AUTONOMOUS AIR TRAFFIC CONTROL – COLLISION AVOIDANCE FOR UAVs USING MDP

Ravi Vikranth **DABBIRU**
Department of Computer Science
SVP College of Engineering
Visakhapatnam, INDIA

Kartik ***SIDDHABATHULA***
Assistant Professor, Department of Computer Science
SVP College of Engineering
Visakhapatnam, INDIA

Abstract— *Unmanned aerial systems (UAS) are the future of modern aviation. The increasing demand forecast and potential for commercial exploitation of Unmanned Aerial Vehicle (UAV) technology inkling towards integrating UAV technology in non – segregated airspace. In order to overcome technological issues; standard framework has to be designed addressing the "Detect – Sense and Avoid (DSA)". Obtaining level 3 situation awareness is also a fundamental need; for safe and efficient Air Traffic Control (ATC) and Command and Control (C2). Uncommon UAV missions types and uncommon flight dynamics, demand additional effort of air traffic controllers. The control methods are Proportional-Integral-Derivative (PID) and sliding mode control. PID will be tested on both heading and pitch and altitude control, while sliding mode will only be applied to pitch and altitude. There will be presented a path-following method, Line of Sight, for heading guidance and a kinematic controller for altitude reference. To effectively plan collision avoidance paths a projected flight path planner is developed. Techniques for predicting the intruder position and creating safe, collision-free paths using the estimates provided by the flight co-ordinates using MDP are presented. A method for calculating the cost of flying each path is developed to allow the selection of the best candidate path. As multiple duplicate paths can be created using the branching planner, a strategy to remove these paths and greatly increase computation speed is discussed.*

Keywords- **UAS, UAV, Air Traffic Control,Command and Control, PID**

I. INTRODUCTION

Military UAV's are used to perform reconnaissance, surveillance, target identification and more often strike missions. With UAVs, the militaries around the world acquired the ability to operate in high threat environments without putting men and machinery at risk. This is not only safer, but potentially also more effective. Some of the Civilian applications of UAVs are in remote sensing of the environment, geographic surveys, border patrol, transport of goods, and search and rescue. Today, over 32 countries are developing UAVs. The most sophisticated military systems are designed by the United States (US) and Israel, while Japan has the most UAVs in operation for agricultural spraying.

The UAV and the UCS communicate over a data link, either over a radio connection or via a satellite relay. UAV developments in the last decades have enabled autonomous operations like Fly back Home, Mid Air Explosion, Drop Zone Failure etc. As autonomy increases; human intervention is dramatically reduced; though his/ her role is reduced to supervision and role allocation.

The crux of the paper lies in operating uavs in non-segregated airspace. In order to integrate uavs in non-segregated airspace, effort is needed on several regulatory and technology issues. The us and europe work jointly on standardized uav airspace integration regulations with their respective air traffic management (atm) strategies.

II. EASE OF USE

A. AIRSPACE INTEGRATION

In order to integrate UAVs in non-segregated airspace, effort is needed on several regulatory and technology issues. The most relevant regulatory issue is airworthiness certification. Current UAVs have different capabilities than manned aircraft. Hence, they do not fit current classification schemes in airworthiness certification. There are a variety of classification schemes for UAVs, like schemes based on operating altitudes, operational characteristics, mass and speed. None of these have been adopted by the aviation authorities yet. In the US, a UAV considering a flight in national airspace still needs individual approval. This approval process, known as a Certificate of Authorization (COA) and contained in the Federal Aviation Administration (FAA) Order 7610.4, requires the case-by-case safety evaluation of each flight

The main technology issue is related to the definition of the see-and-avoid requirement in manned aircraft, that cannot be applied to the UAV counterpart. See-and-avoid partly relies on the human eye. Moreover; Detect, Sense and Avoid (DSA) functions can enable a UAV to respond autonomously and safely to a collision threat in a manner equivalent to pilots of manned aircraft. However, current technology in manned aircraft is limited, e.g. using Traffic Collision Avoidance System (TCAS) II for self-separation is considered unsafe in a UAV.

