

NASA Langley's Formal Methods Research in Support of the Next Generation Air Transportation System

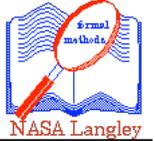
Ricky W. Butler
César Muñoz

<http://shemesh.larc.nasa.gov/fm/>

<http://research.nianet.org/fm-at-nia/>



Notable Formal Methods Projects 2000 - 2008



Concept of Operations

Small Aircraft Transportation System (SATS)
Enhanced Oceanic Operations (EOO)

System-level Requirements

Rockwell Collins: Asynchronous Models of Flight Deck (GALS)
Airborne Information for Lateral Spacing (AILS)
Spacecraft Autonomy and AI Planning

Software/Hardware Requirements

Rockwell Collins: Flight Guidance Systems/Flight Management Systems
Mode Confusion Elimination

Software/Hardware Designs

Scalable Processor-Independent Design for Extended Reliability (SPIDER)
Honeywell/TTTech Full Authority Digital Engine Control (FADEC)
Honeywell DEOS Operating System
Conflict Detection and Resolution (CD&R) Algorithms
Runway Incursion Prevention System (RIPS)

Software/Hardware Implementations

Automatic Code Generation for KB3D
Fixed Structure Neural Networks



This Talk



Concept of Operations

Small Aircraft Transportation System (SATS)

Enhanced Oceanic Operations (EOO)

System-level Requirements

Rockwell Collins: Asynchronous Models of Flight Deck (GALS)

Airborne Information for Lateral Spacing (AILS)

Spacecraft Autonomy and AI Planning

Software/Hardware Requirements

Rockwell Collins: Flight Guidance Systems/Flight Management Systems

Mode Confusion Elimination

Software/Hardware Designs

Scalable Processor-Independent Design for Extended Reliability (SPIDER)

Honeywell/TTTech Full Authority Digital Engine Control (FADEC)

Honeywell DEOS Operating System

Conflict Detection and Resolution (CD&R) Algorithms:

-- **KB3D algorithm**

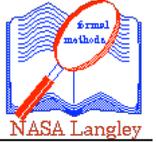
-- **Loss of Separation Extension**

Runway Incursion Prevention System (RIPS)

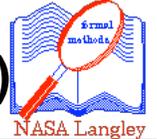
Software/Hardware Implementations

Automatic Code Generation for KB3D

Fixed Structure Neural Networks



SATS



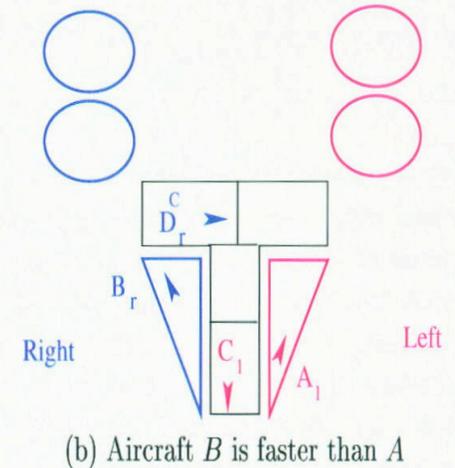
- SATS goal: significantly increase the capacity of regional airports.
- One of the most revolutionary aspects of the SATS approach is the use of a **software system** that will sequence aircraft into the SATS airspace with no air traffic controller present.
- There are **serious safety issues** associated with these software systems and their underlying key algorithms.

Investigated Formal Verification of SATS operational procedures and algorithms using formal methods.

- TEAM: César Muñoz, Víctor Carreño and Gilles Dowek
- Unusual formal methods project in that it was a formalization of a *concept of operations*.



- A formal **finite-state machine model** of the SATS operational procedures (24 transition rules)
- **Exhaustive analysis** of entire state space
- **Six Safety Properties** verified including
 - At most one aircraft cleared at a given fix
 - There is always a Missed Approach Holding Fix for every aircraft
 - No more than 2 aircraft on missed approach for a given fix.
 - Runway incursions do not occur
- Liveness properties verified, (e.g. no deadlocks)



Operational procedures captured in 24 formal transition rules:

3.2.5 Approach Initiation for Lateral Entry (right, left)

The Approach Initiation for Lateral Entry (right) procedure is illustrated in Figure 10. An aircraft in lateral entry is allowed to initiate the approach only if the following conditions hold:

- It is the first aircraft in the sequence or its leader is already on the approach.
- There is at most one aircraft on base at the opposite side.

If one of these conditions is not met, the aircraft must hold at 2000 feet.

This procedure is encoded by the PVS function `LateralApproachInitiation`.

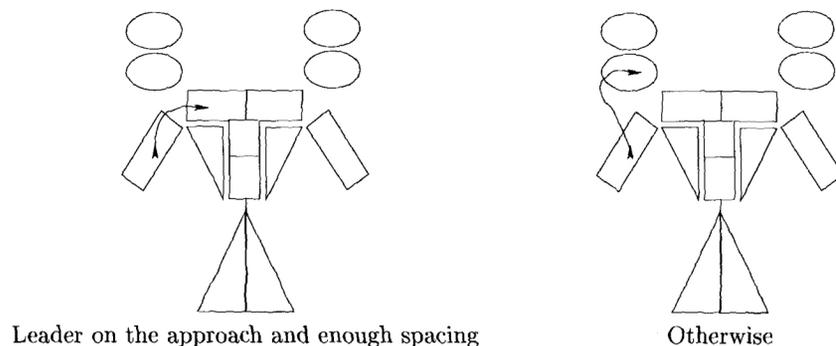
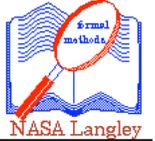


Figure 10: *Approach Initiation for Lateral Entry (right)*

- Nine issues identified via analysis
 - two required changes to the rules of the ConOps,
 - five where implicit or explicit omissions,
 - and two were clarifications.
- All recommendations from FM team adopted by SATS conops team



Example Entry Rule, Vertical, Right

A vertical entry to the right Initial Arrival Fix (IAF-R) is granted if all of the following hold:

- There are less than 2 aircraft either at IAF-R or assigned to IAF-R as a MAHF.
- No aircraft currently on the final approach assigned to IAF-R as a MAHF
- No aircraft executing a missed approach with IAF-R as its MAHF
- No aircraft performing a lateral entry to IAF-R
- No aircraft at IAF-R holding at 3000 feet or transitioning to 2000 feet.



