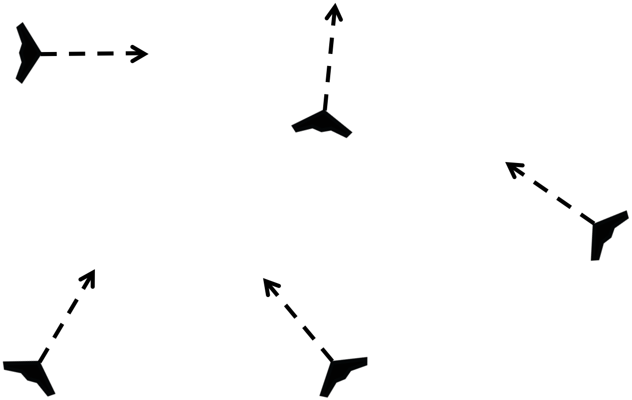
Horizontal Collision Avoidance Coordination: A Brief Investigation

Hao Yi Ong

SISL, Stanford University

Outline

Multiagent coordination and control



Horizontal coordination

given potential collisions, what is the best action for each aircraft?

- incorporate horizontal coordination to resolution advisory
  - observation that coordinated TCAS encounters result in decreased risk ratio
- resolution based on look-up table and coordination message
  - choose ownship action from relative position, relative heading, intruder speed, ownship speed, and intruder coordination message
- note that each aircraft's view of the world is slightly different
  - measurement noise, intruder range quantization, etc.

ACAS X_U versus UTM

problem is mathematically similar, but with practical differences

ACAS X_U	UTM
decentralized control	centralized control
state uncertainty	state uncertainty
limited coordination bandwidth	“unlimited” coordination bandwidth
$\sim$40 s to collision	$\sim$3 min to conflict

Outline

Multi-agent Markov decision process

- formulating and solving ACAS X_U problem as a decentralized Markov decision process (DEC-MDP) is infeasible
  - DEC-MDP: each agent observes local state, collective observability
  - for $n \geq 3$ agents, problem is NEXP-complete
- strategy involves decomposition and local optimization techniques
  1. decompose problem into $O\left(n^2\right)$ pairwise encounters
  2. use pairwise solutions to get locally optimal solution to full problem

Decomposition

solve pairwise encounter subproblem as MDP

- state (x, y, ψ)
  - intruder's position and heading relative to ownship
  - discretized state space to reduce problem complexity

- action
  - aircraft banking maneuver for fixed time period
  - choice of resolution bank angle out of a finite set corresponding to left turns, right turns, and straight paths

- state transition
  - Dubins kinematic model used to find nominal next state
  - multi-linear interpolation for transition probabilities

Decomposition

- reward
  - captures safety and operational considerations

Variable	Reward	Condition
minSep	-1000	minSep $<$ reqMinSep
minSep	$-10 \exp\left(\text{minSep}^2/\text{reqMinSep}^2\right)$	-
bankAngle	$-0.02 \sum \text{bankAngle}^2$	-

- solution
  - QMDP generates state-action value function as a look-up table
  - look-up table forms offline solution to a pairwise encounter

Online operation

each aircraft

1. computes initial local solution to observed problem
   - local solution derived using utility fusion and the joint equilibrium search for policy (JESP) local optimization technique
   - note again that all aircraft can have different observations
2. broadcasts its coordination message
   - includes own action and suggested turn sense for each aircraft
   - turn sense: go straight, turn left or right, or do anything
3. incorporates coordination messages to offline solution
   - convert suggested turn sense into online costs
   - run JESP again using each aircraft's broadcast actions and online costs to obtain final solution

Online operation

each aircraft

1. computes initial local solution to observed problem
   - local solution derived using utility fusion and the joint equilibrium search for policy (JESP) local optimization technique
   - note again that all aircraft can have different observations
2. broadcasts its coordination message
   - includes own action and suggested turn sense for each aircraft
   - turn sense: go straight, turn left or right, or do anything
3. incorporates coordination messages to offline solution
   - convert suggested turn sense into online costs
   - run JESP again using each aircraft's broadcast actions and online costs to obtain final solution

Online operation

each aircraft

1. computes initial local solution to observed problem
   - local solution derived using utility fusion and the joint equilibrium search for policy (JESP) local optimization technique
   - note again that all aircraft can have different observations
2. broadcasts its coordination message
   - includes own action and suggested turn sense for each aircraft
   - turn sense: go straight, turn left or right, or do anything
3. incorporates coordination messages to offline solution
   - convert suggested turn sense into online costs
   - run JESP again using each aircraft's broadcast actions and online costs to obtain final solution

