

Intelligent UAS Sense-and-Avoid Utilizing Global Constraints

Javier Barreiro and Minh Do and David E. Smith

NASA Ames Research Center
Moffett Field, CA 94035

javier.barreiro, minh.do, david.smith@nasa.gov

Abstract

Sense-and-avoid (SAA) is a critical research topic for enabling the operation of Unmanned Aircraft Systems (UAS) in civilian airspace. SAA involves two planning related problems: 1) plan-recognition to predict the future trajectory of nearby aircraft, and 2) path planning to avoid conflicts with nearby aircraft that pose a threat. We have designed and built components of a novel intelligent sense-and-avoid (iSAA) reasoning framework that takes into account information about aircraft type, transponder code, communications, local routes, airports, airspace, terrain, and weather to more accurately predict near- and medium-term trajectories of nearby aircraft. By using this additional information both the on-board control software and the ground-based UAS operator can make more informed, intelligent decisions to effectively predict and avoid conflicts and maintain separation. While this capability benefits all categories of UASs operating under both Instrument Flight Rules (IFR) and Visual Flight Rules (VFR), it is absolutely essential for allowing smaller UASs to operate VFR at low altitude in uncontrolled airspace for operations such as survey work, wildlife tracking, aerial photography, utilities inspection, crop dusting, and package delivery.

Introduction

Unmanned Aircraft Systems (UAS) come in a wide range of sizes, from tiny to quite large as illustrated in Figure 1. As you would expect, they have an equally wide range of performance characteristics and capabilities, including cost, operating altitudes, range, speed, instrumentation, communication abilities, and payload capacity. To date, UASs have been used predominantly for military applications, but there is growing demand to employ these vehicles for a wide range of civilian purposes, including such things as: search and rescue, traffic monitoring and reporting, wildlife monitoring and surveys, fire and flood monitoring, pipeline and transmission line inspection, aerial photography, cropdusting, and package delivery. All of these applications, require operation of UASs in civilian airspace at lower altitudes. To date, UASs have essentially been operated under Instrument Flight Rules (IFR), which means that they have to adhere to a strict flight plan, and Air Traffic Controllers (ATC) have authority over their operation and route. This is not practical for smaller vehicles, for vehicles operating at low altitudes (below radar coverage), and for many of the applications envisioned for these vehicles. To operate in this environment,



Figure 1: Some small, medium, and large UASs with widely varying performance and capabilities.

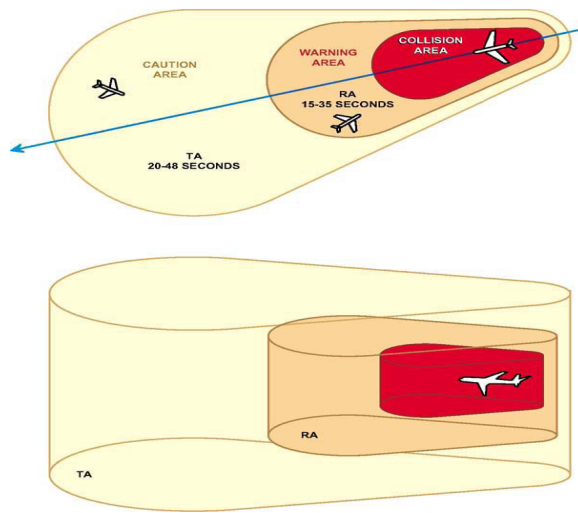


Figure 2: Scope of traffic advisories (TA) and resolution advisories (RA) for TCAS II.

these vehicles must be able to operate under Visual Flight Rules (VFR), and be able to Sense and Avoid (SAA) other aircraft in the same way that most small aircraft currently operate.

There are many aspects to the SAA problem including sensor technology, sensor integration, communication technology and security, threat recognition, and threat resolution. Because of the interest in UASs, there has been a great deal of work on many of these topics. The focus of this work is on the threat recognition and resolution problems, for which we assume a variety of inputs from various possible sensors and sources.

Related Work

Since 2000, the Traffic Collision and Avoidance System (TCAS II) (Federal Aviation Administration 2011) has been required in many countries (including the US, Europe, China, Australia and India) on all large commercial transport aircraft. TCAS is designed to reduce the chances of mid-air collision with other transponder or TCAS equipped aircraft by issuing advisories and threat resolutions to the TCAS equipped aircraft. It does this by 1) projecting the current track of any nearby aircraft (*intruder*) into the future, 2) recognizing if this poses a threat of a Near Mid-Air Collision (NMAC), and 3) issuing a resolution advisory to avoid the NMAC. If both aircraft are TCAS equipped, the advisories are automatically coordinated to ensure that the responses do not conflict. Figure 2 illustrates the scope of TCAS reasoning and advisories.