Example Entry Rule in Formal Language



```
VerticalEntry(side) (this) : list[SCA] =
  IF virtual(this,side) < 2 &
    NOT on_approach?(this,side) &
    length(this`maz(side)) = 0 &
    length(this`lez(side)) = 0 &
    length(this`holding3(side)) = 0 THEN
  LET a = aircraft(this,side) IN
  LET next = this WITH [
    `holding3(side) := add(this`holding3(side), a),
    `nextseq        := next(a),
    `nextmahf       := opposite(a`mahf),
    `nextid         := this`nextid+1,
    `rule           := 1*sign(side)
  ] IN
  (: next :)
ELSE
  null
ENDIF
```



SATS Verification: Hybrid Analysis

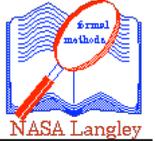


- **Hybrid model** extends the discrete model
- In contrast to the original model, the proposed model enables the verification of safety **spacing** requirements of SATS HVO operations.
- To this end, aircraft performances, such as ground speed ranges, and information about the SCA geometry, such as length of the approach segments, were integrated into the original model.
- Thus, in the hybrid model, the concept of operations is described by the continuous dynamics of aircraft and the discrete events within the SCA.
- Using **theorem proving** and **model checking** techniques, we have exhaustively explored the hybrid model and mechanically verified spacing requirements over all nominal operations.
- The SATS HVO development, excluding the PVS tools Besc, PVSio and ProofLite, is about 2800 lines of PVS specification and lemmas and 6500 lines of proofs.

Hybrid Verification of an Air Traffic Operational Concept, César Muñoz and Gilles Dowek, IEEE ISoLA Workshop on Leveraging Applications of Formal Methods, Verification, and Validation, BibTex Reference, 2005.



SATS: Using Implicit Intent Information

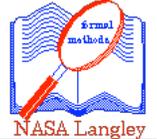


- The intent based conflict detection method described in this paper does not make use of information exchange to calculate the intended aircraft trajectory.
- It infers the aircraft trajectory from established data such as published routes and published approaches. We have called this *implicit* intent conflict detection.
- Implicit intent eliminates the need and associated cost, complexity, and communication bandwidth of the data link used in explicit intent conflict detection.

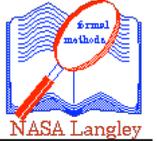
Implicit Intent Information for Conflict Detection and Alerting, Víctor Carreño and César Muñoz, Proceedings of the 23rd Digital Avionics Systems Conference, DASC 2004, BibTeX Reference, 2004.



See <http://research.nianet.org/fm-at-nia/SATS/>

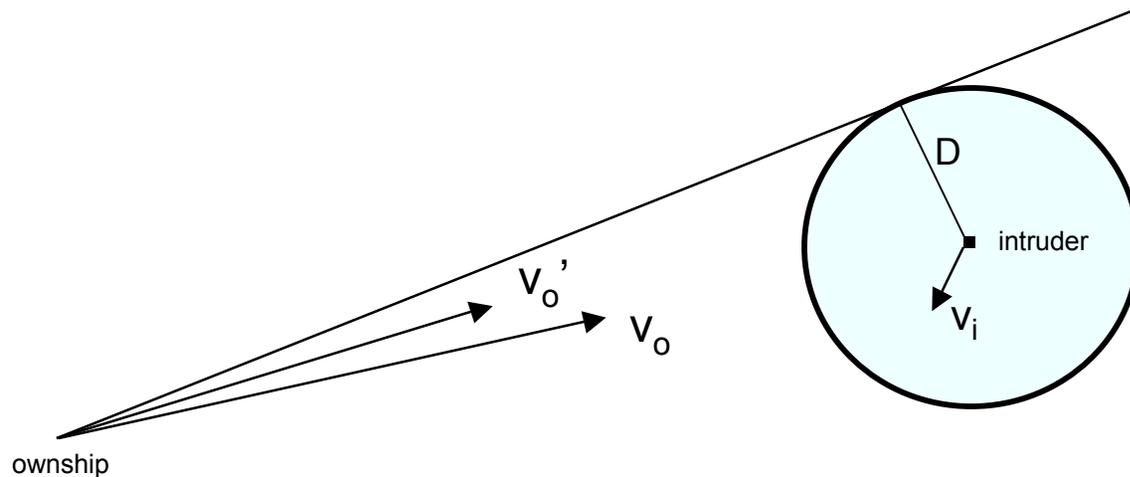


- Formal Analysis of the Operational Concept for the Small Aircraft Transportation System, César Muñoz, Víctor Carreño and Gilles Dowek, Rigorous Engineering of Fault-Tolerant Systems, BibTex Reference, 2006.
- Safety Verification of the Small Aircraft Transportation System Concept of Operations, Víctor Carreño and César Muñoz, AIAA 5th Aviation, Technology, Integration, and Operations Conference, BibTex Reference, 2005.
- Conflict Prevention and Separation Assurance Method in the Small Aircraft Transportation System, Maria Consiglio, Víctor Carreño, Daniel Williams, and César Muñoz, AIAA 5th Aviation, Technology, Integration, and Operations Conference, BibTex Reference, 2005.
- Hybrid Verification of an Air Traffic Operational Concept, César Muñoz and Gilles Dowek, IEEE ISoLA Workshop on Leveraging Applications of Formal Methods, Verification, and Validation, BibTex Reference, 2005.
- Implicit Intent Information for Conflict Detection and Alerting, Víctor Carreño and César Muñoz, Proceedings of the 23rd Digital Avionics Systems Conference, DASC 2004, BibTeX Reference, 2004.
- Conflict Detection and Alerting in a Self Controlled Terminal Area, Maria Consiglio, César Muñoz, and Victor Carreño, Proceedings of the 24th Congress of International Council of Aeronautical Sciences, ICAS 2004, BibTex Reference, 2004.
- Modeling and Verification of an Air Traffic Concept of Operations, César Muñoz, Gilles Dowek, and Victor Carreño, Proceedings of the International Symposium on Software Testing and Analysis, ISTTA 2004, BibTex Reference, 2004.
- Abstract Model of the SATS Concept of Operations: Initial Results and Recommendations, Gilles Dowek, César Muñoz, and Victor Carreño, NASA/TM-2004-213006, BibTex Reference, 2004.