B. OBJECTIVE and SCOPE

Today's Air Traffic Control (ATC) and Command and Control (C2) have not been designed with SWIM-based capabilities. Although global strategies aim to achieve coordination and synchronization between UCS and other systems, the UCS also lacks the ability of sharing information widely.

One of the developments is conflict probing, which is defined as predicting the future separation between friend and foe for a set of friend system velocity vectors - representing possible combinations of track, flight path angle and velocity. Conflict probing supports UAV operators to obtain level 3 situation awareness with a better conflict detection capability, and with a limited false alarm rate. Obtaining level 3 situation awareness is also a fundamental need for safe and efficient ATC. Uncommon UAV mission types and uncommon flight dynamics, demand additional effort of air traffic controllers2. Conflict probing might compensate for this additional effort in a different way.

With conflict probing, no external information is involved. Conflict probing involves the yet available state information and presents a spatially integrated depiction of areas to avoid. This hands on information about the areas where a loss of separation occurs can directly support a controller with his separation task.

To make a clear distinction between different concepts in situation awareness and conflict scenarios, the most relevant concepts are defined. Within situation awareness, the following three concepts are distinguished: traffic awareness, future conflict awareness and conflict awareness. The threesome is classified into three awareness levels: detection, integration and anticipation. Obviously, the higher goal to provide support for traffic awareness is to prevent conflicts. For conflicts specifically, a distinction in phase is made between the situation where a conflict exists; and the situation prior to a conflict.

III. PROBLEM FORMULATION

The present work addresses the problem of distributed collision avoidance among autonomous unmanned aircraft using multi-agent technology. The approach proposed in this paper involves collision avoidance between cooperative agents using peer to peer negotiation, flight plan sharing and predefined maneuvers, and in this regard it is closely connected to recent approach proposed in. But with the advantage of using cooperative resolution maneuver through negotiation and with different detection mechanism. This will results in a more efficient conflict resolution approach with less flight path maneuvering cost.

Due to the simplicity of the approach, solution may be obtained quickly allowing our approach to be implemented in time critical requirements. The work considers the problem of a finite number of autonomous aircraft sharing the same airspace. That is, only speed changes and/or turn commands are allowed. The task of each UAS is to reach a final configuration from a given start waypoints positions avoiding all possible conflicts.

This paper focuses onto the development of a new functional architecture for unmanned aircraft CAS and device approach utilized for determining the possibility of collision and providing the maneuvering command to the aircraft in the presence of a potential conflict caused by another cooperative aircraft agent. The proposed functional architecture provides a flexible and an extensible CAS with a mechanism to easily integrate and use it on cooperative UASs to manage them to avoid colliding with each other. The resolution process between two or more cooperating UASs is based on agent-to-agent negotiation protocol.

A. STATES OF THE AIRPLANE

The plane location – this is recorded in a three dimensional co-ordinate system

a. The flight direction (X-axis : East, Y-axis : North)
b. The flight speed
c. The plane ascends/ descends
d. The plane intention to land or their status in holding pattern.

The two main functions of our ATC agent are to detect and prevent collisions and to process landing. Agent will display dialog boxes for collision detection and when there are requests for landing. The windows will stay open until either it is timed out or the human operator overrides the agent decision. Our agent maintains three queues viz.;

a. landing request queue – along with a request to land, each plane generates a landing priority between 0 to 4. ‘0’ is the default priority having no unusual circumstance while 4 is the highest priority number corresponding to the most urgent emergency. The landing request queue is arranged with the ascending priority so the most urgent priorities are handled first.
b. Collision queue – planes near enough to pose a collision threat are assigned a priority from 1 to 4. I signifies near miss; whilst, 4 signifies a collision that is about to happen.
c. Holding Pattern Queue – maintains the order of planes that have re quested landing but due to high air traffic are in the holding pattern.