Utility fusion

- uses state-action values from pairwise encounter subproblems to get a proxy to the full problem's state-action value
  - also called cost fusion
- for a set of pairwise MDP states and actions

$$s = (s_0, s_1 \ldots), \quad a = (a_0, a_1 \ldots)$$

the best action for each aircraft (indexed 0) can be found via

  - max-sum strategy

$$a_0 = \underset{a_0}{\operatorname{argmax}} \widetilde{U}^\star(s, a) = \underset{a_0}{\operatorname{argmax}} \sum_i \widetilde{U}_i^\star(s_i, a_i, a_0)$$

  - max-min strategy

$$a_0 = \underset{a_0}{\operatorname{argmax}} \widetilde{U}^\star(s, a) = \underset{a_0}{\operatorname{argmax}} \left(\min_i \widetilde{U}_i^\star(s_i, a_i, a_0) \right)$$

Local optimization

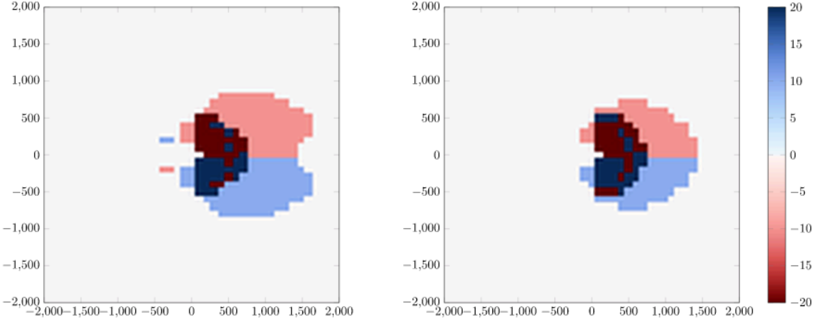
- JESP: all but one action is held fixed while the remaining aircraft computes the best policy in response to the others
  - "best" according to state-action value proxy from utility fusion
  - inspired by game-theoretic level-k thinking, where decisions are made based on beliefs about what other players will do
- coordination message incorporation
  - each aircraft's broadcast own action used as fixed actions for all other aircraft's JESP
  - reward action for each coinciding turn sense suggestion

Outline

Pairwise encounter solution

for a head-on encounter with an intruder

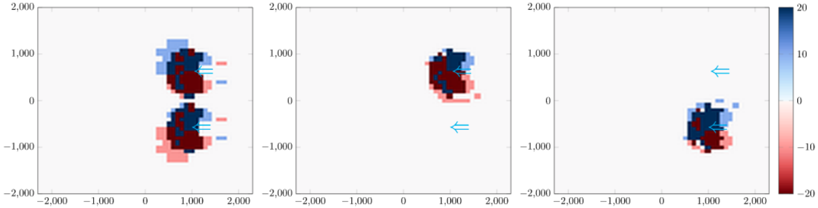
- minimum separation of 500 m
- ownship at 10 m/s going right and intruder at 10 m/s going left
- +ve and -ve values correspond to left and right turn angles



3-aircraft scenario

max-sum policy

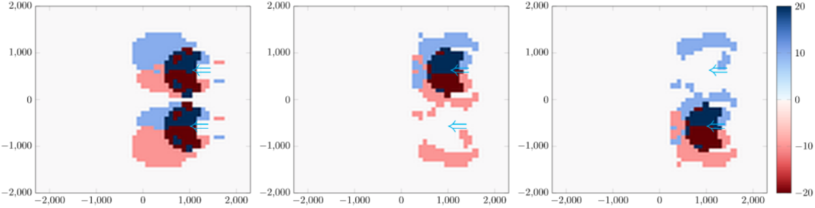
- same as pairwise encounter: 500 m minimum separation and positive and negative angles correspond to left and right turns
- left inset indicates what ownship should do given the xy position, remaining 2 indicate what intruders should do if ownship is at the shown xy position



- slight asymmetry due to convergence towards different local optima

3-aircraft scenario

max-min policy



- more conservative than max-sum
- slight asymmetry again due to local optima convergence
- "patches" on middle and right insets indicate that both intruder aircraft should both turn to the left or right in a synchronized fashion

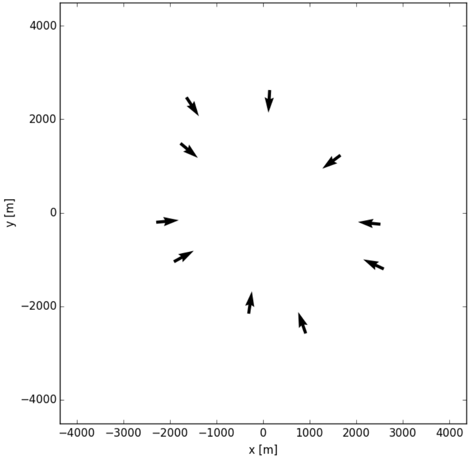
Stress tests

experimental set-up

- randomly initialize 2–9 aircraft in an annulus
  - all start with 10 m/s speed for simplicity
  - all start with pairwise minimum separation $>$ 750 m
  - all headed towards center of annulus
- state updates
  - measurement imperfection modeled with position and heading discretization (nearest 50 m and 10 degrees, respectively)
  - ODE solver used for dynamics equations, with Gaussian noise in acceleration and turning
  - action for each aircraft used to generate PID control policy during simulation