CD&R: KB3D

Contributors: César Muñoz, Alfons Geser, Gilles Dowek, Víctor Carreño, Radu Siminiceanu, Jeffrey Maddalon, André Galdino, Mauricio Ayala and Ricky Butler

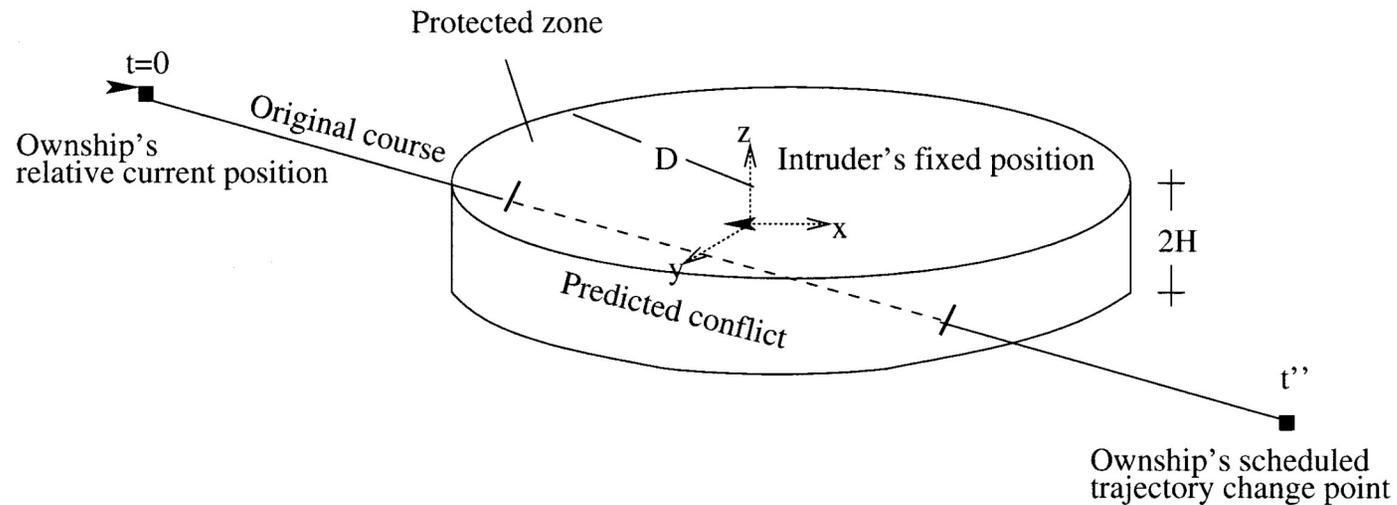
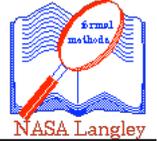


Today the primary responsibility for aircraft separation is borne by the air traffic controller.

- Current method is based on human-factors oriented experimentation with high fidelity simulations.
- But as software takes on more and more responsibility for detecting potential conflicts and recommending or executing the evasive maneuvers, we will need additional methods to guarantee safety of software.
- **The correctness of the algorithm must be established for all possible situations.**
- Simulation and testing cannot accomplish this.



Axes Translation To Facilitate Analysis



Position and velocity translation of axes

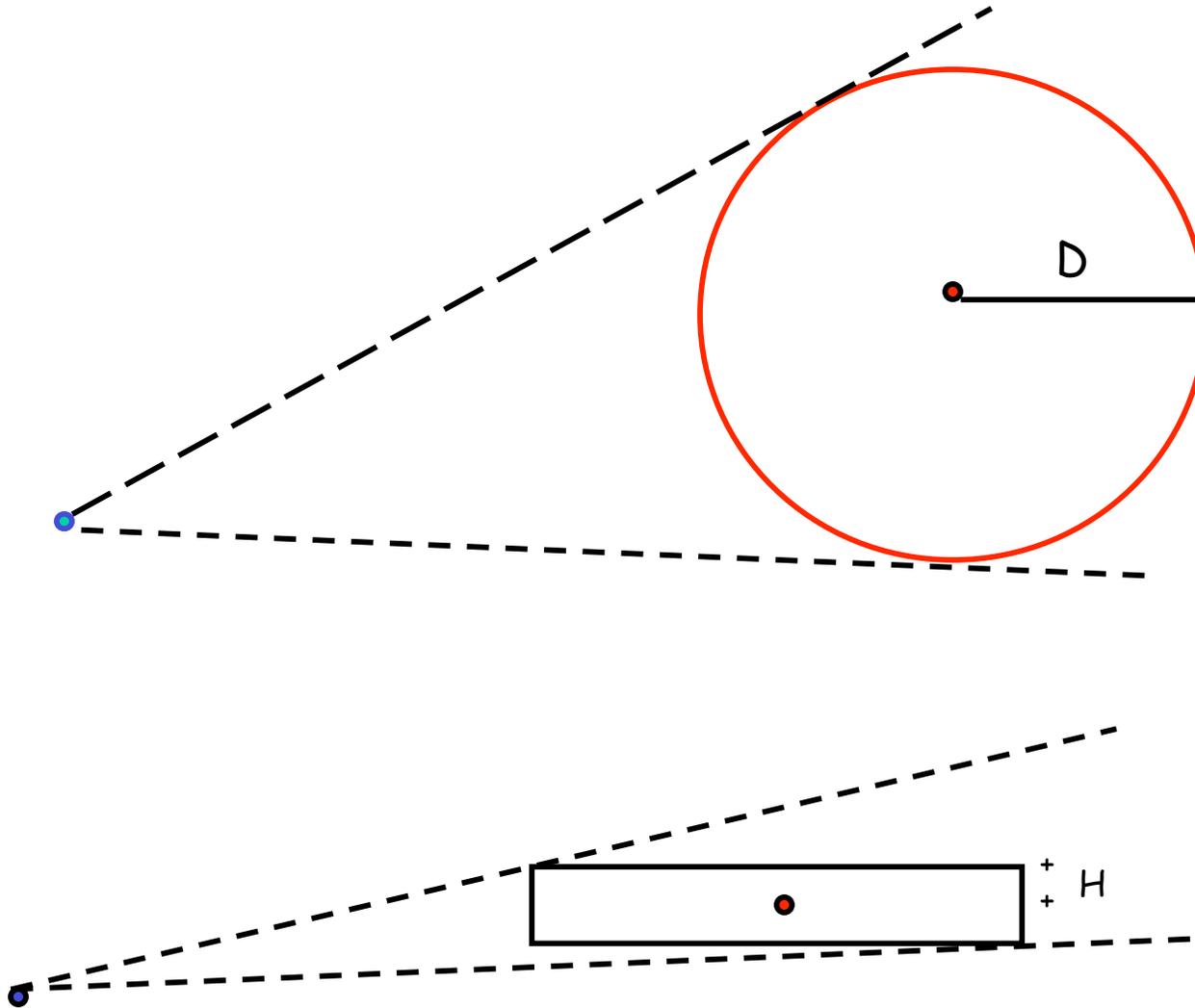
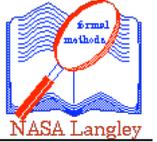
$$\vec{s} = (s_x, s_y, s_z) = \vec{s}_o - \vec{s}_i$$