While TCAS has reduced the incidence of NMACs for larger aircraft, there are a number of limitations with the system. First, the cost, hardware and power requirements limit installation to larger aircraft. Second, the system projects that the intruder will continue on its current course. While this may be a reasonable assumption for en-route aircraft operating at higher altitudes, it is less accurate, and less robust

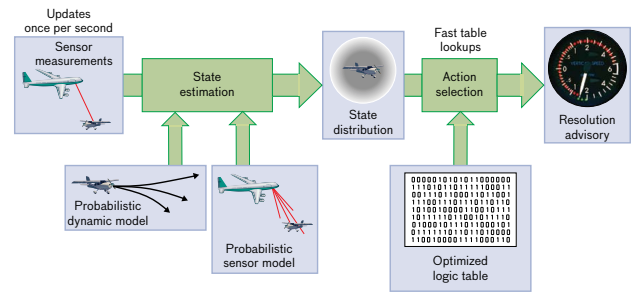


Figure 3: Architecture of ACAS X, from (Kochenderfer *et al.* 2012).

within the terminal area, particularly for smaller intruders operating VFR at low altitudes. Third, the resolution logic consists of hand generated heuristic rules, which are incomplete and difficult to verify. Finally, the system is designed for only short-term conflict resolution (25-40 seconds from NMAC). As a result, the resolution advisors are fairly aggressive maneuvers, and are restricted to actions of climbing, descending, and otherwise constraining vertical speed. There are no horizontal avoidance actions.

The ACAS X system (Kochenderfer *et al.* 2012) is a proposed successor to TCAS II that is currently undergoing testing and evaluation. ACAS X improves on several of TCAS II's limitations in particular, it is 1) capable of utilizing positional information from a broader range of sources, 2) it uses a probabilistic model of the intruders likely trajectory, and 3) the resolution logic is based on offline solution of an MDP, and is therefore more systematic, complete, and robust. As a result, ACAS X has demonstrated the ability to resolve more conflicts, while issuing fewer unnecessary resolution advisories. The architecture of ACAS X is shown in Figure 3. Unfortunately, ACAS X is still aimed at short-term conflict resolution, and the advisories are still limited to climbing, descending, and constraint of vertical speed.

Approach

In this work, we are interested in addressing medium-term conflict avoidance for small UASs operating VFR at low altitude in a mixed air traffic environment. By medium-term we mean detecting and resolving conflicts between 30 and 120 seconds before they would occur. By doing this, we can maintain greater separation between aircraft, and conflicts can be resolved using less aggressive and less costly maneuvers. On the negative side, it is more difficult to accurately predict the trajectory of another aircraft this far in advance. The focus on VFR operation at low altitudes also presents challenges: 1) trajectories are less predictable for VFR traffic, and 2) terrain and weather may be a significant factor in predicting those trajectories and resolving conflicts – notably, aircraft operating VFR are required to maintain certain clearances from clouds and terrain, both vertically and horizontally.

We regard the problem of predicting the future trajectory of an aircraft as probabilistic plan recognition. Conflict res-

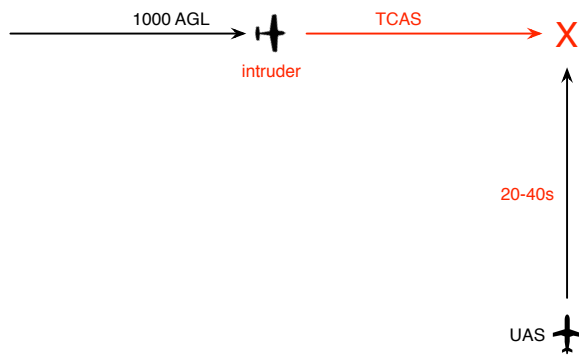


Figure 4: Intruder Prediction for TCAS II.

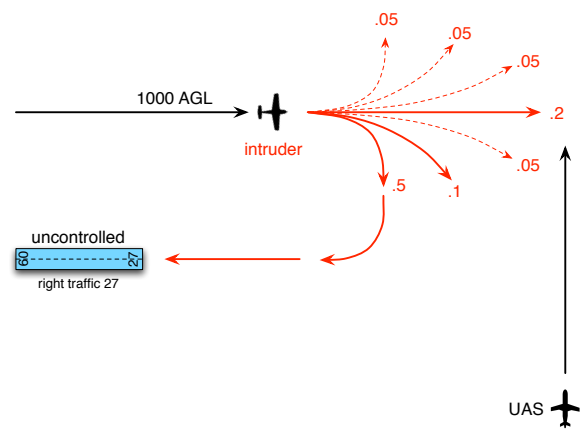


Figure 6: Intelligent threat prediction.