Example initial configuration

aircraft test initialization



Stress test policies

1. naive, closest-pair solution (no utility fusion)
   - simply finds the closest intruder and execute the best action based on the pairwise encounter MDP look-up table
2. decentralized, uncoordinated with utility fusion
   - assumes all white noise intruders and executes best JESP action based on assumption
3. decentralized, coordinated with utility fusion
   - follows our algorithm described in previous section
4. centralized with utility fusion
   - executes JESP solution for all aircraft, with measurement uncertainty (quantization of state variables)

Stress test policies

1. naive, closest-pair solution (no utility fusion)
   - simply finds the closest intruder and execute the best action based on the pairwise encounter MDP look-up table
2. decentralized, uncoordinated with utility fusion
   - assumes all white noise intruders and executes best JESP action based on assumption
3. decentralized, coordinated with utility fusion
   - follows our algorithm described in previous section
4. centralized with utility fusion
   - executes JESP solution for all aircraft, with measurement uncertainty (quantization of state variables)

Stress test policies

1. naive, closest-pair solution (no utility fusion)
   - simply finds the closest intruder and execute the best action based on the pairwise encounter MDP look-up table
2. decentralized, uncoordinated with utility fusion
   - assumes all white noise intruders and executes best JESP action based on assumption
3. decentralized, coordinated with utility fusion
   - follows our algorithm described in previous section
4. centralized with utility fusion
   - executes JESP solution for all aircraft, with measurement uncertainty (quantization of state variables)

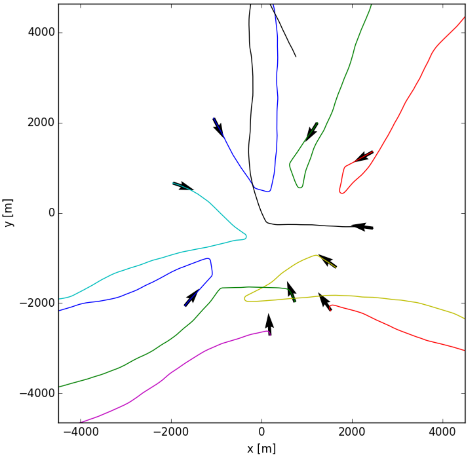
Stress test policies

1. naive, closest-pair solution (no utility fusion)
   – simply finds the closest intruder and execute the best action based on the pairwise encounter MDP look-up table
2. decentralized, uncoordinated with utility fusion
   – assumes all white noise intruders and executes best JESP action based on assumption
3. decentralized, coordinated with utility fusion
   – follows our algorithm described in previous section
4. centralized with utility fusion
   – executes JESP solution for all aircraft, with measurement uncertainty (quantization of state variables)

Example simulation result

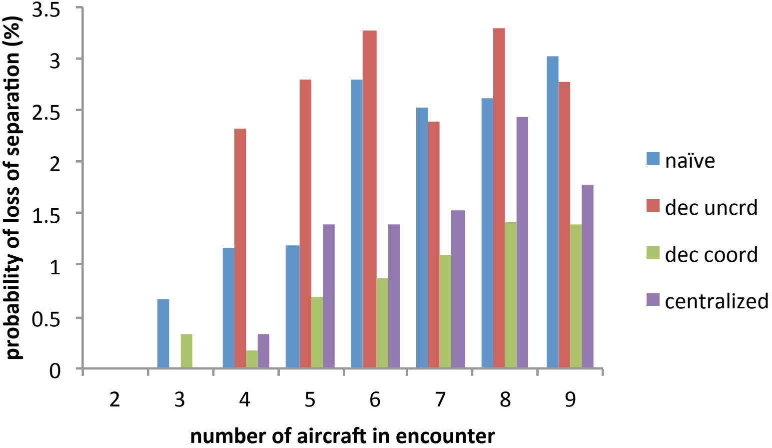
aircraft simulation: decentralized, coordinated with utility fusion

