Computing Risk for Unmanned Aircraft Self Separation with Maneuvering Intruders

Jason Adaska
jason.adaska@numerica.us
Numerica Corporation

Outline

- Self Separation Capability
- Computing Risk with Maneuvering Intruders
- Simulation Results

Self Separation Component

- **Sense-and-Avoid** (SAA) consists of two components:
  1. Collision Avoidance (CA)
  2. Self Separation (SS)

- **Self Separation** - keep "a safe distance from other aircraft so as not to cause the initiation of a collision avoidance maneuver" [FAA Workshop on SAA, 2009]

- **Challenges of SS**
  – "Long" time horizon predictions (i.e., 1-3 minutes)
  – VFR (unexpected maneuvers) ➔ dead reckoning is risky
  – Non-cooperative intruders ➔ intruder position/heading/speed uncertain even at current time (noisy AB/GB SAA Sensor)

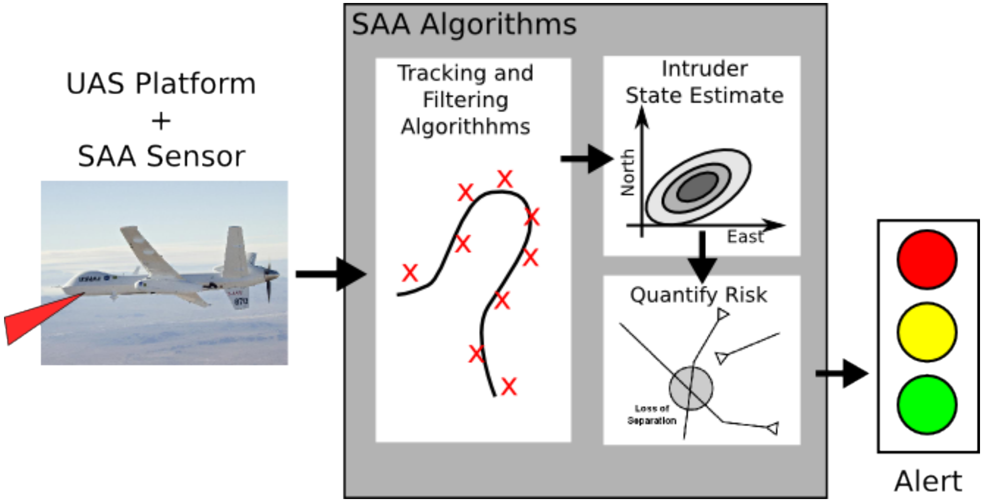
Elements of a Self Separation Capability

Envisioned Pilot Alerting Scheme

Input:
- (noisy) intruder state estimate
- ownship intent
- safety threshold

Output:
- Alert if computed risk exceeds threshold (a la TCAS traffic advisory)

Failure Modes
- *Missed detection* – no alert but maneuver was required
- *False alarm* – alert but no maneuver was required
- Probability of Detection/False Alarm (Pd/Pfa) quantifies performance



