The Generic Resolution Advisor and Conflict Evaluator (GRACE) for Detect-And-Avoid (DAA) Systems

Michael Abramson
Mohamad Refai
Crown Consulting Inc.

Confesor Santiago
NASA Ames Research Center

Outline

- Background
- Motivation
- GRACE Algorithm at a Glance
- Generic Conflict Evaluator (GCE)
- Generic Resolution Advisor (GRA)
- Evaluation of GRACE Algorithm
- Conclusion

Background

Integration of Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS)

- Detect-and-Avoid (DAA) system must enable UAS to steer clear of potential collisions with other aircraft
- Replace the FAA's "see-and-avoid" requirement when there is no pilot onboard <u>with the same or better safety</u>
  - How to quantify "well-clear"?
  - How to communicate with ATC and pilots-in-control?
  - How to interoperate with TCAS or ACAS-XU?
  - How to ...?

<u>The Problem</u>:

How to build a model of DAA system that would help establish requirements to future DAA systems?

Motivation

The Need:

Conflict detection and resolution algorithm for DAA system modeling

- **Flexible**

  to use as a testbed for evaluating new features of DAA systems

- **Fast**

  for real-time (human-in-the-loop) and faster than real-time (parametric and NAS-wide) simulations

- **Lightweight**

  easy to integrate in our DAA system model

GRACE Algorithm at a Glance

Flexible: open architecture

- Use any suitable trajectory predictor
- Configurable
  - separation standards
  - maneuver types (horizontal / vertical / speed)
  - cost functions
- Composed of two loosely coupled algorithms
  - Generic Conflict Evaluator (GCE)
  - Generic Resolution Advisor (GRA)

Fast: execution time ~ milliseconds

Lightweight: seamlessly integrated into DAA simulation

- Simple API and implementation
- Small jar with minimal dependencies

Generic Conflict Evaluator (GCE)

Definitions:

- **Ownship**: aircraft controlled by DAA system
- **Intruder**: other aircraft that can create a collision threat
- **Separation standard**: a logical condition of safe separation
- **Violation** (**loss of separation**): violation of separation standard
- **Threat**: violation properties
  - Severity level
  - Closest Point of Approach (CPA)
  - Time of first loss of separation
  - ...
- **NMAC**: Near-Mid-Air-Collision / violation of NMAC zone
- **Encounter** (**conflict**): a flight situation resulting in violations occurred at least at some trajectory states

Generic Conflict Evaluator (GCE)

Accepts multiple separation standards configurable to cover the wide range of spatial and temporal scales

- Horizontal separation or Horizontal Miss Distance *HMD*
- Vertical separation *VSEP*
- Time metrics
  - *tau* or *tauMod* (tau with distance modifier *DMOD*)
  - altitude-dependent tau (as per TCAS sensitivity levels)
  - vertical tau (time to co-altitude)
  - time to CPA or to first loss of separation

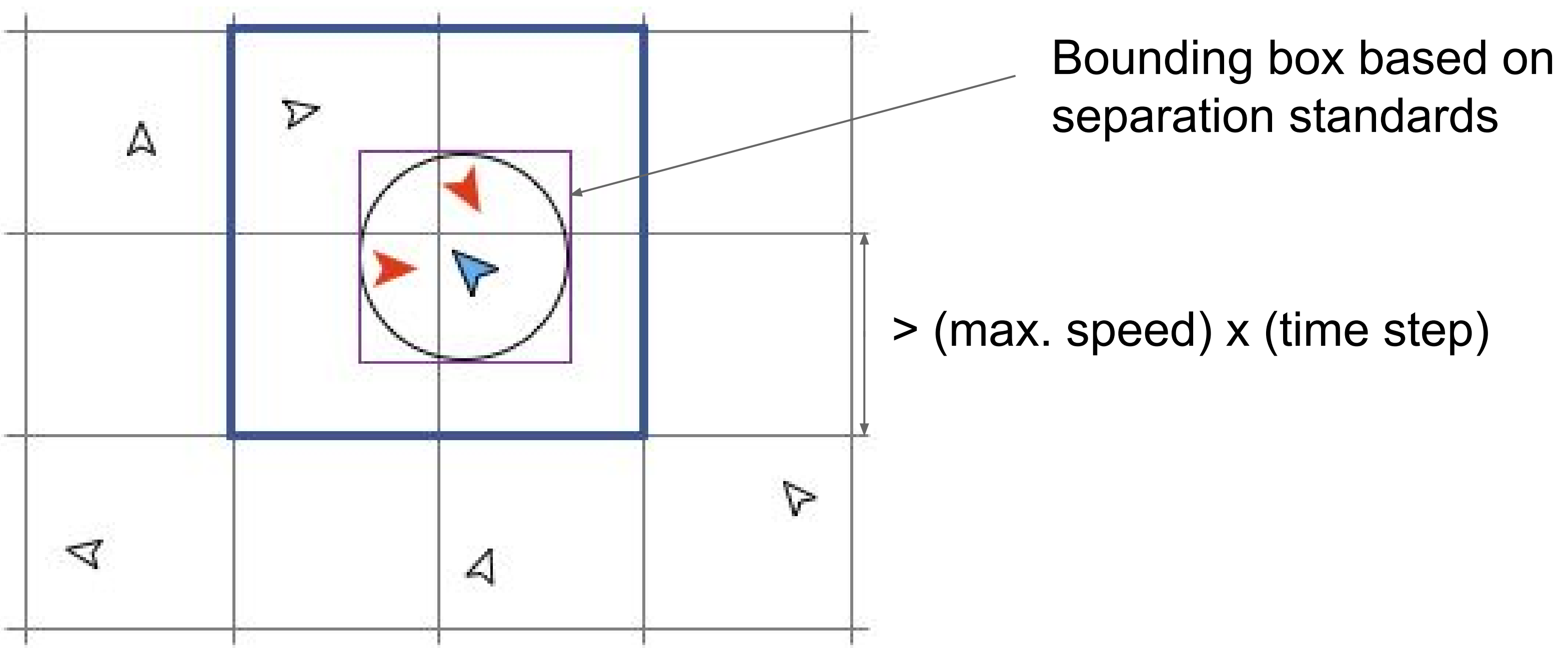
Examples

- DAA Well Clear as defined by the Minimum Operational Performance Standards (MOPS) for DAA systems (expected to be published in June 2017):
  HMD > DMOD = 0.66 nmi, VSEP > 450 ft, tauMod > 35 sec
- NMAC (as defined for TCAS): *500 ft x 100 ft*