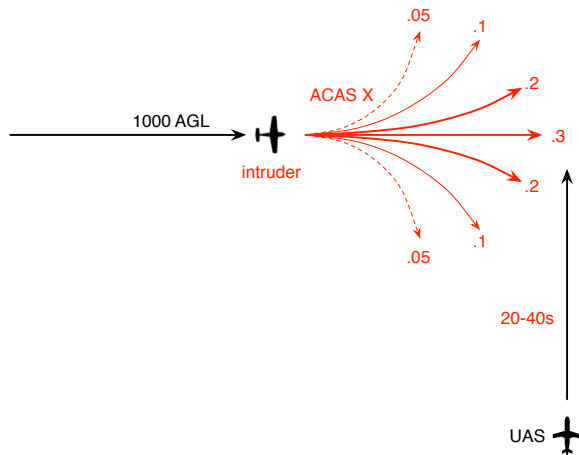


Figure 5: Probabilistic Intruder Prediction for ACAS X.

olution is a problem of path planning under uncertainty. We believe that both of these are knowledge intensive processes that rely on more than just the previously observed trajectory of the aircraft. In particular, prediction needs to take advantage of information about such things as aircraft type, local routes and traffic patterns, transponder operation, communications, terrain and weather. Conflict resolution needs to take account of many of the same things as well as the goals of the UAS.

To illustrate this, consider the simple example shown in Figure 4. Here, a UAS and another aircraft (the intruder) are both at 1000 ft AGL (above ground level) on intersecting courses. TCAS II would predict that the intruder would remain on its present (straight and level) course resulting in a conflict at the point X. Somewhere between 25 and 40 seconds before the intersection, the TCAS logic would advise an aggressive climb to avoid the intruder. Figure 5 shows the difference for the ACAS X model: here there is a probabilistic model of the possible future positions of the intruder (derived from real data of aircraft tracks).

In Figure 6 there is one piece of additional information available – the intruder aircraft is downwind from Runway 27 at an uncontrolled airport. In this case, if the aircraft is level or descending it is much more likely it is in the traffic

pattern for this airport, and will be turning on a base leg, and then final leg as shown in Figure 6. As a result, the probability distribution for the future location of the aircraft should be strongly biased towards turning base. If we have additional information that the aircraft is descending, or that the landing gear is down, then it is even more likely that the aircraft will turn base and there will not be any conflict. In contrast, if we know that the aircraft is climbing, or that the landing gear is up, or that it has a discrete transponder code then it is more likely that the aircraft is on a downwind departure from the airport, and will not be turning base. Likewise, if the wind is strongly favoring the opposite runway (Runway 09), or the traffic pattern is on the south side of the airport rather than the north side (left traffic instead of right traffic), or the runway is too short for this type of aircraft, then it is even less likely that the aircraft will turn base; in fact, it is unlikely that the aircraft is in the traffic pattern for this airport at all.

What’s happening here is that information about local routes, aircraft type, wind conditions, and other factors can have a huge influence on our prediction of the track of the intruder. Terrain, thunderstorms, and airspace restrictions, can have a similar effect. All of these factors are evidence that needs to play a role in the prediction problem.

The rest of the paper is structured as follows: we first describe our system architecture for medium-term threat detection using probabilistic plan recognition techniques, and describe the current status of our implementation and integration with the ACES traffic simulation environment. We then discuss our intended future work on: 1) improving our prediction algorithm by automatically generating dynamic Bayesian Networks tailored to the location of the intruder; and 2) developing medium-term threat resolution software utilizing fast online probabilistic path planning.

System Architecture

Sense-and-avoid procedures generally involve the following steps: 1) **intruder detection** – identify nearby aircraft, 2) **threat detection** – predict the trajectory of potential intruders and decide if any may cause a threat to the intended flight

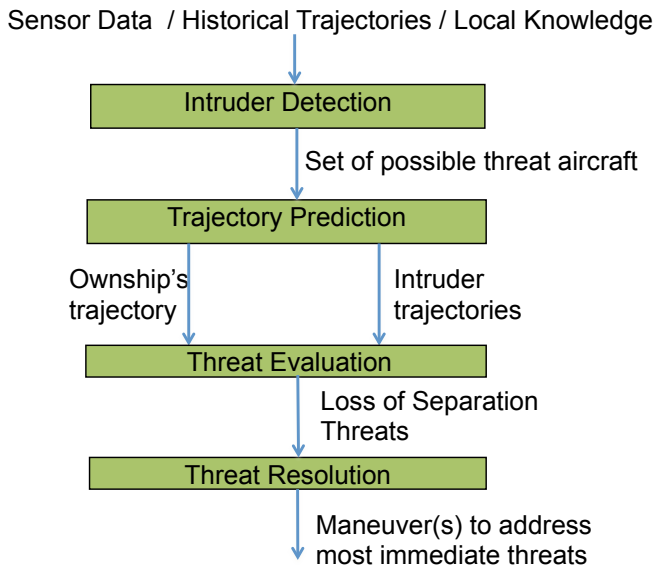


Figure 7: SAA Software Architecture

path, 3) **threat resolution** – devise an appropriate course of action to avoid the threat.