Computing Future Conflict Risk

Risk = Probability of Future NMAC

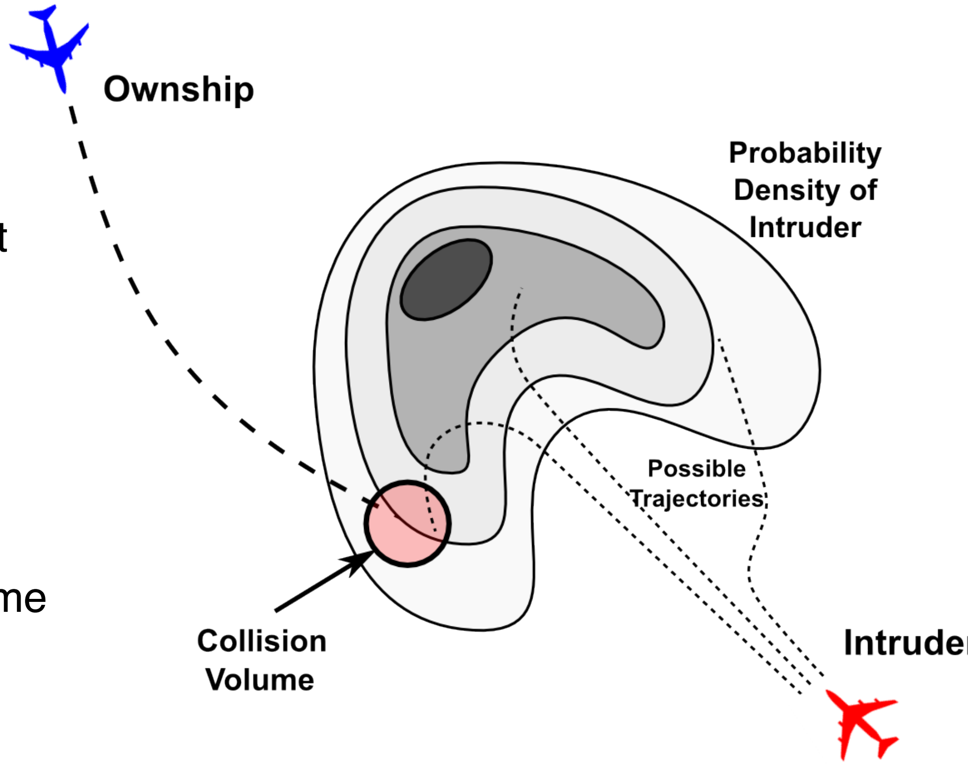
- Alert when computed probability exceeds safety threshold
- [Weibel, Edwards, Fernandes 2011] suggest defining “well-clear” in probabilistic terms

Probability of Future NMAC (PNMAC)

- Stochastic model for intruder determines a probability density of future position(s)
- Future ownship trajectory and collision volume defines future NMAC region
- PNMAC at time t is an integral:

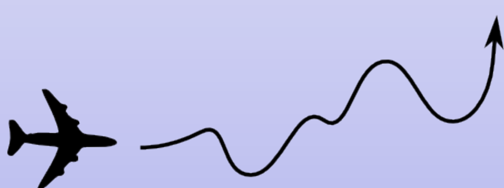
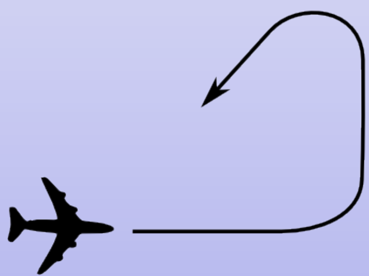
$$\mathrm{PNMAC}(t) = \int \cdots \int_{\mathrm{C.V.}} \rho(t,x)\,dx$$

- Approximate PNMAC over time horizon w/ sequence $\{\mathrm{PNMAC}(t_i)\}$



Models of Uncertain Maneuvering Intruders

- **Future intruder trajectory** is the primary uncertainty
  - Assume non-reactive intruders (i.e., blunder scenario)
  - *Focus on modeling approach amenable to online algorithms!*

Dead Reckoning	Brownian Diffusion	Jump Linear System
• Deterministic, straight line extrapolation of current state	• Stochastic trajectory • White-noise drives velocity/ acceleration. • Ubiquitous in target tracking (e.g., Kalman filter)	• Stochastic trajectory • Models maneuvering modes (level flight, coordinated turn, etc.) • Uncertain sequence/ duration of maneuvers
		