Generic Conflict Evaluator (GCE)

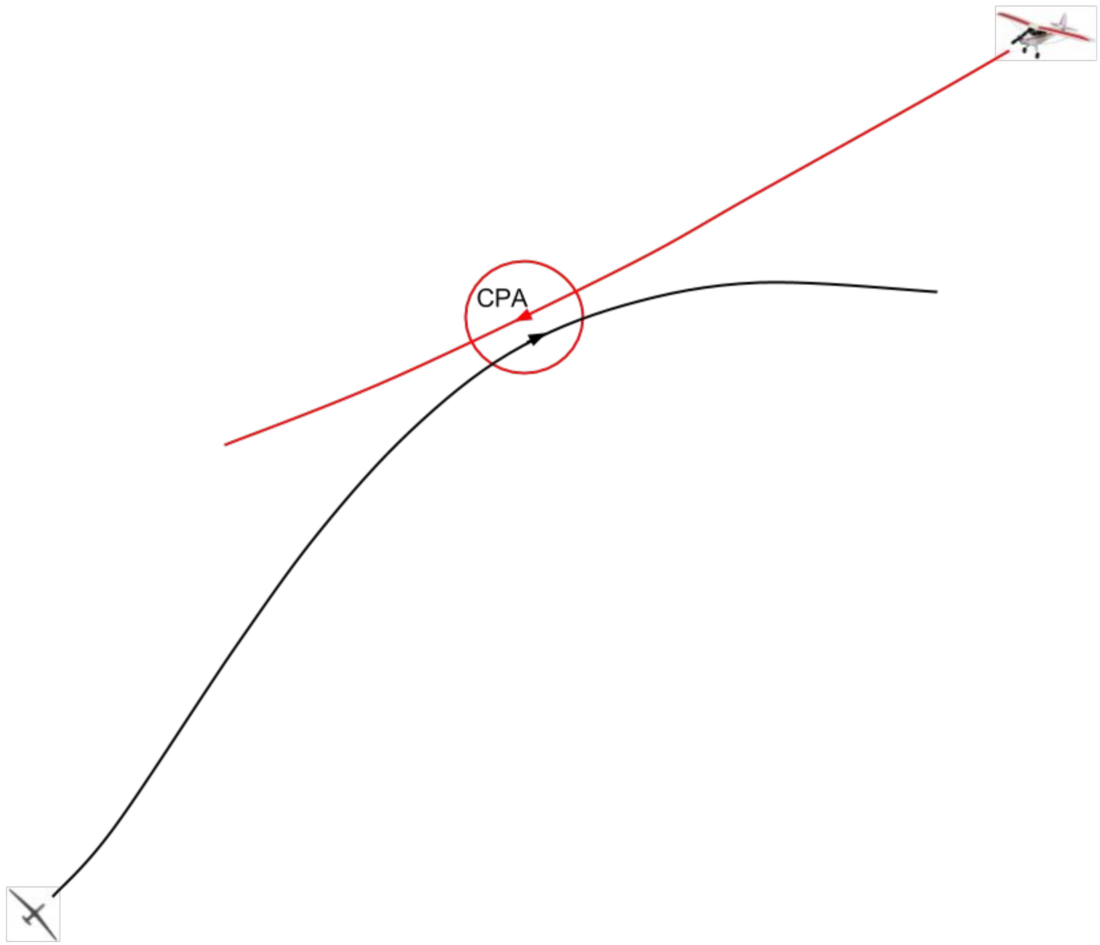
Grid-based threat detection

1. Request predicted intruder trajectories at discrete time steps
2. Determine grid cell size from separation standards and max. speeds
3. Map intruders to grid cells (stored as a time sequence of hash maps)
4. Detect threats from intruders mapped to cells in proximity to ownship



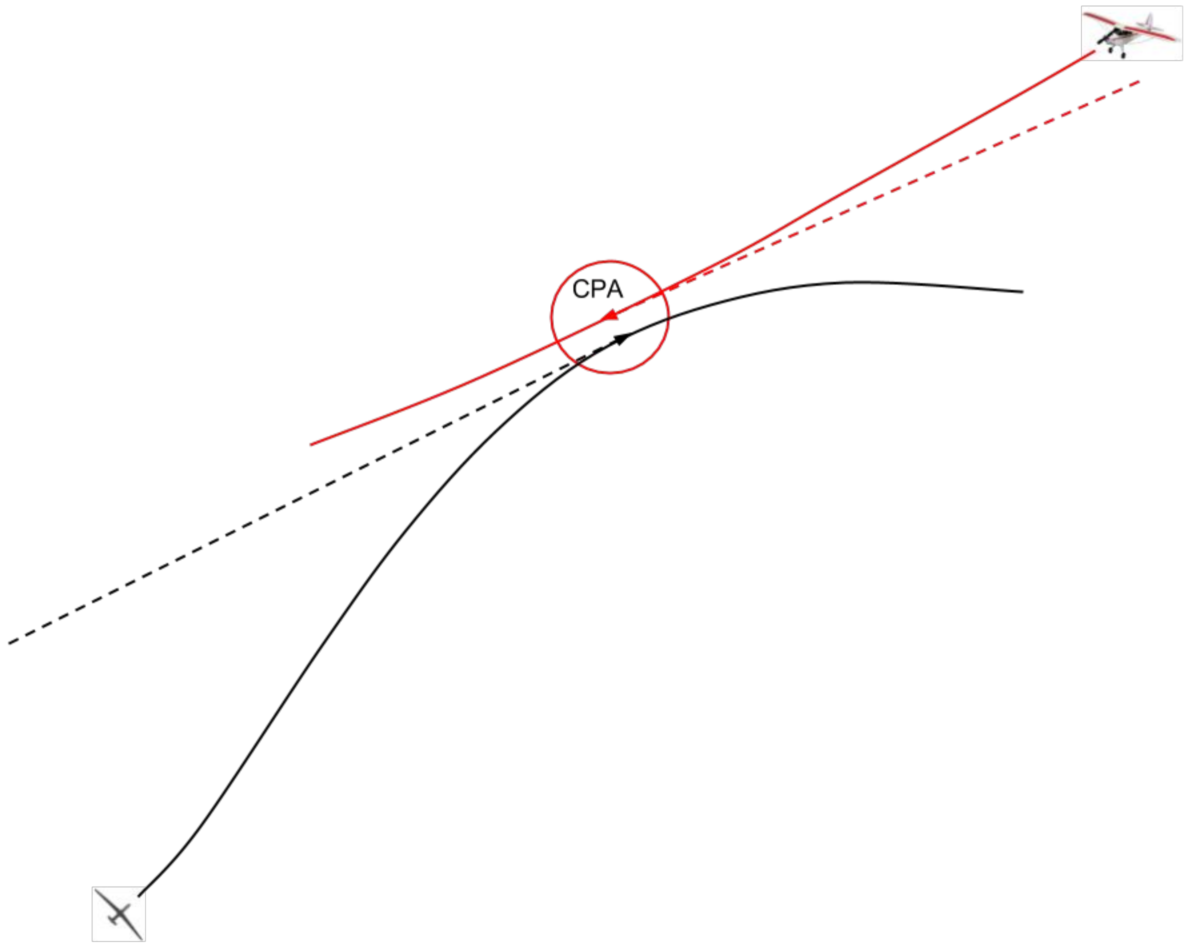
Generic Resolution Advisor (GRA)