- no loss of minimum separation (or NMAC) observed



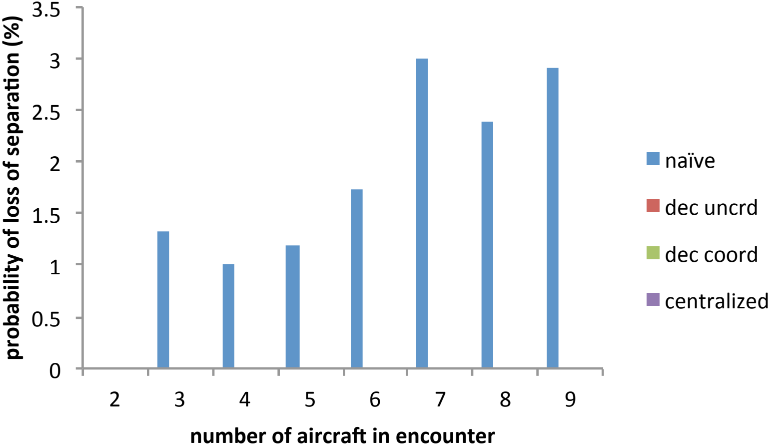
Simulation results

max-sum utility fusion



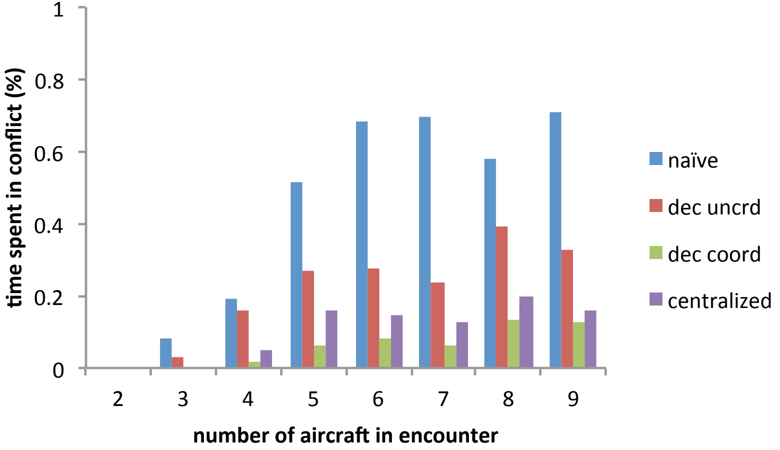
Simulation results

max-min utility fusion



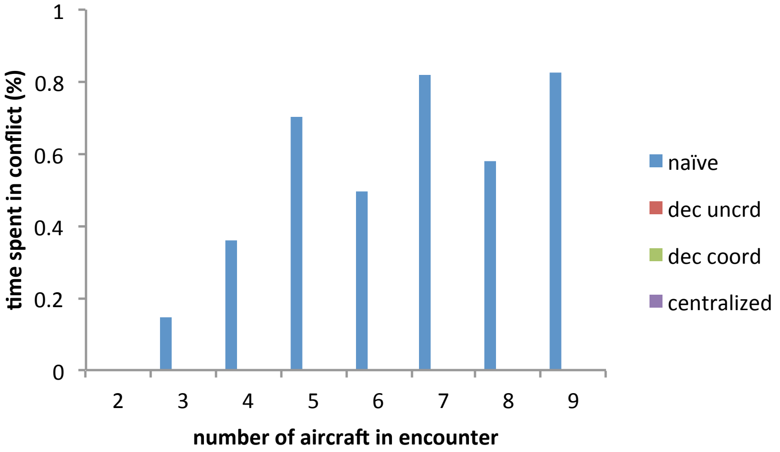
Simulation results

max-sum utility fusion



Simulation results

max-min utility fusion



Discussion

- max-min utility fusion works much better than max-sum
  - max-min results in more conservative policy (*e.g.*, planes turn earlier, as suggested in heatmap)
  - in general, leads to more alerts (alert transitions not modeled)
- although number of pairwise loss of separation is greater for decentralized, uncoordinated control vs. naive control, the amount of time spent in loss of separation is much higher for the latter
  - note that loss of separation does not imply collision in UTM's context
- surprisingly, decentralized controller with coordination does better than centralized controller (max-sum only)
  - more accurate ownship information improves policy vs. centralized controller, which only has access to quantized state variables

Outline

Extensions

- extend JESP with tabu search for better local optimization
  - heuristic to find better local optima than first one found with standard JESP
- use finer state discretization for pairwise encounter MDP
  - experiment and find out whether finer state discretization is worth the extra complexity (offline policy computation and online look-up)
- incorporate return-to-path actions
  - switch between return-to-path controller and collision avoidance depending on environment (more relevant to UTM)

Questions

- how to model observation noise and state transitions better
  - current set-up uses an MDP for pairwise encounters
- reconsider incorporation of coordination messages
  - how to better use suggested actions
  - switch to "do not turn left/right"-type messages (may not make sense, unlike vertical coordination)
- how to consider encounters between different aircraft types
  - difference in maneuverability, dynamics (VTOL?)