/validation community)*
  Incorporate high-level formal modeling/analysis into design process.
VI. Conclusions

Contributions:

• With respect to TCAS:
  – Extracted simplified high-level TCAS model.
  – Obtained safety conditions for TCAS’s execution categories.
  – Combined per-category guarantees into overall safety conditions.

• With respect to formal analysis and verification:
  – Demonstrated use and practicality of high-level modeling and analysis techniques.