Start from CPA for highest priority threat from GCE



Generic Resolution Advisor (GRA)

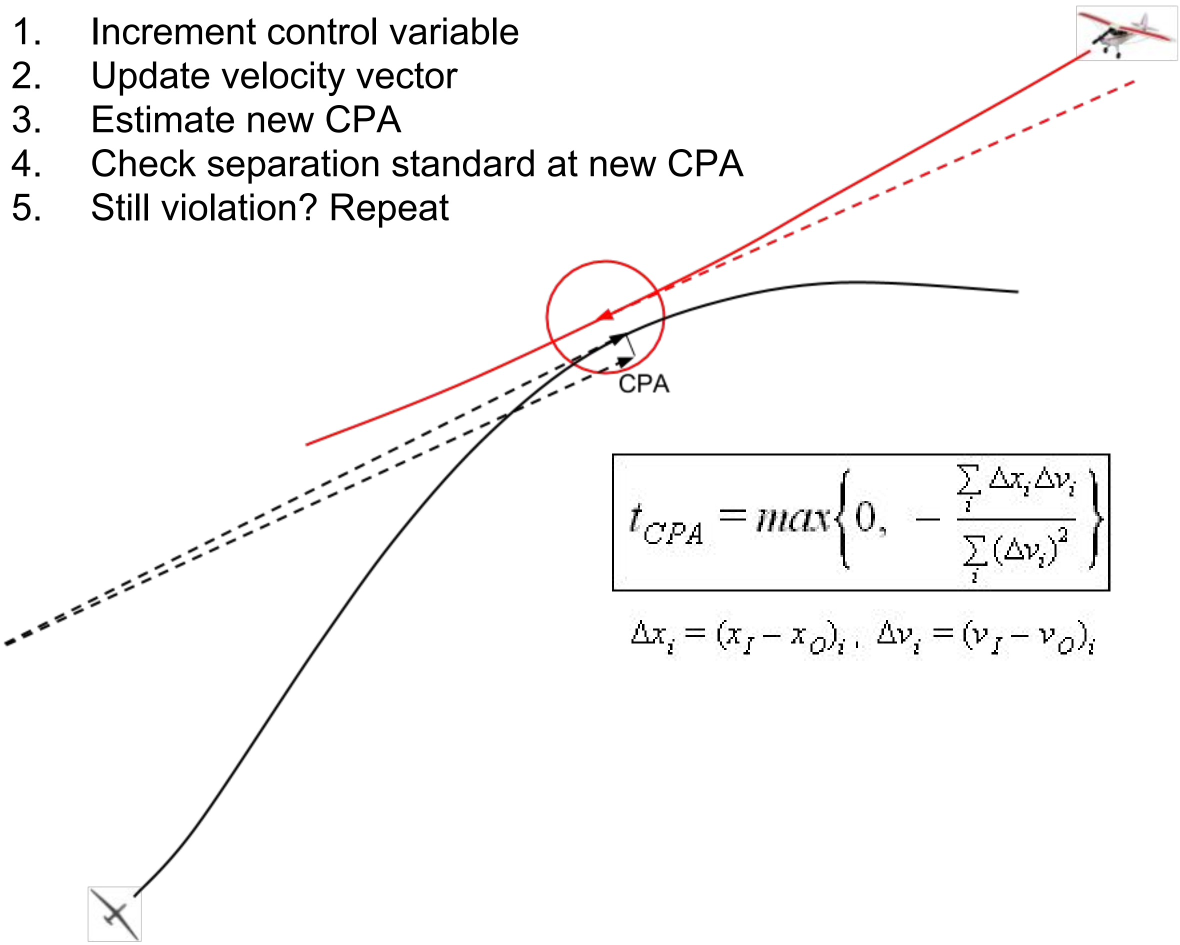
Extrapolate trajectories backward from CPA



Generic Resolution Advisor (GRA)

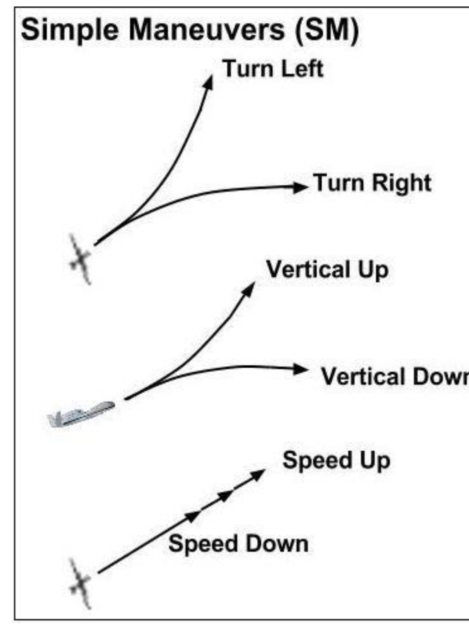
Apply Simple Maneuver to linearized ownship trajectory

1. Increment control variable
2. Update velocity vector
3. Estimate new CPA
4. Check separation standard at new CPA
5. Still violation? Repeat



$$t_{CPA} = max\left\{0,\ -\frac{\sum_i \Delta x_i \Delta v_i}{\sum_i (\Delta v_i)^2}\right\}$$