The ATC agent maintains a combined priority queue of all planes in its radar screen using the range of priorities from 0 to 4. ATC agent may use the planes landing priority when there are no other concerns. However, the ATC agent will use other cues for reprocessing its priority queue. In general, the following are the factors that contribute to the agents determination of priority;

a. planes requests and landing priority
b. perception of possible collision threats
c. projection of paths for planes initially detected to be on a collision course
d. landing requests from different pilots with priorities that are confliction with prior requests
e. planes in various landing stages
f. status of planes in holding pattern

The agent will issue one of the following commands;

a. clear to land
b. go to holding pattern
c. come out of holding pattern
d. avoid a collision by a maneuver (turn, ascend, descend)

B. COLLISION QUEUE

Distance between every plane is checked against every other plane in the radar range. Distances are checked in the horizontal plane and then comparisons are made in the vertical axis. There are two different sets of collision ranges for determining a collision threat. One range is for flight in the air space before they request landing.

This range is two miles on the xy axis and a 1000 feet between planes in the z-axis. The second range is when the plane or planes are going to be landing or in holding pattern. This range is one mile apart on the xy-axis and 500 feet between planes in the z-axis. If the separation distances are violated, the program will assign a number to correspond to the severity of collision threat. Collision threat is a priority number similar to landing priorities. The number range from 1 to 4. The following are the general flight conditions;

a. 1- designates very low likelihood of a collision; the planes have reached the boundaries of collision concern i.e. if the planes are 2 miles apart on the xy axis and almost 1000 feet apart on z-axis.
b. 2- if the planes are 1 mile apart on the xy –axis and about 500 feet on the z-axis
c. 3- if the planes are ¾ mile apart on the xy-axis and about 400 feet on the z-axis
d. 4- if the planes are ½ mile apart on the xy axis and about 300 feet apart on the z - axis

C. *PRIORITY QUEUE*

The priority queue contains the planes in the descending order so the situation with the highest priority is addressed before the situations with lower priority e.g. if there is a collision and a near miss, the agent will attend to the collision before solving the near miss.

If the distance between two planes is within 1 mile on the xy – axis and 500 feet away, the agent will command one of the planes to ascend or to descend. When the distances are closer than 1 mile and 500 feet the agent will command the planes to turn left or right.

The timer is very important, if the air traffic controller does not handle the problem right away there on could be an accident. So with the timer insures that either the human operator or the agent will handle the problem. The timer will be set to the severity of the collision threat. The closer the two planes are the less time will be used on the counter.

D. *THE LANDING and HOLDING PATTERN QUEUE*

There are five different stages in the landing flight plan. A stage is a certain part of a flight plan for landing the plane. The following are the stages;

a. **Stage 0** – receives confirmation that it can proceed to land. The plane flies to the east side of the airport and aligns with a line that is heading toward south of the airport
b. **Stage 1** – when the plane is on southward line and heading south at about 170 degrees
c. **Stage 2** – when it turns right to start heading back to the runway. It will maintain stage two until it reached a heading of 350 degrees, at which point it will be lined up with the runway.
d. **Stage 3** – lined up with the runway.
e. **Stage 4** – when the plane is on its final approach to the runway and it starts descending for landing. Once the plane is on the ground, it would leave stage four and taxis over to the tarmac to park.

Note: only one plane can be landing at a given time and will be given permission to land. All other requests for landing will receive a command to enter holding pattern.

Holding Pattern:

Holding pattern is a circular flight at a specified altitude. The agent maintains a queue of planes in the holding pattern.

a. the agent maintains a queue of planes in the holding pattern.
b. The agent monitors progress of planes along landing stages.
c. The agent maintains a list of planes in the 5 flight stages.

E. *Landing Request Procedure:*

The following is the standard procedure adopted;

a. once the agent receives a request to land, the agent will check to see if they can land
b. it will check to see if any other planes are landing at that time
c. if the plane is in stage 0, 1 or 2; the agent cannot send another plane to land. It will have to wait until the plane that is landing which is past stage two.
d. The agent will have to put the planes into a holding pattern around the tower until the landing path is clear
e. If the first three stages are clear, then the agent can tell the requesting plane is clear to land
f. A plane in the holding pattern is given permission to land when the current landing plane enters stage three and is popped from the holding pattern queue.