$$\vec{v} = (v_x, v_y, v_z) = \vec{v}_o - \vec{v}_i$$

Of course, one must translate results back to original axes in implementation code (easy to do).



KB3D: Horizontal/Vertical Solutions (Translated Frame)





Resolution Maneuvers



The KB3D algorithm

- Is a generalization of Karl Bilimoria's CD&R algorithm (used in FACET) to 3 dimensions
- generates maneuvers where only one of **vertical speed**, **ground speed**, or **heading** are changed. (Easier for Pilot to Fly)



Let $\mathbf{v}'_o = (v'_{ox}, v'_{oy}, v'_{oz})$ be the resolution velocity vector for the own

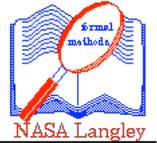


Let $\mathbf{v}_o = (v_{ox}, v_{oy}, v_{oz})$ be its original velocity vector

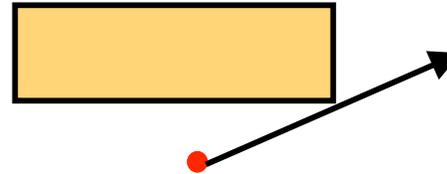
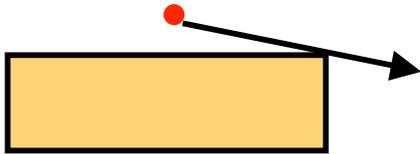
- Vertical Speed Only $v'_{ox} = v_{ox}$, $v'_{oy} = v_{oy}$
- Ground Speed Only $v'_{ox} = k v_{ox}$, $v'_{oy} = k v_{oy}$, $v'_{oz} = v'_{oz}$
- Heading Only $v'^2_{ox} + v'^2_{oy} = v^2_{ox} + v^2_{oy}$, $v'_{oz} = v'_{oz}$



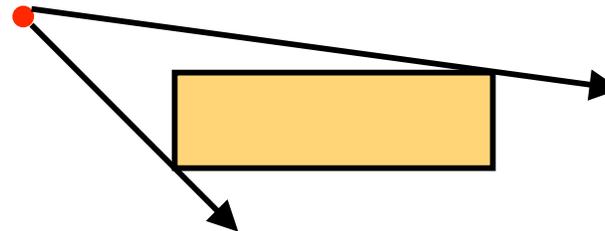
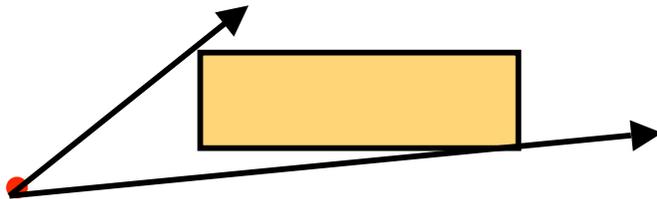
The Vertical Solutions



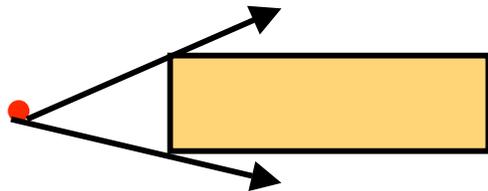
IF not horizontally separated THEN



ELSE IF $|s_z| \geq H$

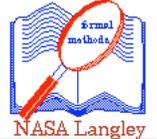


ELSE





A Component of the KB3D Algorithm



```
break_v_symm(s:Vect3) : Sign =
  IF s`x < 0 OR (s`x = 0 AND s`y < 0) THEN 1 ELSE -1  ENDIF

kb3d_vertical_speed_vz(s:Vect3,vo:Vect3,vi:Vect3| precondition?(s)(vo-vi) AND NOT on_line?(s)): real =
  IF sq(x(vo-vi)) + sq(y(vo-vi)) = 0 THEN          %% Relative ground speed is zero
    vi`z
  ELSIF sq(s`x) + sq(s`y) <= sq(D) THEN          %% inside horizontal zone
    vertical_THETA2(s,vo,vi)
  ELSIF abs(s`z) >= H THEN                          %% outside horizontally AND vertically
    LET v1 = vertical_THETA1(s,vo,vi,-sign(s`z)) IN
    LET v2 = vertical_THETA2(s,vo,vi) IN
    IF abs(vo`z-v1) < abs(vo`z-v2) THEN
      v1
    ELSE
      v2
    ENDIF
  ELSE                                             %% outside horizontally, inside vertically
    LET v1 = vertical_THETA1(s,vo,vi,-break_v_symm(s)) IN
    LET v2 = vertical_THETA1(s,vo,vi,break_v_symm(s)) IN
    IF abs(vo`z-v1) < abs(vo`z-v2) THEN
      v1
    ELSE
      v2
    ENDIF
  ENDIF
ENDIF
```