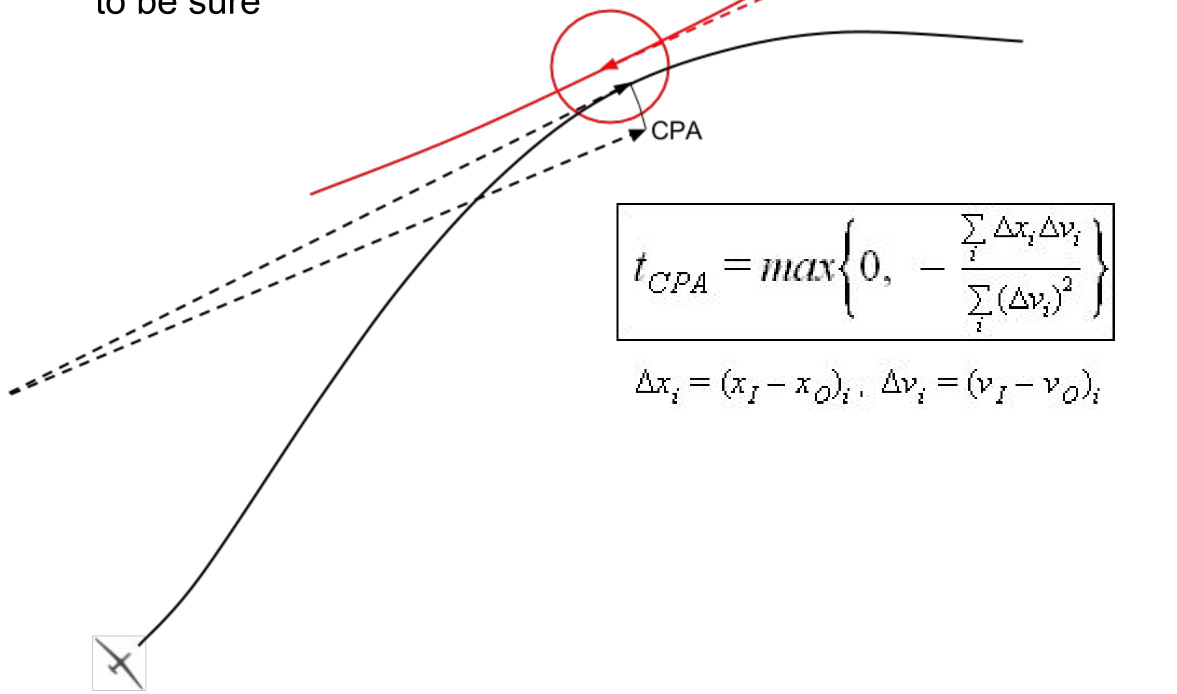
$$\Delta x_i = (x_I - x_O)_i\ ,\ \Delta v_i = (v_I - v_O)_i$$



Generic Resolution Advisor (GRA)

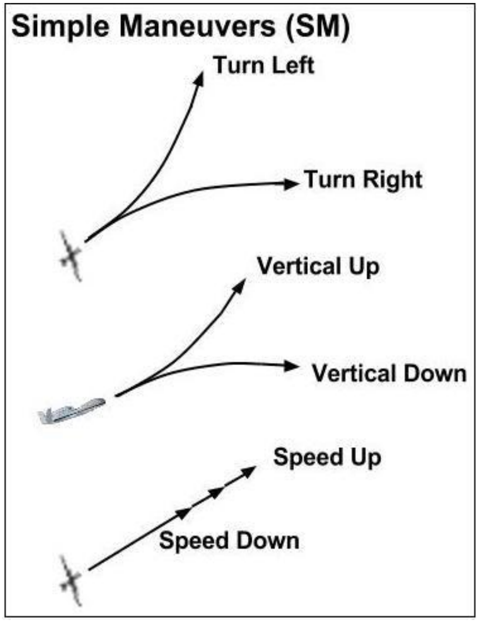
Apply Simple Maneuver to linearized ownship trajectory

1. Increment control variable
2. Update velocity vector
3. Estimate new CPA
4. Check separation standard at new CPA
5. No violation? Repeat one more time just to be sure



$$t_{CPA} = max\left\{0,\ -\frac{\sum_i \Delta x_i \Delta v_i}{\sum_i (\Delta v_i)^2}\right\}$$

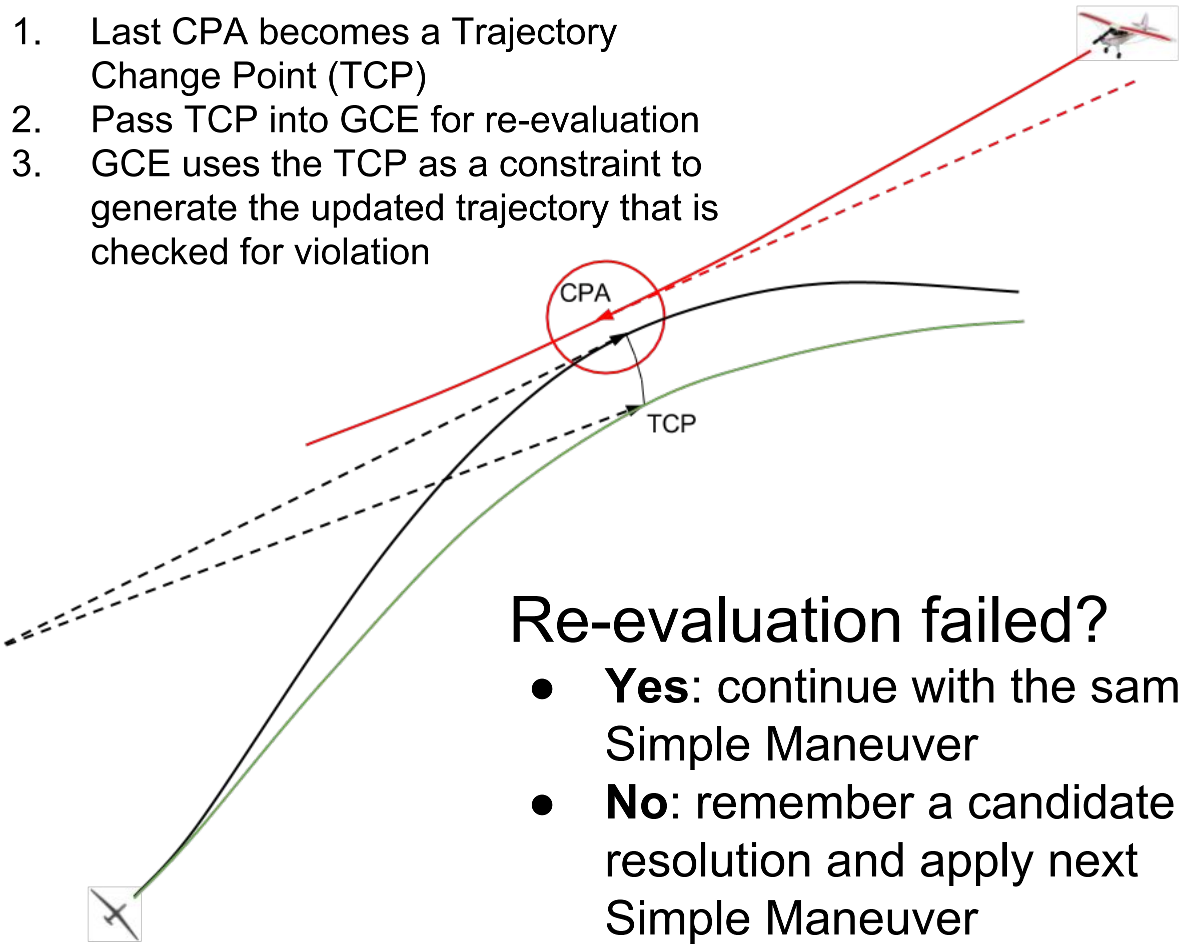
$$\Delta x_i = (x_I - x_O)_i\ ,\ \Delta v_i = (v_I - v_O)_i$$