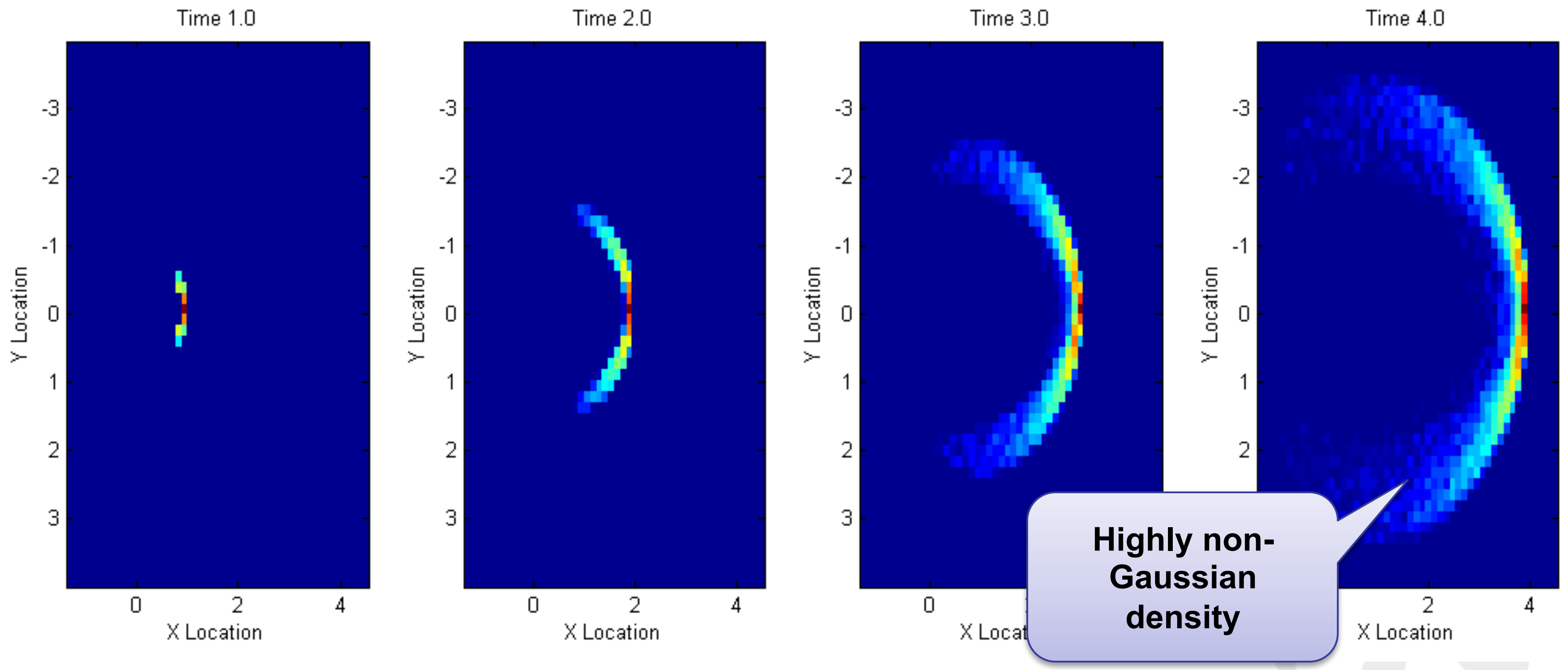
Realistic trajectory variation

Jump Linear Systems (JLS)

$$\frac{d\mathbf{x}}{dt} = A_{z(t)}\mathbf{x}$$

x(t) continuous state (pos.-vel.)
z(t) discrete maneuver mode
A(z) linear maneuver dynamics

Example density evolution for 2D aircraft model w/ 3 modes:
(0) Level flight, (1) Coordinated Turn Left, (2) Coordinated Turn Right



Connecting JLS Moments to JLS Density

Need density to compute PNMAC.

- Unlike Brownian diffusions, JLS density has no closed form solution, but **moments analytically computable**

$$\mathrm{PNMAC}(t) = \int \cdots \int_{\mathrm{C.V.}} \rho(t, \mathbf{x}) \, d\mathbf{x}$$