Figure 7 shows how these different components work together to provide the pilot/operator with a set of evasion maneuvers whenever a threat is detected. The Airspace Concept Evaluation System (ACES) (Airspace Systems Division, NASA Ames Research Center) is a simulation environment for the National Airspace that has been developed at NASA Ames Research Center. ACES includes an example SAA implementation called GenericSAA which follows this architecture. Since GenericSAA constitutes a working implementation that is compatible with our conceptual model of the problem, we have adopted it and enhanced it to fit our approach.

The **intruder detection** component is relatively straightforward, as it consists of applying specific vertical and horizontal separation thresholds for any aircraft detected in the vicinity of the ownship. In some cases any detected aircraft will automatically be considered an intruder, depending on the capabilities of the sensors available to the aircraft.

The **threat detection** component consists of **trajectory prediction** and **threat evaluation**. We have formulated a probabilistic approach that predicts the trajectories of potential intruders given their historical paths and relevant knowledge about the aircraft and the region. Figure 8 shows the components of our trajectory prediction framework, which feeds into the threat evaluation component:

Leg Extraction: unlike existing algorithms that utilize only the intruders observed trajectory as points to help predict short-term future trajectory points, we go one step further in extracting trajectory legs. A trajectory leg is a continuous portion of the trajectory where the aircraft is in a particular mode such as: climbing, turning right, flying straight, etc. A sequence of legs reveals the flights pattern and helps

predict subsequent legs. We have built new software to extract legs from the observed state information for an aircraft based on three state variables: 1) speed; 2) altitude; and 3) heading.

Route Database: To understand the intent of each trajectory leg, we match it against defined route segments in the area. For example, if an observed aircraft is downwind for an active runway, is at traffic pattern altitude, and is level or descending, its very likely that it will subsequently turn on a base leg for the runway. We constructed a database of legs for traffic patterns, instrument approach procedures, and airways for an interesting test area near Sacramento, CA that contains several airports and a mix of different kinds of traffic. Each leg is described by its name, type, start and end fixes, altitudes for those fixes, heading, lateral tolerance, altitude tolerance, and heading tolerance. These tolerances are much tighter for something like an instrument approach than for something like a federal airway, or crosswind departure.

Segment Matching: we developed and built an algorithm to match the trajectory legs, created from the observed history of an intruders trajectory, to the segments in the route database. Our matching algorithm decides that a leg L could be a given route segment S in our database if all points in L 1) are located within the geometric region defined by Ss start/end points and tolerances, and 2) have heading bounded by Ss heading and heading-tolerance values.

Predicting Future Intruder Trajectory: we use a Dynamic Bayesian Network to infer the probabilities of different options that the intruder might pursue next. We constructed our Bayesian Net utilizing the open source JavaBayes package. In addition to any route segment matches for the observed trajectory legs (described above), the Bayesian Network has variables representing many pieces of information that might be available about the intruder and area: 1) VFR or IFR; 2) transponder code; 3) aircraft characteristics (e.g. aircraft size, wing type, tail type, number of engines, engine type, observed gear position); 4) communications on relevant frequencies, 5) ceiling and visibility, and 6) wind direction and intensity. Feeding this information as evidence into the Bayesian Network, we are able to infer the probabilities of possible next legs, or other actions by the intruder.

Threat Evaluation: Our trajectory prediction algorithm returns multiple possible medium-term future trajectories with associated probabilities for each aircraft. Threat Evaluation consists of taking these trajectory predictions and detecting possible loss-of-separation violations during the time horizon for the SAA algorithm (30-120secs in our case). We carry this out by performing a fine-grained (1sec) discretization of all of the probabilistic trajectories, mapping them to a 2D data structure (ignoring altitude) and reporting any possible overlaps. We currently inherit the GenericSAA approach, which filters out aircraft flying at different altitude form the ownship's, this simplification eventually needs to be removed to account correctly for intruders that are rapidly climbing or descending.