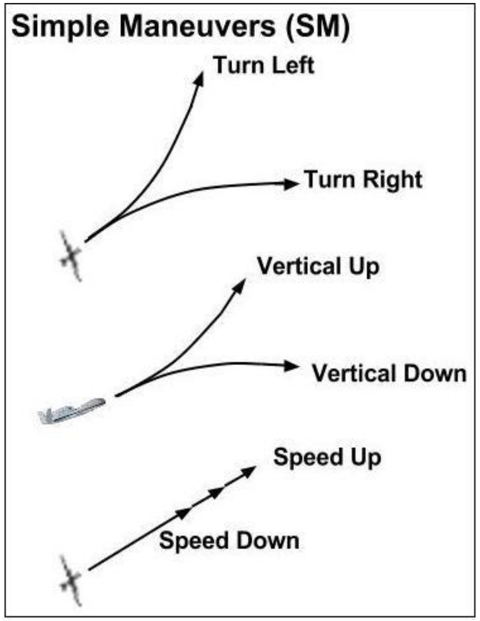


Generic Resolution Advisor (GRA)

Re-evaluate the candidate resolution for all intruders

1. Last CPA becomes a Trajectory Change Point (TCP)
2. Pass TCP into GCE for re-evaluation
3. GCE uses the TCP as a constraint to generate the updated trajectory that is checked for violation





Re-evaluation failed?

- **Yes**: continue with the same Simple Maneuver
- **No**: remember a candidate resolution and apply next Simple Maneuver

Generic Resolution Advisor (GRA)

Find the best solution using a configurable cost function

- Threat of collision cost:
  - penalizes for collision threats from **all** intruders
  - naturally dominates when close to collision
- Maneuver change cost: penalizes frequent changes of maneuvers
- Rank cost: favors the right-of-way compliant maneuvers
- Maneuver strength cost: penalizes too aggressive maneuvers
- Other costs can be added if needed

Evaluation of GRACE Algorithm

NAS-wide 24-hour study

- About 20K UAS missions distributed geographically and temporally (Ayyalasomayajula et al, 2013)
- VFR traffic based on recorded nation-wide radar data
- Transitional class E airspace
- Mix of cooperative and non-cooperative traffic
- DAA surveillance included a directional airborne radar with 8-nmi range
- Possible late detections by surveillance due to
  - “Noisy” VFR tracks
  - Limited sensor field of view
- Unknown intruder intent
- Separation standards based on DAA Well Clear

Evaluation of GRACE Algorithm

Simulations

1. **Unmitigated**: detection only, no resolutions (baseline)
2. **Mitigated**: detection and resolutions with immediate execution and “recapture”

Metrics

- **Violations**: DAA Well Clear violated
- **NMAC**: NMAC separation standard violated
- **Predicted**: per cycle (at every simulation timestep)
- **Actual**: per encounter (at CPA)
- $\mathbf{S_{NMAC}}$: separation at CPA relative to NMAC: exceed 100% for Actual NMAC; lower values correspond to larger separations

Evaluation of GRACE Algorithm

NAS-wide study: effect of mitigation

Simulation	Predicted Violations	Actual Violations	Predicted NMAC	Actual NMAC	S_{NMAC} (%)	
					Mean	Max
Unmitigated	115409	1894	2220	44	29.4	546.3
Mitigated	14394	359	68	0	20.2	92.2

- Mitigation substantially reduced the number of violations
- Remaining violations are mostly due to late detections by surveillance or unexpected intruder maneuvers
- Mitigation eliminated **all** actual NMACs

Conclusion

GRACE Algorithm

- Met performance requirements of DAA simulations
- Validated and used to inform DAA MOPS in a number of
  - human-in-the-loop experiments
  - flight tests with live aircraft
  - NAS-wide simulations and parametric studies
- Can be used by manufacturers and developers of
  - decision support tools for pilots in command
  - systems for fully autonomous UAS DAA operations
  - research and automation software for manned flights

Future work

- Improve robustness of DAA resolutions, taking into account uncertainties of intruder positions and intent

Questions / Discussion

Contact

Michael Abramson

Senior Engineer

Crown Consulting Inc.

michael.abramson@nasa.gov

Backup Slides

Why GRACE?