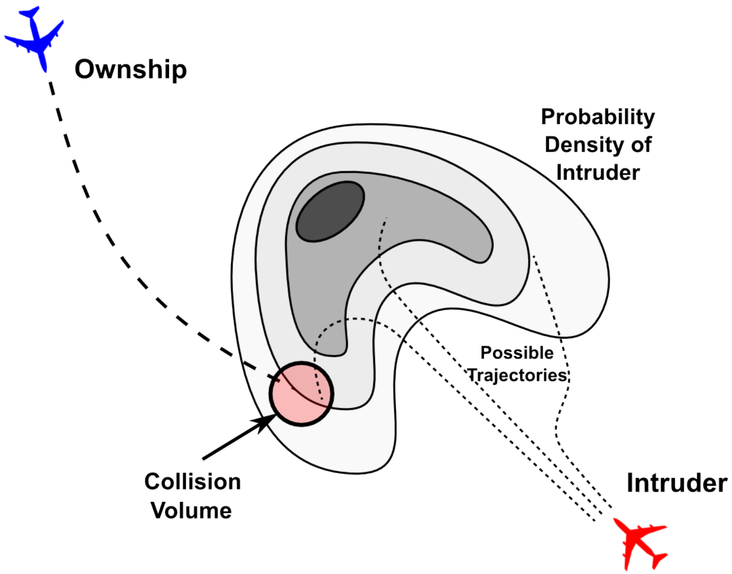
Problem:

- Mean and covariance are not enough to determine “shape” of JLS density ρ(x)
- Moment inversion (many moments ➔ density) is non-trivial
- Computing high order moments is impractical

Resolution:

- Partition the trajectories into sets
- Approximate density in each set with low order moments (e.g., Gaussian sum)

$$\rho(x) = \sum_{a \in A} \mathrm{P}(a) \rho(x|a) \approx \sum_{a \in A} \mathrm{P}(a) \hat{\rho}_a(x)$$

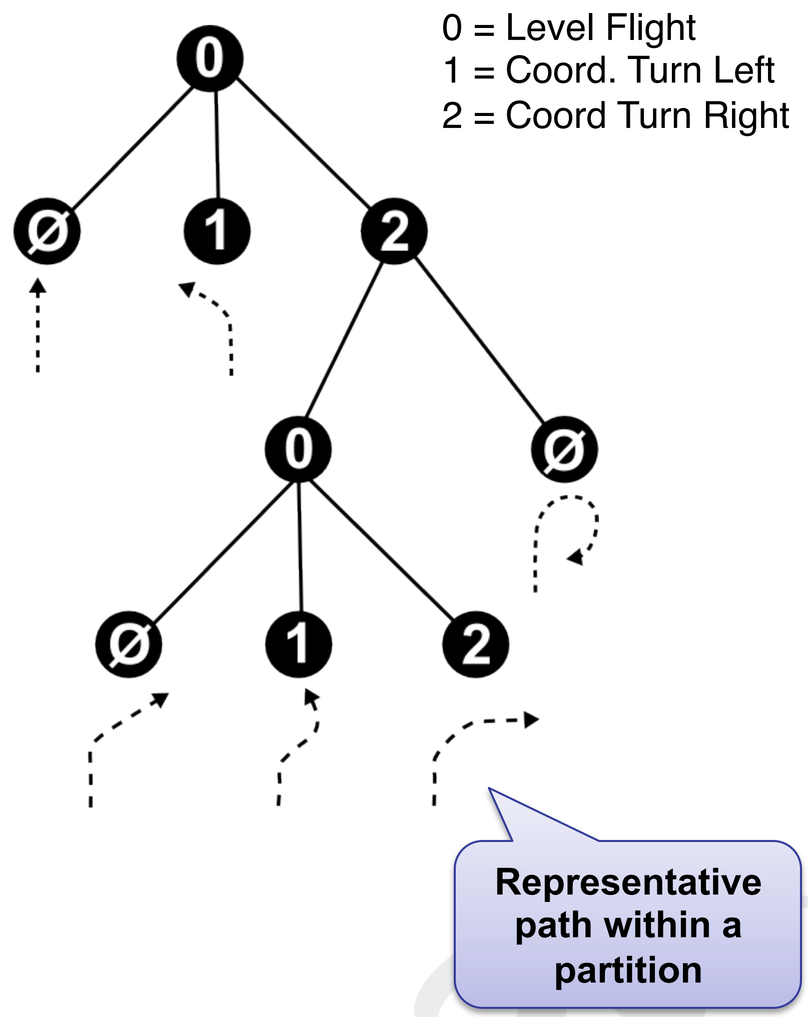


JLS Partition Tree

JLS density partitioned via the maneuver mode process z(t)