F. *Combined Queue:*

The combined queue is constructed by merging the three queues in the ascending priorities;

Example:

Landing request queue :

ND239D -> wants to land

ND1002 -> also wants to land but medical emergency

With medical emergency. Hence, priority

2.0

Landing Queue : ND1002(2.0)/ ND239D(0)

Planes MN3230 and SD392 are about 1.32 miles apart along XY-axis and 950 feet on Z – axis, and the planes are getting closer in the next second. The situation will be assigned a priority number 2.4.

Collision Queue : MN3230/ SD392(2.4)

The queue for holding pattern is empty since there are no planes waiting to land

Holding Queue : NIL

After merging the queue will have the following

Combined Queue : **MN3230/SD392(2.4) ND1002(2.0) ND239D(0)**

The agent will attend to the collision between MN3230 and SD392 then the emergency landing will be given a clear to land, and ND239D(0) will be told to enter holding pattern until ND1002 reaches stage 2 of its landing.

IV. MARKOV DECISION PROCESS

The specific Collision Queue problem above is addressed by navigating a UAV from an initial position (approach/ mid air) to its destination (runway) in the two dimensional plane with capability of avoiding multiple intruders.

a. Flight Dynamics Modeling: In this work, a reduced order non – linear dynamic equation are used to model the aircraft motion. The equations of motion are as follows;

$$\begin{cases} \dot{x} = V_a \cos\Psi \cos\gamma \\ \dot{y} = V_a \sin\Psi \cos\gamma \\ h = \sin\gamma \end{cases} \quad (1)$$

where (x, y) is the position of the UAV with respect to earth co-ordinate system, where the positive x-direction points to EAST and positive y-direction points to NORTH. Va, Ψ, γ and h represent the speed, heading angle, flight pitch angle and altitude of the UAV respectively. Assuming the UAV is flying in two dimensional plane with constant speed, the flight path angle equals to zero. Thus equations of motion can be written as,

$$\begin{cases} \dot{x} = V_a \cos\Psi \\ \dot{y} = V_a \sin\Psi \\ h = \sin\gamma \end{cases} \quad (2)$$

For sake of safely turning the UAV, it is required that dynamic constraints are not exceeded to prevent the UAV into an irreversible state. Therefore, a coordinated turn is considered to describe a turning maneuver. The coordinated turn condition is expressed as;

$$\dot{\Psi} = (g/V_a) \text{ x } \tan\phi \quad (3)$$

Where ϕ and g denote the roll angle for UAV and gravitational acceleration respectively. ϕ is the control input of UAV, which is assumed to be bounded as $|\phi| \leq \phi_{max}$.

b. Formulation: Whether the trajectory of UAV has to be re-planned for avoiding collision or not depends on the feedback of the collision detection. Detecting the conflict is determined by the miss distance at the closest point approach. The conflict detection between the UAV and multiple aircraft is only considered in a common horizontal plane. If the predicted distance between the UAV and the intruder is smaller than the minimum separation distance r_{safe}, within a specific block of time T_{det}, the collision will occur. It is also assumed that intruders follow the initial trajectory without any avoidance maneuver. The velocity of the intruder can be calculated. For the UAV, the closest approach distance, $\vec{d}_m$ can be derived as;

$$\vec{d}_m = \vec{V}_r \text{ x } (\vec{d} \text{ x } \vec{V}_r) \quad (4)$$

Where $\vec{d}$ denotes the vector locating the intruder with respect to the UAV, and $\vec{V}_r$ is the unit vector in the direction of relative velocity vector of the UAV with respect to the intruder;

$$\vec{V}_r = \frac{v_r}{||v_r||} \quad (5)$$

The relative velocity of the UAV with respect to the intruder, $\vec{V}_r$ is obtained by;

$$\vec{V}_r = \vec{V}_B - \vec{V}_A \quad (6)$$

Where v_r is located at the vector direction of the time to the closest point r. with the relation between $\vec{d}_m$ and $\vec{d}$, $\vec{d}_m$ can be derived as;

$$\vec{d}_m = \vec{d} + \vec{V}_r T \quad (7)$$

Thus, it could be derived that;

$$T = (\vec{d} \text{ x } \vec{v}_r)/ (\vec{V}_r \text{ x } \vec{V}_r) \quad (8)$$