General purpose conflict detection and resolution algorithm

- Fast and lean
- Designed without any assumptions regarding:
  - temporal and spatial scales
  - performance capabilities of aircraft
  - sensor and communication systems
  - degree of autonomy
- Can be used:
  - in decision support tools for ground pilots
  - in fully autonomous UAS DAA operations
  - potentially in manned flights
- Used extensively in research on UAS DAA systems

Background

2012: The U.S. Congress mandated the “safe integration” of Unmanned Aircraft Systems (UAS) in the National Air Space (NAS)

↓

2012: The FAA aviation rulemaking committee is looking into amending Part 91.113 that prescribes aircraft right-of-way rules, to allow for an electronic “Detect-and-Avoid” (DAA) system enabling UAS to steer clear of potential collisions with other aircraft

↓

2013: The RTCA is developing the minimum operational performance standards (MOPS) for a UAS DAA System

↓

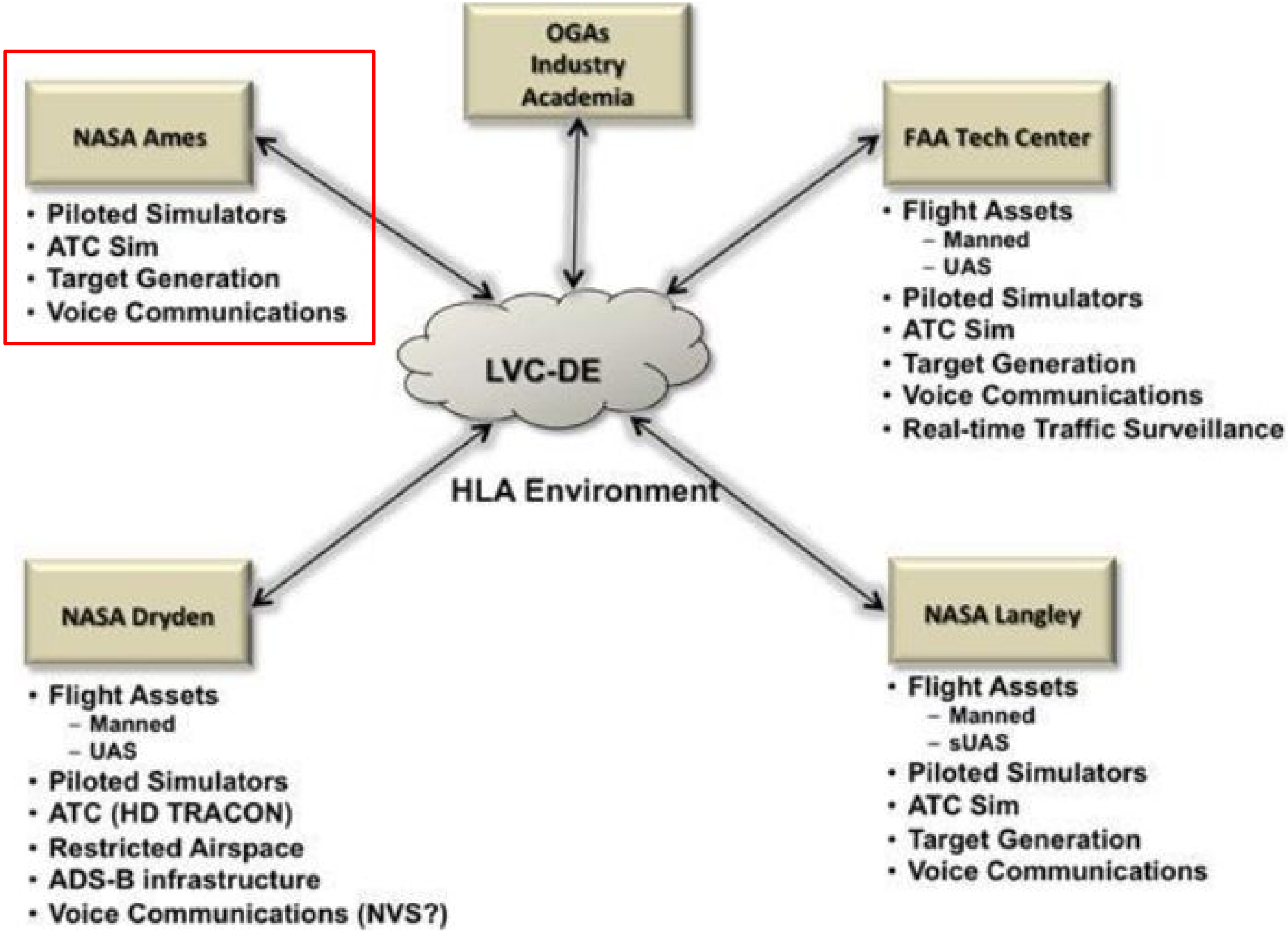
2013: NASA developed a portable software system, Java Architecture for DAA Extensibility and Modeling (JADEM)

Background

UAS Integration in the NAS: NASA Ames role

- **determine the required information to be displayed in the Ground Control Station (GCS) to support the development of standards and guidelines through prototyping and simulation**
- **analyze integration of UAS control and communication system and ATC communications to validate recommendations for regulations and standards**
- **collect and analyze UAS hazard and risk related data to support safety case recommendations for the development of certification/regulations**
- **conduct a "virtual" type design certification effort to develop a "UAS playbook" for industry to obtain type design certificates**
- **develop a relevant test environment to support evaluation of UAS concepts and technologies using a Live Virtual Constructive – Distributed Environment (LVC DE)**
- **instantiate a GCS with display/information to demonstrate compliance with requirements**

Background


OGAs
Industry
Academia
NASA Ames
• Piloted Simulators
• ATC Sim
• Target Generation
• Voice Communications
FAA Tech Center
• Flight Assets
– Manned
– UAS
• Piloted Simulators
• ATC Sim
• Target Generation
• Voice Communications
• Real-time Traffic Surveillance
LVC-DE
HLA Environment
NASA Dryden
• Flight Assets
– Manned
– UAS
• Piloted Simulators
• ATC (HD TRACON)
• Restricted Airspace
• ADS-B infrastructure
• Voice Communications (NVS?)
NASA Langley
• Flight Assets
– Manned
– sUAS
• Piloted Simulators
• ATC Sim
• Target Generation
• Voice Communications

JADEM

Design philosophy

Keep **I**t **S**imple, **S**tupid	● Simple interface and customization ● Few generic classes vs. many specialized classes ● Use Bridge and Adaptor design patterns
Don't **R**epeat **Y**ourself	● One system - many clients ● Unified unit conversion ● Low-level flight data management