Example Correctness Theorem: Vertical



Aircraft : TYPE = [# s : Vect3, v : Vect3 #]

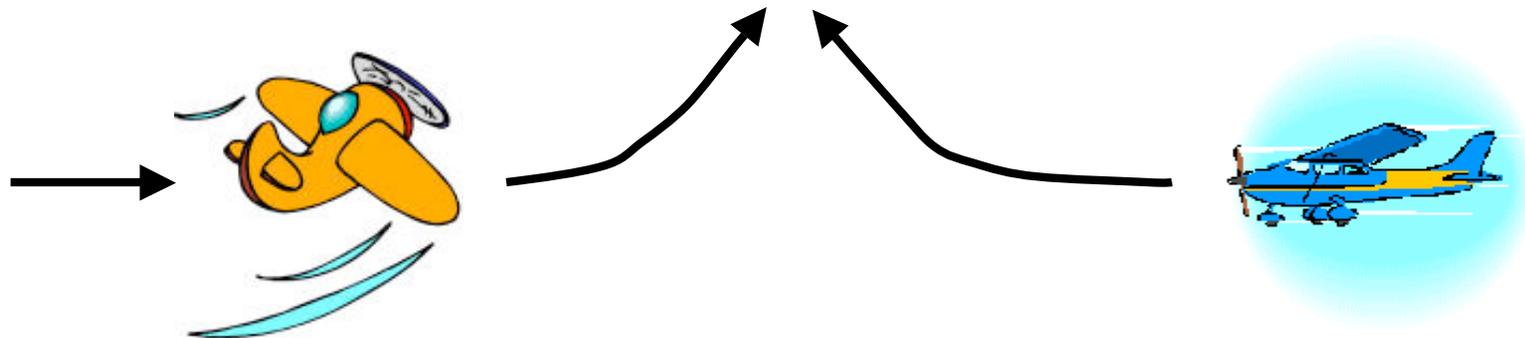
a,b: VAR Aircraft

```
kb3d_vertical_speed_correctness : THEOREM
  precondition?(a`s-b`s)(a`v-b`v) AND
  NOT on_line?(a`s-b`s) AND
  sq(a`v`x)+sq(a`v`y) /= 0 IMPLIES
  LET nva = kb3d_vertical_speed(a,b) IN
  NOT predicted_conflict?(a`s-b`s,nva-b`v)
```

where

```
predicted_conflict?(s,v):bool =
  EXISTS (t:nnreal) : NOT abs(s`z+t*v`z) >= H      AND
  NOT (s+t*v)*(s+t*v) >= sq(D)
```

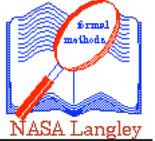
Don't Want:



- For two aircraft executing the CD&R algorithm, **prove**
- Recommended/executed trajectories are always in opposite directions
 - In a perfectly symmetric case, there is a symmetry breaking mechanism



Cooperative Theorem



Aircraft : TYPE = [# s : Vect3, v : Vect3 #]
a,b: VAR Aircraft

```
cooperative_kb3d_vertical_speed : THEOREM
  precondition?(a`s-b`s)(a`v-b`v) AND
  ...
  IMPLIES
    LET nva = kb3d_vertical_speed(a,b) IN
    LET nvb = kb3d_vertical_speed(b,a) IN
    NOT predicted_conflict?(a`s-b`s,nva-nvb)
```

where

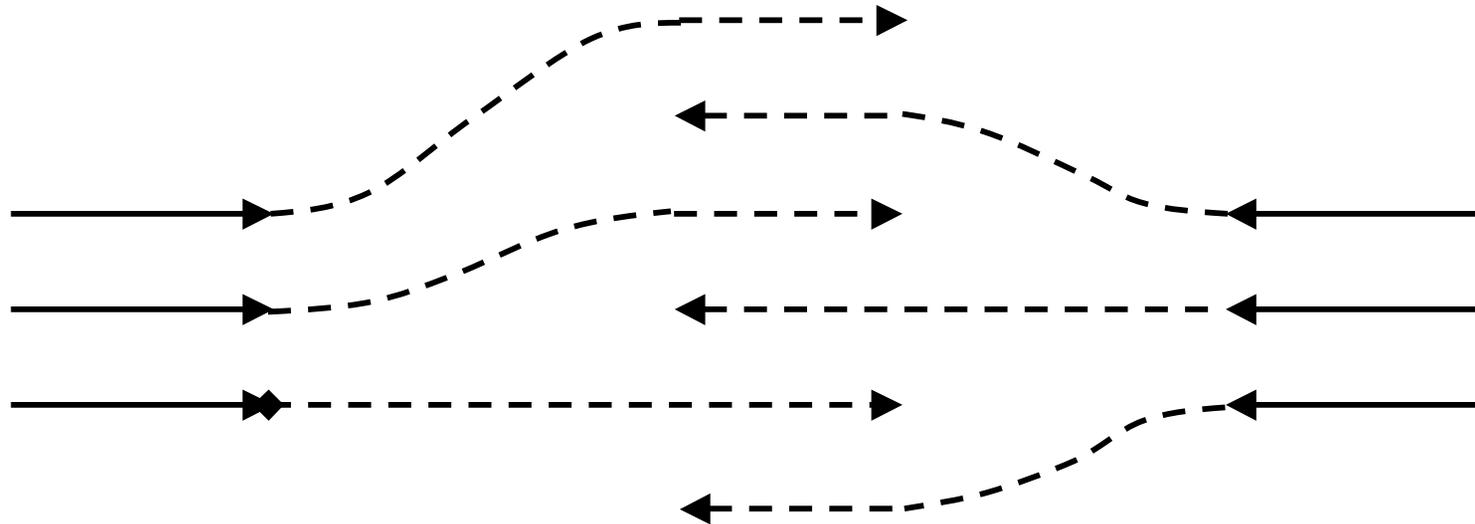
```
predicted_conflict?(s,v):bool =
  EXISTS (t:nnreal) : NOT abs(s`z+t*v`z) >= H      AND
  NOT (s+t*v)*(s+t*v) >= sq(D)
```