Threat Resolution: The threat evaluation component returns a set of probabilistic trajectories that potentially violate loss-of-separation thresholds within the time horizon for the SAA algorithm. Threat Resolution consists of com-

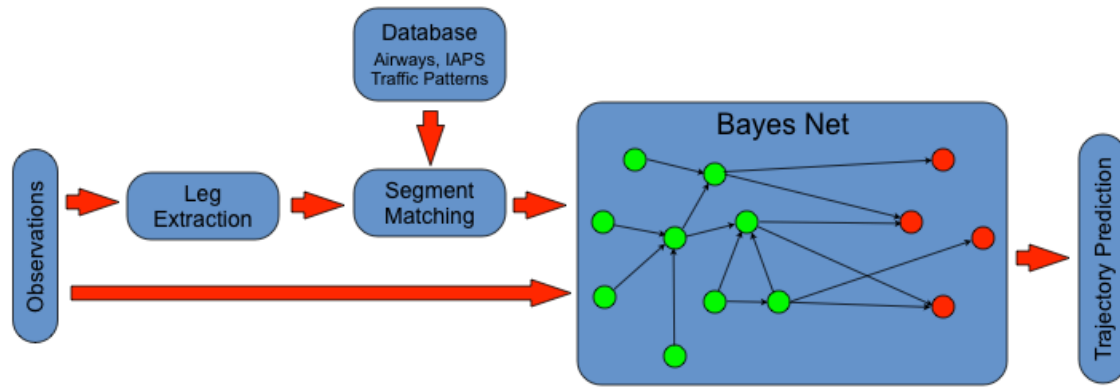


Figure 8: Trajectory Prediction Architecture

puting aircraft maneuvers which will steer the ownship clear of the trajectories that pose threats. This is typically done by choosing from a set of maneuvers that eliminate conflicts and optimize some risk/benefit function. We propose a probabilistic threat resolution approach explained in detail in a subsequent section. Currently, we reuse the approach implemented by the GenericSAA framework, which has the following characteristics: 1) Only the most immediate threat is addressed. 2) Resolutions are computed by examining a small set of maneuvers that modify one of {Horizontal Position, Altitude, Speed}, in that order. 3) The first maneuver that yields a conflict-free trajectory projection is chosen as the solution, no optimization is performed at this point.

We have implemented most of the elements in our Trajectory Prediction architecture: Leg Extraction, Segment Matchings, and Route Prediction using a Dynamic Bayesian Network. For the time being we have manually created Route Database entries for a small test area near Sacramento, CA. We have integrated our probabilistic trajectory prediction with ACES' GenericSAA framework. GenericSAA provides Threat Evaluation and Threat Resolution functionality and therefore allows us to evaluate our approach in the context of a complete SAA solution. For verification purposes we have also built a lightweight simulation and visualization mechanism (Figure 9) that allows us to 1) Set up intruder and ownship flights and 2) Step through time to incrementally visualize the aircraft states, projected trajectories, computed threats and computed threat resolution maneuvers.

Improving Probabilistic Path Prediction

As described in the previous section, we developed a prototype dynamic Bayesian Network model that takes advantage of knowledge about aircraft, routes, traffic patterns, topography, airspace, weather information, and observed communications in order to better predict the path of an intruder aircraft. While this additional information can be very powerful, the states in the Bayesian network must be tailored to the specific area of the intruder. For example, if the intruder is near a particular airport, then possible next states for the

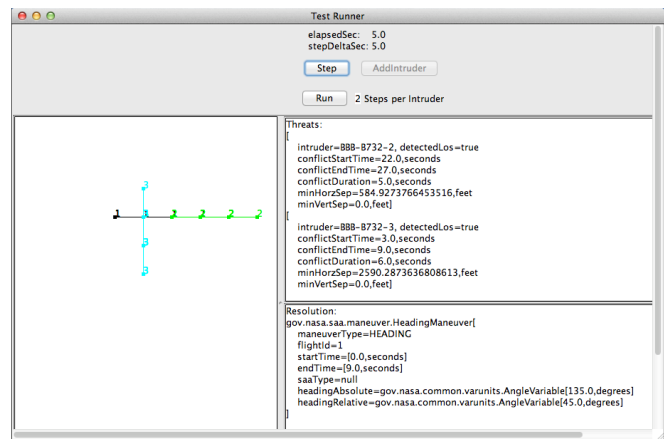


Figure 9: SAA Simulator

aircraft may include traffic pattern legs or approach legs for that airport. However, if the aircraft is at 4000 ft AGL, then those legs are not relevant as possible next states for the aircraft. In our current prototype, we dodged this issue by including all possible states for the local area whether or not they were reachable by the intruder within the time window of interest. This resulted in the current and next state nodes in the Bayesian network having more than 30 possible values. As a result, the conditional probability tables for some of these nodes required more than 1000 entries. Although most of the entries in this table are zero, it is impractical to generate these tables by hand for anything other than a limited geographical area. In addition, the conditional probability tables are not very compact or easy to understand. Based on our experience, we believe that it is possible to encode this information much more succinctly, and automatically generate the Bayesian network for a particular location. The key to this is to recognize that traffic patterns and procedures at an airport can be encoded as probabilistic automata. For each leg in the procedure there are a set of possible successor legs with different probabilities. Those probabilities are