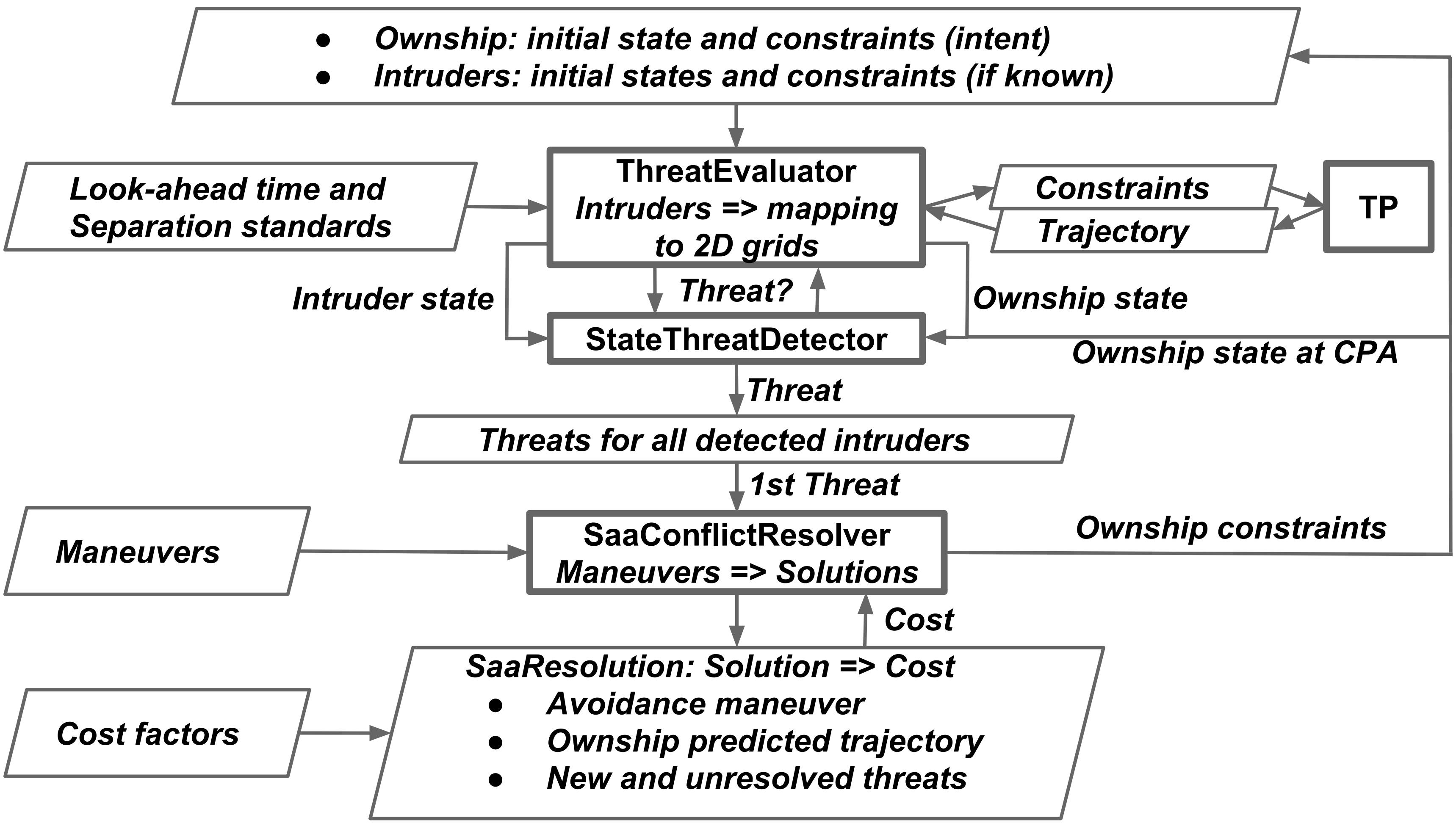
JADEM

Services and spinoffs

- Grid-based mapping for any 2D targets
  - useful for fast search of “neighbors” in 2D space
- Unit conversion and operations with dimensional variables
  - time, space, speed, angle, and mass variables
- Generic flight data management classes
  - natural “language” to express any data, constraints, and operations for aircraft states and entire flights
- Multimodal Adaptable Trajectory Generator (MATG)
  - a general purpose TP, more flexible than others
- Generic surveillance (sensors) model
- GRACE

Overview of GRACE Algorithm

High-level architecture



Generic Conflict Evaluator (GCE)

Key ideas

- Threat evaluation with configurable separation standards
- Grid-based mapping (Von Viebahn, Watkins)

Outputs

- All intruder states mapped to 2D cells for all timesteps up to a specified look-ahead time
- Violations of separation standards (threats) with Closest Points of Approach (CPA)

Generic Conflict Evaluator (GCE)

Separation standards (multiple severity levels)

- Horizontal and vertical separation
- Horizontal miss distance
- Tau-separation, which can be a

  simple tau: $\tau = -\frac{r}{dr/dt}$, or

  modified tau: $\tau_{mod} = -\frac{r - d_{mod}^{2}/r}{dr/dt}$, or

  altitude-dependent tau as defined by TCAS sensitivity levels
- time to separation as a filtering condition.

Generic Resolution Advisor (GRA)

Key ideas

- Local optimization at CPA for highest priority threat
- Discretized solution space by considering a small set of simple maneuvers
- Each candidate resolution is represented by a single Trajectory Change Point (TCP)
- Call GCE to re-evaluate the candidate resolutions for all intruders