Hierarchical Tree Structure

- Partition defined by root-leaf paths
- Two Refinement types:
  - **Next transition**: determine next maneuver mode.
  - **Time split**: bisect time span for a maneuver transition
- Structure allows *iterative refinement* by expanding a leaf

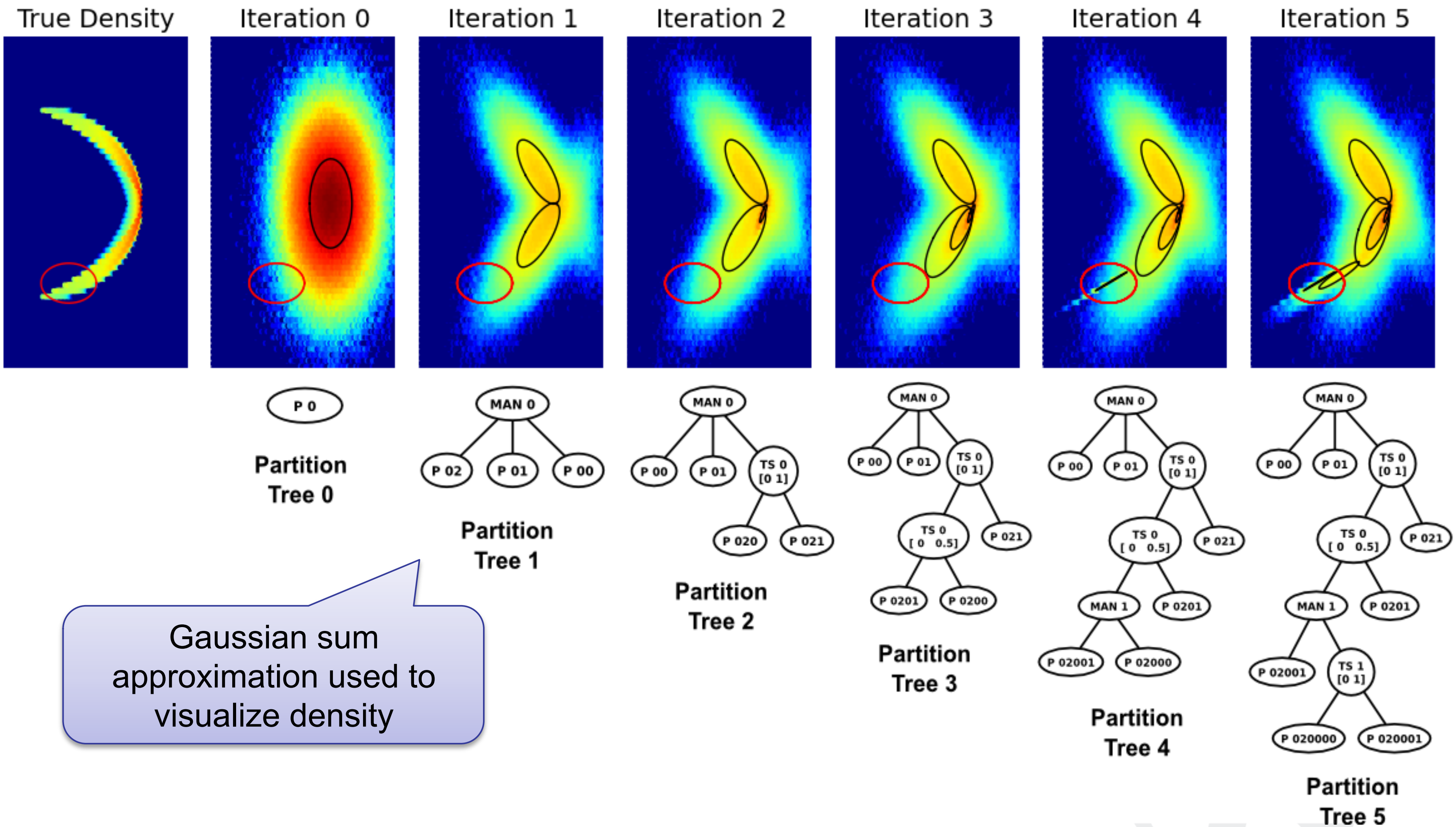


Benefits

- Moments easily computed for each leaf
- Work required to compute PNMAC can be adapted to encounter
  - Coarse partition sufficient when risk level is clear (very high PNMAC, very low PNMAC)
  - Refinement can be directed toward CV