[Since, $\vec{d}_m$ and $\vec{V}_r$ are orthogonal and hence $\vec{d}_m x \vec{V}_r = 0$]

The true state space model in the collision avoidance problem is continuous and consists of the following components for both aircraft present in the encounter:

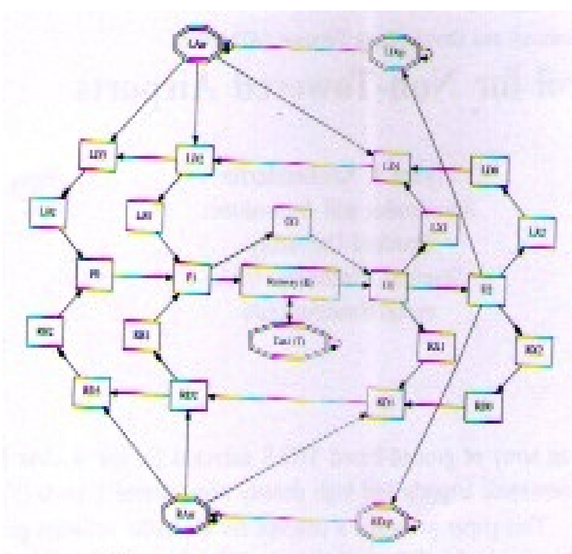
- Position specified in GCS
- Orientation specified as yaw, pitch and roll angles
- Air speed, air speed acceleration
- Vertical rate, vertical acceleration
- Yaw rate, pitch rate and roll rate

This set of components is referred to as the aircraft state vector. This is a very high-dimensional continuous space (26 dimensions for both aircraft together). The action space for a UAV is also continuous as it is possible to choose and apply any vertical and/or horizontal accelerations within friend system's performance limits.

State Space

The size of a discretized state space is exponential in the dimension and in the case of 26 dimensions, it is highly undesirable to have even two discrete values per dimension. Prior to discretizing the state space, the state space must be represented in low dimensional sub – space. To encode relative positions and velocities of the aircraft, we chose RCS as our main representation. In this coordinate system, the state consists of the following components:

X : horizontal distance from friend system to intruder aircraft;
Y : vertical distance from friend system to intruder aircraft;
RelativeVx : (relative) velocity in X , representing the horizontal closure rate;
OtherVy : vertical velocity of intruder aircraft; and
OwnVy : vertical velocity of friend system.



Aircraft states in the pattern.

This 5-dimensional state space is discretized by dividing each dimension into a finite number of bins . The sizes of the bins may be non-uniform. The overall state-space is then a set of 5-orthotopes (5-dimensional boxes or hyper rectangles) that exhaust a continuous piece of the overall 5-dimensional state space. Contemplating the state space with two sets of special states: start states and done states. These states are used to model situations; when the state space is initialized (and the encounter has not started), and when the encounter is over, respectively. Because the vertical velocity of friend system is always known, we always include it in the state space. So, the start and done state sets both contain a member for each bin of OwnVy, modeling flight at some vertical velocity before the start of or after the termination of, an encounter. Having discretized the state space in this way, a state may be represented simply as an index into the set of boxes spanning the space, or an index to one of the start or done states.

Action Space

We adopted a simple discrete action-space model that consists of commands to friend system to apply positive or negative fixed vertical accelerations for a fixed duration (usually 1 s). For the MDP CAS, our action space consists of 17 uniform samples from the ±8 ft/s^2 (± 0.25 g) acceleration range imposed by the aircraft performance limits; A – {-8,-7,…, -1, 0, 1, …, 7 ,8}. It is possible to sample the range of vertical accelerations more densely, but the solvers would require more time to find policies with tight regret bounds

Reward Model

The reward function in our MDP formulation is in the form of costs (or negative rewards) rather than positive rewards. It is designed with the following three objectives in mind