N Aircraft: Collaborative Properties



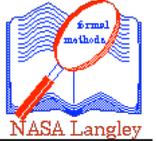
- For N aircraft executing the CD&R algorithm, **PROVE** all recommended/executed trajectories maintain separation:



Our CD&R algorithms do not need explicit handshake to achieve coordinated resolutions. The only information exchanged is position, and velocity via ADS-B.



Status Of Formal Verifications



- KB3D **formally verified** for **two aircraft** (one maneuvers)
- KB3D **formally verified** to be **cooperative** (both maneuver)
- KB3D vertical maneuvers **formally verified** to be **collaborative** (for N aircraft assuming adequate airspace above)

Current work:

- Adding ability to recover from loss of separation
- Adding target altitude intent information
- Integrating with **prevention bands**
- Extending analysis to cover input inaccuracies and errors



See <http://research.nianet.org/fm-at-nia/KB3D/>



Conflict Detection and Resolution for 1,2,...,N Aircraft, Gilles Dowek and César Muñoz, 7th AIAA Aviation Technology, Integration and Operations Conference, BibTex Reference, 2007.

Formal Verification of an Optimal Air Traffic Conflict Resolution and Recovery Algorithm, André Galdino, César Muñoz, and Mauricio Ayala, 14th Workshop on Logic, Language, Information and Computation, BibTex Reference, 2007.

KB3D Reference Manual - Version 1.a, César Muñoz, Radu Siminiceanu, Víctor Carreño, and Gilles Dowek, NASA/TM-2005-213769, BibTex Reference, 2005.

Provably Safe Coordinated Strategy for Distributed Conflict Resolution, Gilles Dowek, César Muñoz, and Víctor Carreño, AIAA Guidance Navigation, and Control Conference and Exhibit 2005, BibTeX Reference, 2005.

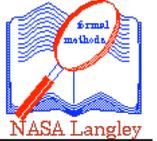
Jeffrey Maddalon, Ricky Butler, Alfons Geser and César Muñoz, Formal Verification of a Conflict Resolution and Recovery Algorithm, NASA/TP-2004-213015, BibTex Reference, 2004.

Ricky Butler, Alfons Geser, Jeffrey Maddalon, and César Muñoz, Formal Analysis of Air Traffic Management Systems: The case of Conflict Resolution and Recovery, Proceedings of the 2003 Winter Simulation Conference, WSC 2003, BibTex Reference, 2003.

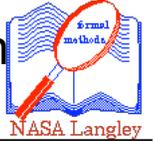
Alfons Geser and César Muñoz, A Geometric Approach to Strategic Conflict Detection and Resolution, Proceedings of the 21st Digital Avionics Systems Conference, DASC 2002, BibTex Reference, 2002.

Alfons Geser, César Muñoz, Gilles Dowek, and Florent Kirchner, Air Traffic Conflict Resolution and Recovery, ICASE Report 2002-12, BibTex Reference, 2002.

Gilles Dowek, Alfons Geser, and César Muñoz, Tactical Conflict Detection and Resolution in a 3-D Airspace, Proceedings of the Fourth International Air Traffic Management R&D Seminar ATM 2001, BibTeX Reference, 2001. Extended version available as ICASE Report 2001-7, BibTeX Reference, 2001.



Loss of Separation



- This work is motivated by some recent TMX studies of the KB3D algorithm.
- The TMX studies explored the capabilities of KB3D to deal with multiple aircraft in complex traffic situations.
- The traffic density was approximately 3x of today's traffic and was generated by extrapolation from existing traffic patterns.
- There were almost no situations where a loss of separation occurred.
- But, it became clear that the algorithm should be generalized to recover from those situations.
- Team: Cesar Munoz and Rick Butler