Adaptive JLS Density Refinement

Density iteratively adapted to Collision Volume (red)



Quantifying Self Separation Performance

Benefit of more accurate PNMAC is improved Pd/PFa

Pilot Alerting Scheme – alert when computed PNMAC > τ

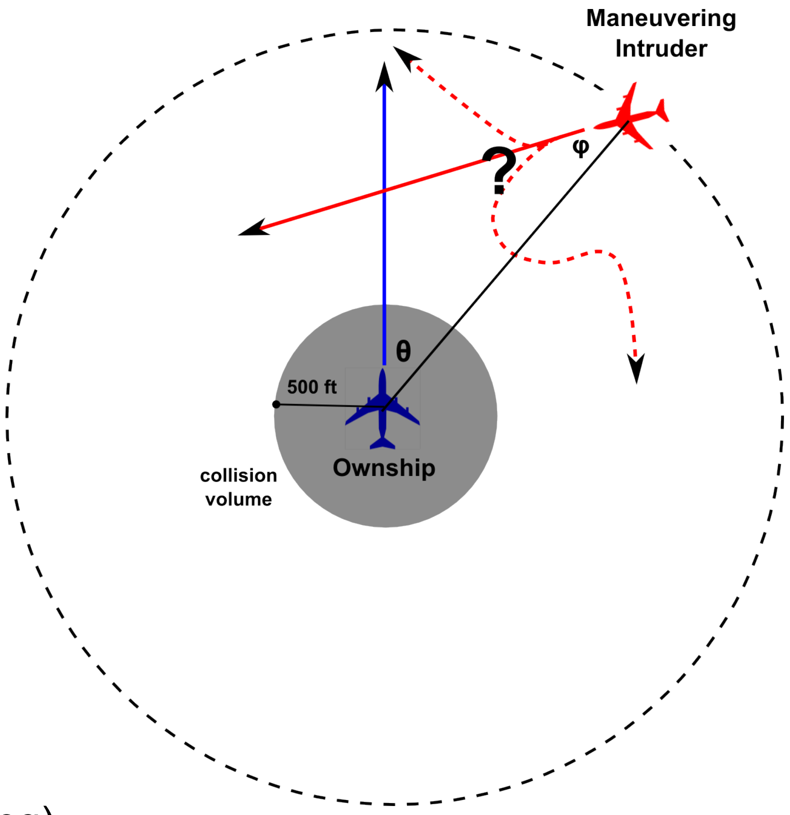
- Goal:
  - generate an alert when ownship maneuver is needed to avoid future (1-3 minutes) NMAC.
  - Stay silent otherwise.

Quantifying performance

1. Simulate a random set relevant encounters
2. Compute PNMAC for each encounter
3. Sweep τ to generate PFa vs. Pd (i.e. Receiver Operating Characteristic Curves)

Simulation Results [1/2]

- **Intruder-Ownship Geometry**
  - Range 2.5 to 5.0 nmi
  - θ: 0 to 360°
  - ϕ: -90° to 90°
  - common altitude
- **Ownship intent**
  - 80 knots due North
- **Intruder intent**
  - 140 knots
  - “True” trajectory drawn from JLS
    - Avg. level flight = 3 minutes
    - Avg. turn = 30 second
- **SAA Sensor**
  - Notional onboard radar (ABSAA)
- **Compared Approaches**
  - Closest point of approach (via dead reckoning)
  - PNMAC using Brownian Diffusion (match statistics but not “shape”)
  - PNMAC using Jump Linear Systems