- As the primary goal of the collision avoidance algorithm, the intruder aircraft should never occupy the same bin as friend system in the RCS, which implies a collision or a very dangerous encounter. Note that friend system resides at the origin of the RCS, and it is possible that the origin might be on the edge or vertex of one or more bins rather than being inside a single bin due to the chosen vertical and horizontal division strategy. In which, the collision avoidance algorithm should prevent the intruder from moving into any one of the bins that have any boundaries touching the origin.
- In addition to preventing collision, it is desirable to maintain some protected airspace around friend system where the intruder aircraft should not penetrate. This protected airspace is represented by two parameters: a vertical separation range and a horizontal separation range. The second goal of the collision avoidance algorithm should be to prevent other aircraft moving into any bin that has some parts overlapping with the protected airspace.
- As the last goal, if there is no danger of collision or penetration of protected airspace, friend system should level 'o' and try to maintain a zero vertical velocity. It may be argued that friend system should try to return to its commanded flight path. We have taken the position that, during the handling of a close encounter, it is enough to prefer level flight, and that after the encounter is over, standard navigational procedures can be resumed.

The expected value of taking action a in a state s under policy Π is;

$$Q^{\Pi}_{(s,a)} = E_{\Pi}\{R_t | s_t = s, a_t = a\}$$
$$= E_{\Pi}\{ \quad \}$$

In order to satisfy these goals, the reward may be specified as a function of the state of the system. It is specified using three user-defined parameters:

- **Collision cost:** The cost of any state in which the intruder is in the same X and Y bins as friend system, currently set to – 1000.

- **Protected airspace violation cost:** The cost of any state in which the intruder aircraft is within the protected airspace region in X and Y, currently also set to -1000, and
- **Vertical Velocity Penalty:** The cost for being in a state where the OwnVy bin does not contain 0 ft/s; for the MDP CAS, vertical velocity penalties are linearly proportional to the velocity values that correspond to the centers of the OwnVy bins. It is possible to vary the maximum penalty value in order to reach different equilibrium in balancing evasive maneuvers and level flight.

  All other states are assumed to have a reward of 0. Note that the solution to the MDP will remain the same for any linear scaling of reward values, so only the relative magnitudes have an effect.

State-Transition Model

The initial state distribution specifies that the system starts in a uniformly chosen start state. At each step, an action is taken and the probability distribution over the state space is updated according to the state-transition model. Our state-transition model is characterized by the following parameters:

- Controller frequency, f_T: Duration between successive consultations of the MDP policy for choosing an action. This value is used by the MDP formulation to predict what the state will be in the next iteration.
- Magnitude of our vertical acceleration, OwnAy.
- Our vertical velocity limits, OwnVyMin and OwnVyMax.
- Probability of staying in start state when already in start state.
- Probability of making a transition into any other state when in start state.
- Intruder aircraft's horizontal and vertical acceleration models.

Given these parameters, we compute Pr(s'| s, a) as follows:

- First, we consider each possible pair of vertical and horizontal accelerations a_o that might be chosen by the intruder aircraft, and compute their probabilities p_o as the product of the probabilities in the intruder acceleration models.
- For each vertex of the bin s, we determine how that particular point in state space would be transformed given the execution of friend system acceleration a, and the intruder accelerations a_o.
- The result is a new box, B, in 5-dimensional space. For each new state s0, we compute the percentage of B that overlaps s0; that overlap percentage is $Pr(s'| s, a, a_0)$. Any probability mass outside the boundaries of the modeled state space is assigned Pr(DONE, Own V_y |s, a, a_o).
- Finally, $Pr(s'| s, a) = \Sigma_{a0} Pr(s'| s, a, a_0) p_0$

This method of computing the physical evolution of the system analytically eliminates introducing additional discretization in the computation. Therefore, the effectiveness of the state-transition model depends only on the discretization of the state and action spaces and the fidelity of the vertical and horizontal acceleration models for the intruder aircraft.

COOPERATIVE MANEUVER REALIZATION:

The possible maneuver dimensions include Turns: straight (no-change), left and right; and Speed changes: speed-up and slow-down, which simply require the aircraft to fly the previously intended path at slightly faster or slower speeds. The cooperative resolution maneuvers are fixed during system design based on a set of predefined procedures, proven to be safe, based onto identifying collision angle type which falls into one of the following sectors.