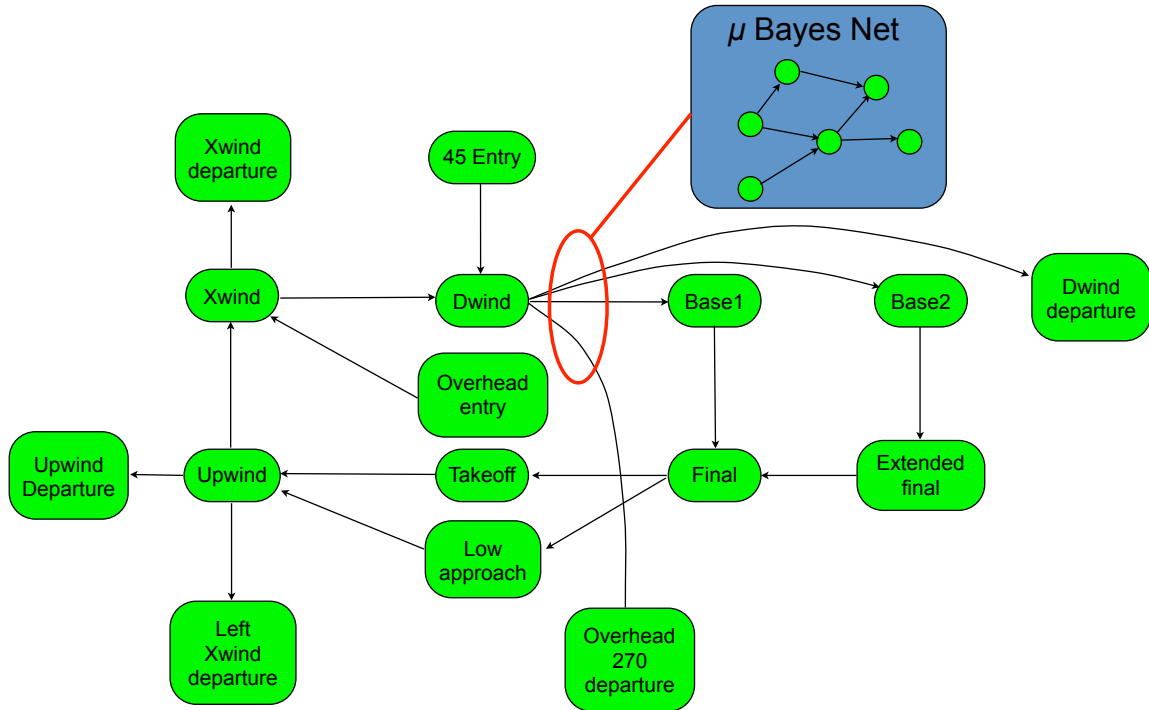


Figure 10: Bayesian Network capturing probabilistically the relation between different characteristics of the intruder aircraft and how they can be used to infer the probability of what the intruder may do next.

influenced by factors such as aircraft type, airspeed, vertical speed, transponder code, etc. As a result, each set of transitions for a leg form a small dynamic Bayesian network. Figure 10 provides an illustration of what this automata looks like for a typical airport traffic pattern. An aircraft remaining in the pattern (practicing landings) would transition from Takeoff to Upwind to Crosswind (Xwind) to Downwind (Dwind) to Base to Final. There are multiple possible transitions at many of the nodes. For example, from downwind, the aircraft could transition to a close-in base (Base1) a more distant base (Base2), could depart downwind, or do an overhead 270 departure. The probabilities of these different options are influenced by other factors, such as aircraft type, speed, climb/descent, altitude, transponder code, communications, etc. We believe that most of this information can be encoded generically for traffic patterns, approaches, departures, and airways, and that we can automatically generate the appropriate dynamic Bayesian Network for any specific location.

We recognize that our approach to “plan recognition” is rather specialized for aircraft trajectories. We have considered recent approaches such as that of Ramirez and Geffner (2010). The trouble is that for this problem there is little ability to predict the long term goal of the aircraft. Instead, we are interested in predicting the next plan or trajectory steps. In addition, the computational overhead of the Ramirez/Geffner approach is substantial and would probably be impractical for this domain. One can, however, think of the dynamic Bayesian Network as encoding the informa-

Lateral resolution actions	Vertical resolution actions
Standard rate left turn	Climb 1000 ft/min
Half standard rate left turn	Climb 500 ft/min
Straight	Level
Half standard rate right turn	Descend 500 ft/min
Standard rate right turn	Descend 1000 ft/min

Table 1: Resolution action.

tion that would be obtained from an approach like that of Ramirez/Geffner – the probability of each possible next plan step is influenced by the previous actions of the aircraft, as well as the probability of the possible goals.