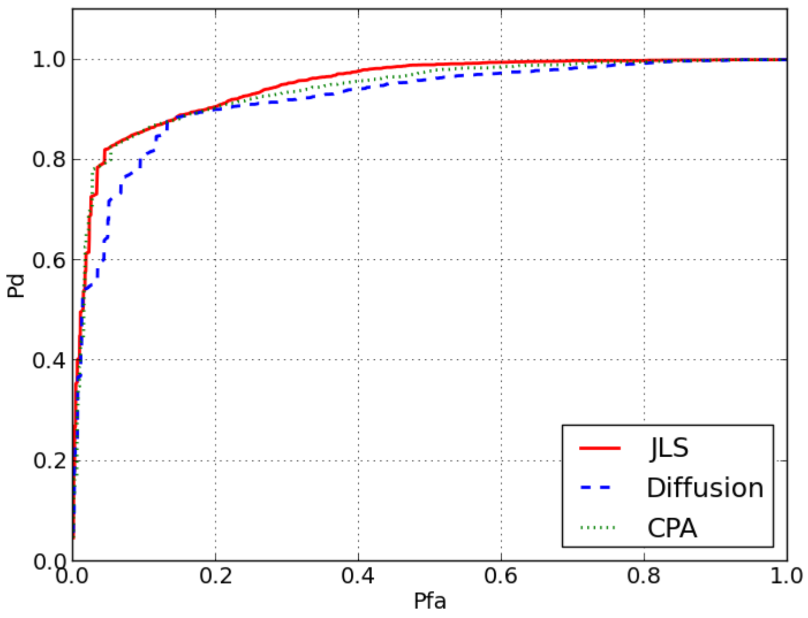
Simulation Results [2/2]

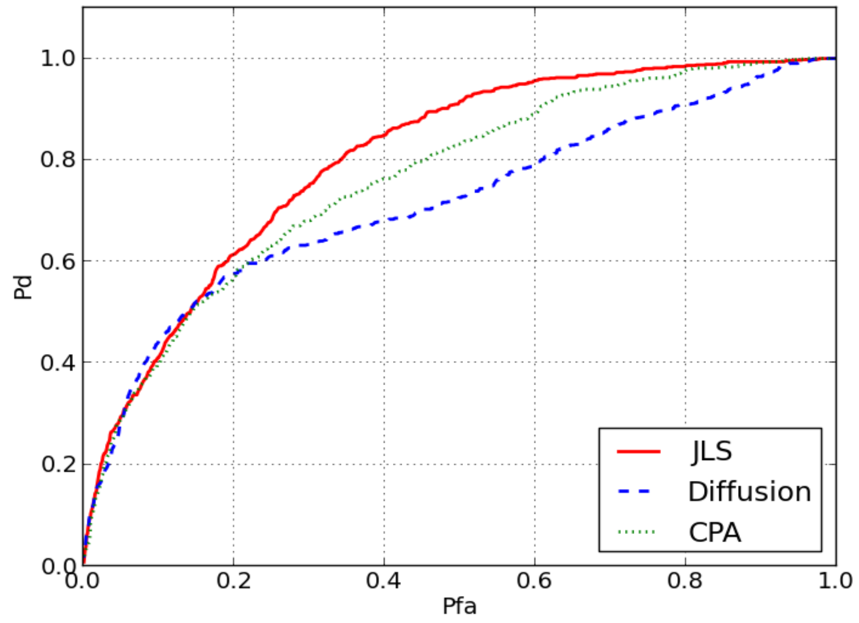
Averaging over all encounters

- Intruder usually doesn't maneuver.
- Diffusions worse than dead reckoning.
- JLS satisfies "do no harm" criterion.
- Using JLS delivers slight improvement.

Averaging over "relevant" encounters

- Intruder maneuvers before time of CPA.
- Maneuver possibilities have an impact.
- Using JLS delivers significant improvement.





Acknowledgements

This work is supported by NASA Phase II SBIR Funding

In particular, thanks to:

Doug Isaacson (ARC)
Mark Pestana (Dryden)
Brandon Suarez (General Atomics)

jason.adaska@numerica.us

(970) 461-2000

www.numerica.us