Outputs

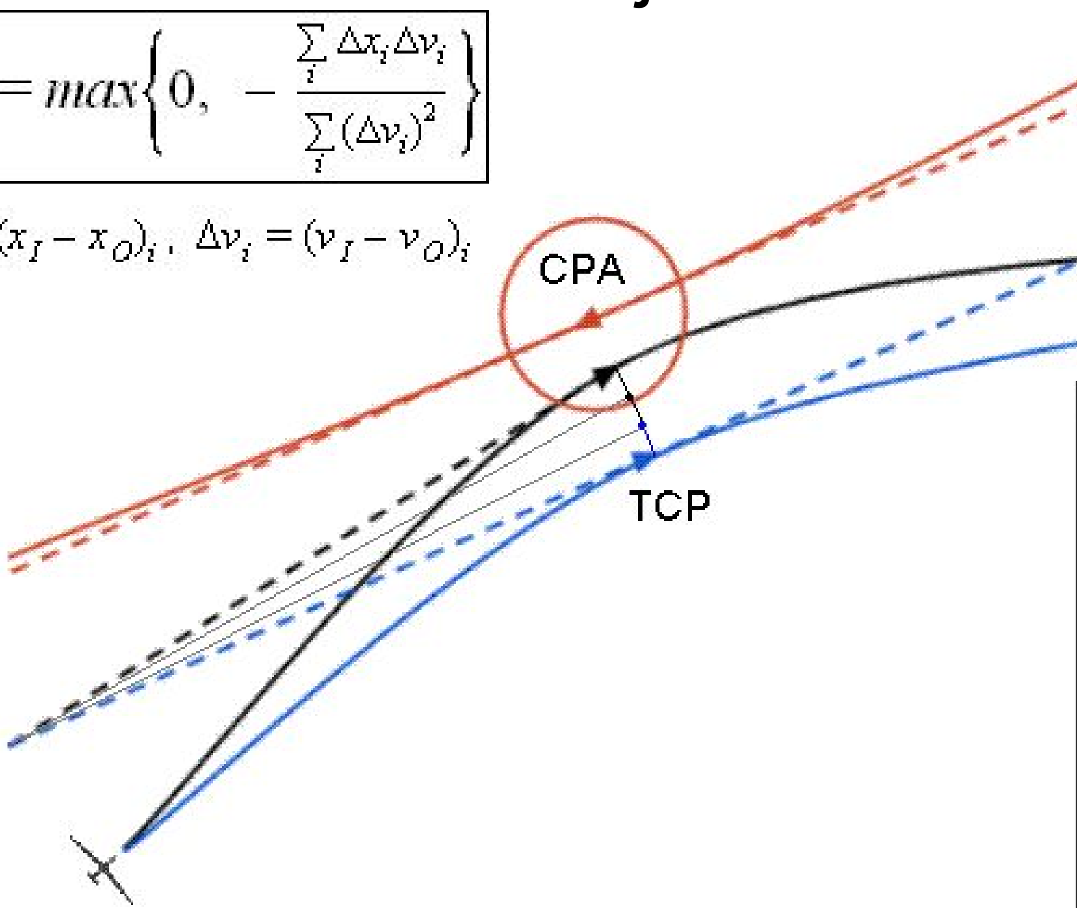
- A set of conflict-free resolutions (one for each simple maneuver)
- The best (least cost) solution

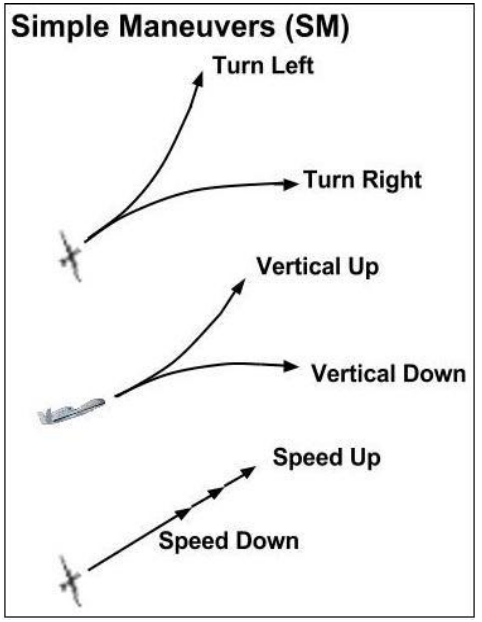
Generic Resolution Advisor (GRA)

TCP for linearized trajectories

$$t_{CPA} = max\left\{0,\ -\frac{\sum_i \Delta x_i \Delta v_i}{\sum_i (\Delta v_i)^2}\right\}$$

$$\Delta x_i = (x_I - x_O)_i\ ,\ \Delta v_i = (v_I - v_O)_i$$





Evaluation of GRACE Algorithm

Parametric study

- 180 encounter geometries for demanding conditions
- Configurations
  - Baseline with “perfect” omnidirectional sensor and immediate execution of GRACE maneuvers
  - Directional airborne radar model paired with a tracker with and without sensor error (“noise”)
  - Model of delayed execution of GRACE maneuvers

NAS-wide study

- UAS missions and recorded VFR traffic for 24 hours
- Baseline configuration

Evaluation of GRACE Algorithm

Metrics

Violations: any separation standard violated

NMAC: Near-Mid-Air-Collision zone (500 x 100 ft) penetrated

Predicted: per cycle (at every simulation timestep)

Actual: per encounter (at CPA)

Changes Per Encounter: number of changes in resolution type divided by the number of encounters

Failure Rate: a ratio of number of Actual Violations to number of encounters in %

$\mathbf{S_{NMAC}}$: separation at CPA relative to NMAC: exceed 100% for Actual NMAC; lower values correspond to larger separations

Evaluation of GRACE Algorithm

Parametric study: effect of surveillance system

Sensor	Predicted Violations	Predicted NMAC	Changes per Encounter	Actual Violations	Failure Rate (%)	S_{NMAC} (%)	
						Mean	Max
Omnidirectional	209	174	0.056	1	0.6	15	15
Directional perfect	246	149	0.006	9	5.1	34.3	39.5
Directional noisy	380	4	1.11	13	7.3	31.8	46.8

- Resolved all conflicts that were not detected too late
- Number of changed resolutions per encounter
  - very low for perfect sensors
  - increased for noisy sensors due to large vertical errors of non-cooperative radar sensor

Evaluation of GRACE Algorithm

Parametric study: delayed pilot response

Total Pilot Delay	Predicted Violations	Predicted NMAC	Changes per Encounter	Actual Violations	Failure Rate (%)	S_{NMAC} (%)	
						Mean	Max
10 seconds	2316	1849	0.64	1	0.6	15.6	15.6
15 seconds	3419	2636	1.00	6	3.3	8	12.5
20 seconds	4686	3463	1.27	7	3.9	9.9	14.7

Even for largest pilot delay (20 sec):

- All NMACs avoided with a comfortable safety margin
- Less than 4% encounters with Well Clear violations
- ~1 changed resolution per encounter

Pilot delay assumed by GRACE (the algorithm delay): 15 sec