Medium-term Probabilistic Threat Resolution

Systems such as TCAS and ACAS X are designed to detect and avoid near term conflicts between aircraft conflicts that would occur in a time frame between zero and 45 seconds. If a Near Mid-Air Collision (NMAC) is imminent (within 25 seconds) they issue a resolution advisory that involves climbing, descending, or constraining vertical speed (e.g. maintain climb, or do not descend). Because these systems deal with near term critical situations, the resolution advisories can be dramatic in nature, and can be disruptive and inefficient for the intended flight path of one or both aircraft.

It is clearly desirable to have medium or longer term conflict detection and avoidance, so that near-term conflict

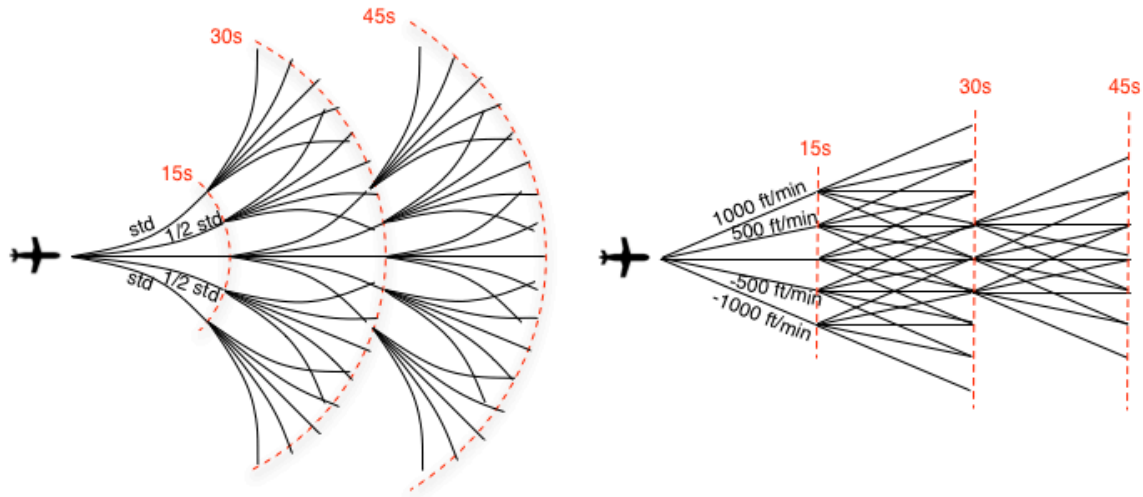


Figure 11: The lateral and vertical action spaces for threat resolution. For lateral maneuvers, standard and half-standard rate turns are considered every 15 seconds resulting in heading changes of 45 and 22.5 degrees respectively. For vertical maneuvers, 500 and 1000 ft/min climb and descent are considered every 15 seconds.

avoidance can be minimized. In particular, we would like to recognize potential conflicts up to two minutes in advance, resolve them using less abrupt maneuvers, and keep the aircraft well clear of each other. Furthermore, for smaller aircraft and UASs operating at lower altitudes, climbing and descending is often not the best approach for resolving conflicts—descent may be limited by terrain or obstacles, and climb may be limited by performance, clouds, or airspace. As a result, most conflicts between VFR aircraft are resolved by heading changes to go around or pass behind the other aircraft.

Our objective is to develop and test an algorithm for medium-term conflict resolution—conflicts that are predicted in the time frame between 30 seconds and 2 minutes. We assume a probabilistic model of the possible pose (location, airspeed, vertical speed, heading, and turn rate) for the intruder aircraft as a function of time. In particular, the model developed and described earlier may predict specific future actions with high probability for the intruder based on inference that the intruder is on a particular route, approach, or traffic pattern, or based on weather, terrain, or airspace constraints. We intend to consider both lateral resolutions (e.g. standard rate left turn), and altitude resolutions (e.g. descend 500 ft/min) as shown in Table 1.