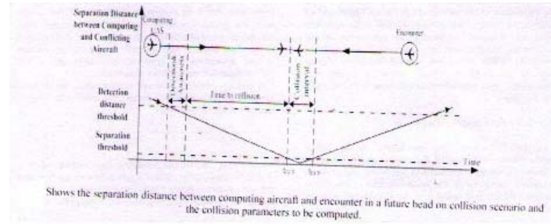


FIG. 2(a) - Sensing Distance

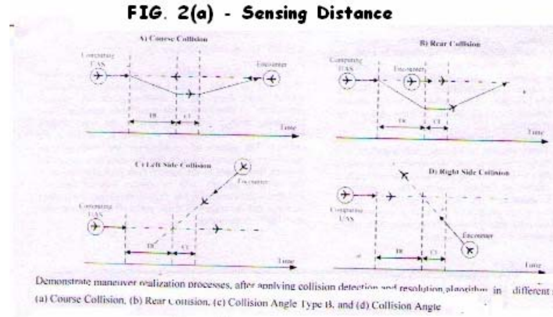


FIG. 2(b) Collision Maneuvers

a. Sector A (Course Collision)

This case occurs when both aircraft are on a heading that will take them directly to their destinations, in which computing aircraft is heading towards conflicting aircraft. A potential conflict exists regardless of the speeds of aircraft. Fig. 2(b) illustrates the conflict scenario and the trajectory flight paths of both agents after applying CAS algorithm. In order to ensure that the head-on conflict is safe, a negotiation protocol is designed such that the computing aircraft deviates at a horizontal distance equal to the distance of protective zones radius of both aircraft to right away from original flight path. The computing aircraft also command conflicting aircraft to continue flying in a straight-line without change in turn or speed.

b. Sector D (Rear Collision)

In this case, a collision angle will vary from 155° to 205°. The rear UAS will detect the conflict and accordingly it will be considered as a computing aircraft and the front UAS as a conflicting aircraft. This case is similar to that of course collision with an exception speed of aircraft is subtractive. Thus, same maneuvers will be performed for both aircrafts with same separation condition to ensure conflict resolution.

c. Sector B (Left Side Collision)

This case occurs when the relative collision angle between 25° to 155° as shown in Fig. Fig. 2(b) In this situation, computing aircraft sends command to conflicting aircraft, through negotiation, to slowdown while it performs speedup. Thus, computing aircraft will reach first at the time of first protection zone violation and therefore it will avoid the conflict. The values of speeding-up and slowing-down maneuvers are computed such that the minimum separation distance that equals to the sum of both protective zones of the colliding aircraft, will be always ensured and no violation will be reported. Fig. Fig. 2(b) demonstrates the conflict scenario and the trajectory flight paths of both agents after applying CAS algorithm.

d. Sector C (Right Side Collision)

As shown in Fig. Fig. 2(b), the relative collision angle will varies from 205° to 335°. In this situation the aircraft that detects the collision and encounter will perform reverse maneuvering commands to that of the sector b collision resolution. That is to make sure that both of them, computing and encounter aircraft, will maneuver in accordance and provide extra safety in case of both performs maneuver. The computing aircraft sends command to conflicting aircraft, through negotiation, to speedup while it performs slowdown. Thus, conflicting aircraft will reach first at the time of first collision point and therefore it will avoid the conflict, as shown in Fig. Fig. 2(b). Same separation distance is required as that of collision resolution sector b for the values of speeding-up and slowing-down maneuvers.