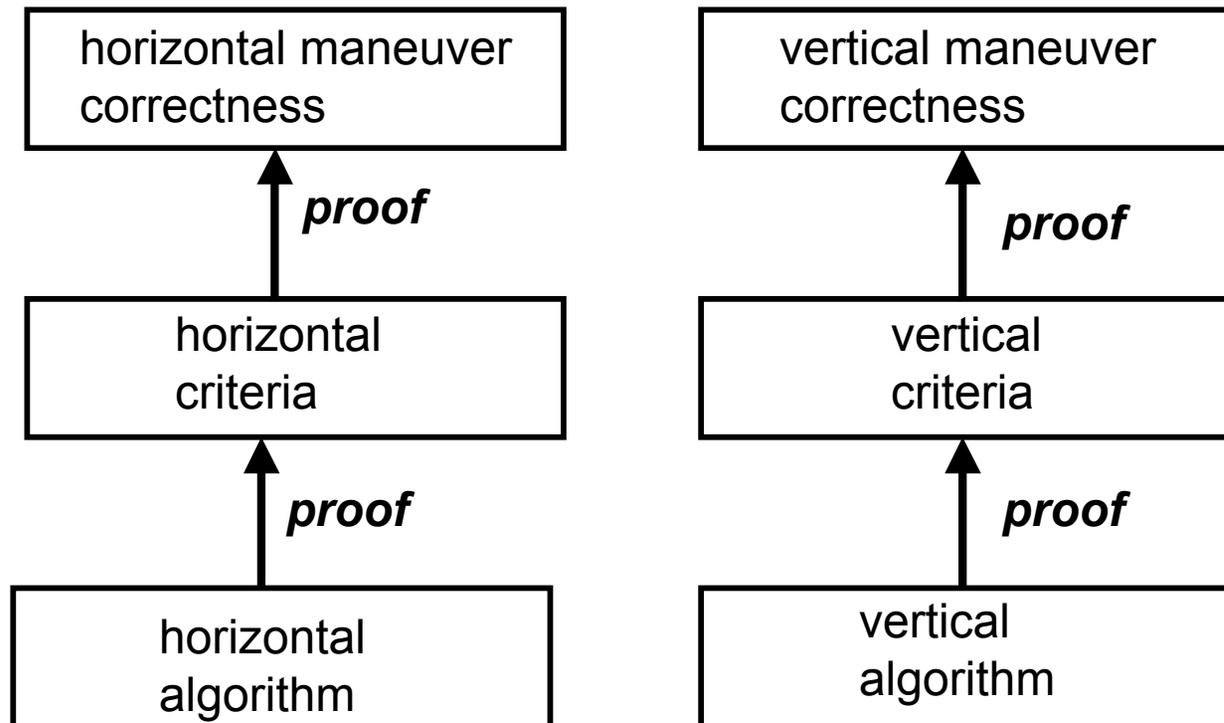
Our Approach



- Rigorous definition of correctness for vertical and horizontal maneuvers
- Simple criteria for loss of separation recovery algorithms
 - Criteria is sufficient to guarantee correctness
 - Criteria is simple enough so that algorithms can be checked against the criteria in a straight-forward way
 - Criteria only uses information available to the local aircraft
- Prove that **Criteria**  **Correctness**
- We want both **independent** and **coordinated** correctness proofs



Approach (Continued)



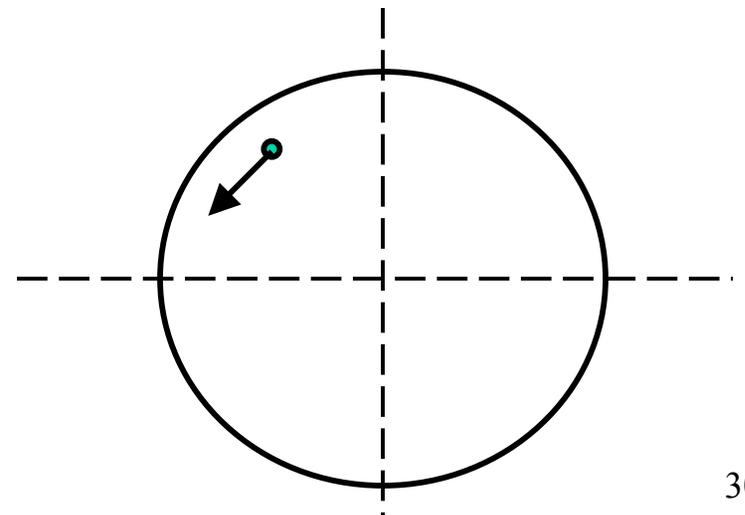
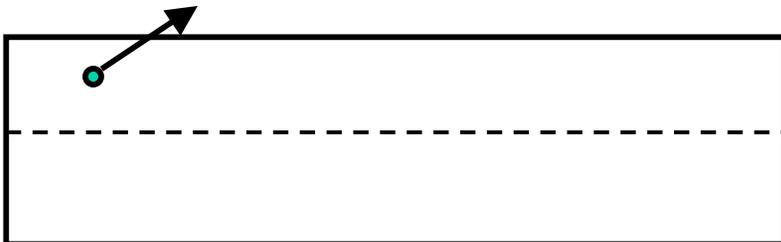
- Proofs are done using the **PVS Theorem Prover** (SRI International)
- Future work will develop horizontal and vertical algorithms that satisfy the intermediate criteria: **Algorithm** \longrightarrow **Criteria**
- It is expected that verification methods developed here will facilitate the proof of correctness of many different kinds of algorithms.



Coordinate Transformation

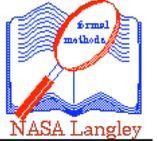


- An aircraft trajectory is modeled as a particle with constant velocity using an initial position p_0 , a velocity vector v and a time parameter t : $p_0 + v t$
 - s_o : Vect3 3D position of ownship
 - s_i : Vect3 3D position of intruder
 - v_o : Vect3 velocity vector of ownship
 - v_i : Vect3 velocity vector of intruder
- Central to the framework is the idea of coordinate transformation $s = s_o - s_i$:
- The intruder is located at (0,0,0)
- s is the position of ownship





Some Definitions



horizontal_separation?(s) = $s_x^2 + s_y^2 \geq D^2$

vertical_separation?(s) = $|s_z| \geq H$

separation?(s):bool =
vertical_separation?(s) OR
horizontal_separation?(s)

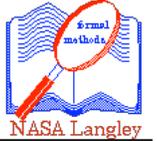
loss_of_separation?(s) : bool = NOT separation?(s)

First Jab at a Definition of Correctness:

EXISTS t: $t > 0$ AND separation?(s+v*t)



Definition of Correctness

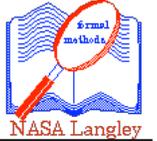


But there are two problems with this as a notion of correctness.

- Almost all trajectories (except parallel trajectories) eventually lead to this condition.
- The time to reach separation may be extraordinarily long (i.e. when the paths are nearly parallel).



Definition of Divergent



- The aircraft are too close and therefore we must insure that they don't get any closer. So we need to explicitly include a concept of distance.

divergent?(s_o, s_i, v_o, v_i): bool =

FORALL t: $\text{dist}(s_o, s_i) < \text{dist}(s_o + t v_o, s_i + t v_i)$

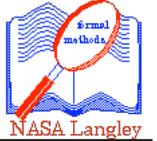
- What distance: **dist** ?

$$\text{xy_dist}(s) := \text{sqrt}(s_x^2 + s_y^2)$$

$$\text{z_dist}(s) := \text{abs}(s_z)$$



Formal Definition of Correctness



- $z_correct?(s, v_i)(v_o')$: bool =
 $z_divergent?(s, v_i)(v_o')$ AND
 $vertical_separation?(s + T_v * (v_o' - v_i))$
- $xy_correct?(s, v_i)(v_o')$: bool =
 $xy_divergent?(s, v_i)(v_o')$ AND
 $horizontal_separation?(s + T_h * (v_o' - v_i))$

T_v = operational parameter that specifies maximum time for vertical recovery.