The vertical resolutions in Table 1 are much gentler than those considered by TCAS and ACAS X for two reasons: 1) most small aircraft are limited to sustained climb rates between 1000 ft/min and 1500 ft/min, and 2) because of the longer time horizon, climbs and descents do not need to be aggressive. From the initial state of our aircraft, each vertical and lateral possibility will be considered at each time step, which we will take to be every 15 seconds. Given that there are 6 time steps from 2 minutes until 30 seconds, and 25 action combinations at each time step, this results in a

state space of 25^6 or approximately 250 million states. However, the vertical and lateral spaces can be considered independently, resulting in two spaces of approximately 16,000 states each, a much more manageable number. An illustration of the lateral and vertical state spaces are shown in the Figure 11.

There are costs associated with the different actions that the aircraft can take. In particular, there is a cost associated with initiating each new action (other than none), and more aggressive actions cost more than less aggressive actions. Given the distribution of possible projected locations for the intruder, each of the leaf states in this state tree can be given a value based on the clearance from the intruder, and the cost of returning to the original route. Using dynamic programming, this information can be backed up through the state space to find the best action at each time step. This approach, is similar to the approach taken in ACAS X, but there are some differences:

1. The action space is quite different—we are considering both lateral and vertical actions, but the actions are less aggressive.
2. The time horizon is much longer, but we use much coarser time steps because the conflict is farther off.

However, the most critical difference is due to our knowledge intensive approach to prediction of the course of the intruder. Because the probability distribution for the intruder is influenced by location (routes, weather, terrain), the values on the leaf nodes of the MDP are dependent on the particular situation. As a result, the MDP cannot be solved in advance and compiled into a decision table as in ACAS X—instead it must be solved in real time. Given the size of the MDP, we believe that this can be done in a few seconds. If this turns out not to be the case, we can resort to approximate policy generation methods for solving the MDP, which

would reduce solution time. Because of the medium-term time frame, optimal solution of the MDP is not as important as for ACAS X.

One final difference between this approach and that of ACAS X is that the value of each leaf node is also influenced by the cost of returning the UAS to its intended course. Deviating more from the intended course increases the penalty, and this must be balanced with maintaining clearance from the intruder. There is no consideration of this issue in TCAS or ACAS X.

Conclusion

There is growing demand to operate UASs in civilian airspace. Enabling this will require much better sense-and-avoid technology, so that UASs of vastly different sizes and capabilities can maintain separation from other aircraft. While most current research has concentrated on near-term collision detection and avoidance utilizing only the observed trajectory history of an aircraft, we target mid-term detection and avoidance utilizing a variety of information beyond just aircraft trajectory.

We concentrated on two tasks: (1) conflict detection: utilizing probabilistic path recognition; and (2) conflict resolution: utilizing fast online probabilistic path planning. Our work is preliminary. We have implemented the components necessary for the conflict detection portion and integrated it with the ACES traffic simulation system. We are refining and improving the dynamic Bayesian Network prediction and plan to implement our probabilistic conflict resolution approach. We hope to evaluate the techniques using a large database of actual air traffic trajectories that have been collected and made available through related efforts at NASA.

Acknowledgements

We would like to thank Eric Mueller for helping us hone our ideas on this project. We would like to thank Confesor Santiago for help with the ACES simulation environment, and the GenericSAA module developed for ACES. This work was supported by the NASA Aeronautics Research Institute (NARI).

References

Airspace Systems Division, NASA Ames Research Center. Airspace Concept Evaluation System (ACES). <http://www.aviationsystemsdivision.arc.nasa.gov/research/modeling/aces.shtml>.

Won-Zon Chen, Leo Wong, Jacob Kay, and Vincent Raska. Autonomous sense and avoid (SAA) for unmanned air systems (UAS). In *Systems Concepts and Integration Panel (SCI) Symposium*, volume RTO-MP-SCI-202. NATO Science and Technology Organization, 2009. <ftp://ftp.rta.nato.int/PubFullText/RTO/MP/RTO-MP-SCI-202/MP-SCI-202-28.doc>.

Federal Aviation Administration. Literature review on detect, sense, and avoid technology for unmanned aircraft systems. Technical Report AR0841, U.S. Department of Transportation, 2009. <http://www.tc.faa.gov/its/worldpac/techrpt/ar0841.pdf>.

Federal Aviation Administration. TCAS II, Version 7.1. Technical Report HQ 111358, U.S. Department of Transportation, 2011. [http://www.faa.gov/documentLibrary/media/Advisory_Circular/TCAS II V7.1 Intro booklet.pdf](http://www.faa.gov/documentLibrary/media/Advisory_Circular/TCAS%20II%20V7.1%20Intro%20booklet.pdf).

Mykel Kochenderfer, Jessica Holland, and James Chryssanthacopoulos. Next-generation airborne collision avoidance system. *Lincoln Laboratory Journal*, 19(1):17–33, 2012.

Miquel Ramrez and Hector Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, pages 1121–1126. AAAI Press, 2010.