T_h = operational parameter that specifies maximum time for horizontal recovery.



Proposed Vertical Criteria



- The part that guarantees divergence:

$z_criteria?(s, v_o, v_i)(v_o')$: bool =

$(v_o' - v_i)'z \neq 0$ AND

$sv_prop?(s, v_o' - v_i)$ AND

$(sv_prop?(s, v_o - v_i)$ IMPLIES

$(v_o - v_i)'z \neq 0$ AND

$sign((v_o - v_i)_z) * (v_o' - v_o)_z \geq 0$ OR

$(v_o - v_i)'z = 0$ AND

$break_vz_symm(s) * (v_o' - v_o)'z > 0$)

% Originally diverging

% Relative vertical speed is not 0

% Vertical speed increases

% Aircraft are z-parallel

% Priority aircraft climbs

- Now add criteria for time to exit vertically:

$z_criteria_tr?(s, v_o, v_i)(v_o')$: bool =

NOT $vertical_separation(s)$ AND

$z_criteria?(s, v_o, v_i)(v_o')$ AND

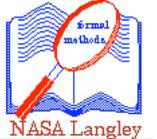
$ttez(s, v_o' - v_i) \leq T_v$

% time to exit

Where $sv_prop?(s,v) = s_z v_z \geq 0$



Independent and Coordinated Vertical Correctness



- $z_independent$: THEOREM
 $z_criteria_tr?(s, v_o, v_i, T_v)(v_o')$
IMPLIES
 $z_correct?(s, v_i)(v_o')$

- $z_coordinated$: THEOREM
 $s \neq 0$ AND
 $z_criteria_tr?(s, v_o, v_i, T_v)(v_o')$ AND
 $z_criteria_tr?(-s, v_i, v_o, T_v)(v_i')$
IMPLIES
 $z_correct?(s, v_i')(v_o')$



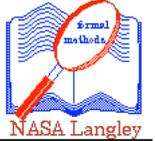
Independent and Coordinated Horizontal Correctness



- xy_independent: THEOREM
 $xy_criteria_tr?(s, v_o, v_i, T_h)(v_o')$
IMPLIES
 $xy_correct?(s, v_i)(v_o')$
- xy_coordinated : THEOREM
 $xy_criteria_tr?(s, v_o, v_i, T_h)(v_o')$ AND
 $xy_criteria_tr?(-s, v_i, v_o, T_h)(v_i')$ AND
 $t_{eh}(s, v_o' - v_i') \leq T_h$
IMPLIES
 $xy_correct?(s, v_i')(v_o')$



We Had Hoped to Prove



xy criteria?(s, v_o, v_i)(v_o') : bool =
dot_prop?(s, v_o' - v_i) AND
(NOT dot_prop?(s, v_o - v_i) OR
NOT dot_prop?(s, v_o - v_o'))

.....

$$\text{tteh}(s, v_o' - v_i) \leq T_h$$

In other words,

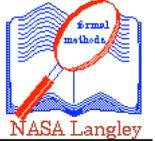
- If each aircraft calculates **new** velocity vector using **original** velocity vector of intruder,
- Together:

$$\begin{aligned} \text{tteh}(s, v_o' - v_i) &\leq T_h \\ \text{tteh}(-s, v_i' - v_o) &\leq T_h \end{aligned}$$

These we give us a global property: $\text{tteh}(s, v_o' - v_i') \leq T_h$



But Then We Found a Counter Example



- We looked for **additional premises** that would enable the proof to go through:
- For example,

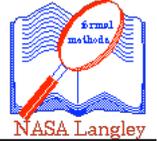
$$\begin{aligned} v_o' * v_o > 0.9 \quad |v_o'| |v_i| & \quad // \text{ cosine angle } > 0.9 \\ s * (v_o' - v_i) > 0.707 \quad |s| |v_o' - v_i| \end{aligned}$$

- Created a Java program to test using $D = 25$ and varying parameters v_{ox}' v_{ix}' v_{oy}' v_{iy}' v_{ox} v_{ix} v_{oy} v_{iy} from -25 to 25
- That is 51^8 test cases = **4.5×10^{13} test cases (AGH!!!!)**
- Step of 4: 4.2×10^8 test cases: OK
- Step of 2: 1.5×10^{11} test cases: OK

Decided to let it run over night with STEP = 1: ????



Counter-Example Found



VO: Vect2 = (-25,-25)

VI: Vect2 = (-25,-24)

NVO: Vect2 = (-11,-19)

NVI: Vect2 = (-7,-25)

S: Vect2 = (-2,12)

D: nat = 25

```
dps(s,vo,vi,nvo,nvi: Vect2): bool = NOT dot_prop?(s,vo-vi) AND  
dot_prop?(s,nvo-vi) AND dot_prop?(-s,nvi-vo)
```

```
% dps(S,VO,VI,NVO,NVI);
```

```
TRUE;
```

```
% ttx(S,NVO-VI);
```

```
<PVSio ttx(S,NVO-VI);
```

```
1.3314613
```

```
<PVSio ttx(-S,NVI-VO);
```

```
1.1073173
```

```
<PVSio ttx(S,NVO-NVI);
```

```
1.8585874
```

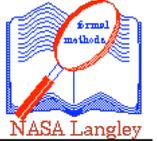
```
<PVSio S*(NVO-NVI);
```

```
==>
```

```
80
```



STEP 1 FOUND COUNTER-EXAMPLES



MORAL: SIMULATION IS NOT SUFFICIENT TO ESTABLISH SAFETY

MORAL: Discretizing the geometry to enable model checking must be done with much caution!