Collision Avoidance - Evasive Maneuvers:

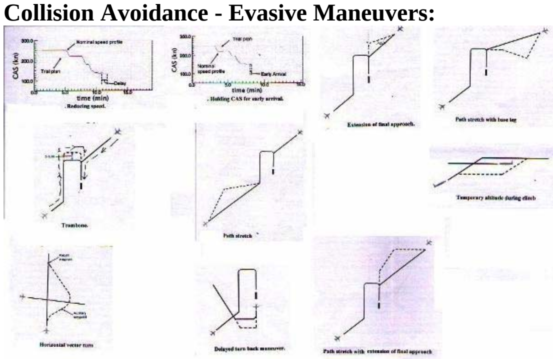


Fig. 3 – Collision Avoidance Evasive Maneuvers

A. Speed Reduction

Speed reduction is the most desirable type of maneuver for pilots and air traffic controllers, since it does not modify the aircraft's planned horizontal route to the airport. Fig. 3 shows an aircraft's calibrated airspeed (CAS) as a function of time, both for the nominal case of its descent to the final approach fix and for the case of speed reduction. The aircraft's initial speed of 250 knots is reduced to 220 knots. As a result, the aircraft crosses the final approach fix approximately 90 seconds later. In general, speed reduction is the preferred method for resolving spacing conflicts. It can absorb up to about 2 minutes of delay inside typical TRACON airspace.

B. Hold Speed

When an arrival aircraft is being scheduled to the FAF earlier than its OETA, increasing the aircraft's speed is usually not an acceptable maneuver since it is constrained by the speed limit of 250 knots CAS in the TRACON. Instead, TAAC computes a maneuver that requires the aircraft to maintain its current speed for a longer time interval compared to its nominal descent procedure. In this way, the aircraft spends more time flying at a high speed and it arrives at the FAF earlier (see Fig. Fig. 3 for an example). The extended hold of a higher speed must be terminated when the aircraft crosses a specified altitude and/or range to the final approach fix where deceleration to a reduced speed must begin. That limits the time reduction achievable by this method to less than one minute for most aircraft types.

This maneuver is defined by one parameter, namely the additional time interval of maintaining the aircraft's current speed. Maintaining a high speed for a longer time interval, however, might result in a steeper descent to the FAF.

C. Offset of Base Leg ("Trombone")

For aircraft whose trajectory includes a downwind leg, extending this downwind leg prior to turning to base leg is a frequently used Maneuver by air traffic controllers. Figure Fig. 3 depicts a case where the base leg of the aircraft was offset by 3 nautical miles in order to Resolve a spacing conflict. In TAR, such trombone resolutions are used if speed reductions alone are insufficient to solve the spacing Conflict. Trombone solutions are always used in combinations with speed reductions as much possible. In the trial planning process, the turn to base is extended in increments of half a mile which yields adequate time control accuracy.

Unlike the other horizontal maneuvers of the TAR that require the aircraft to deviate from its planned route immediately, the trombone Maneuver begins only when the aircraft is about to turn to the base leg from its downwind leg. Therefore, it has the advantage of being Available for modifications to correct imprecise projections of the aircraft's trajectory or of loss-of-separation Conflicts. Such modifications can be done even if the aircraft is only a short time (30-60 seconds) from the revised Turn-to-base waypoint.

The trombone maneuver has the capability of absorbing large amounts of delay as it can extend the downwind Leg for an additional 5 or even 10 nautical miles. Moreover, it can do so in small increments, thus proving a Powerful tool for efficiently resolving spacing conflicts.

D. Path Stretch

Two types of path-stretch maneuvers are used in TAR. Trial planner first attempts to find a resolution using a symmetric path stretch (see Fig. Fig. 3). It selects an auxiliary waypoint that lies on the perpendicular and midway to the vector that connects the aircraft's current position and the return waypoint. Symmetric path stretch generates a one-parameter family of maneuvers, defined by the turn angle from the aircraft's current heading. Symmetric path stretch maneuver is a powerful means to resolve spacing conflicts, where the primary goal is generating additional flying time for the trailing aircraft. As in trombone maneuvers, aircraft's current speed is first reduced as much possible, and it is this new speed profile that is used in generating a trial path stretch maneuver. The trial planner generates both left and right handed turns when maneuvering airspace is available in either direction. If no successful resolution is found using a symmetric path stretch maneuver, TAR will conduct a more extensive search for resolutions by employing the constant-delay elliptic path stretch algorithm. This type of path stretch is characterized by two parameters: a specified delay and the turn angle relative to current heading. Such resolutions can resolve loss-of-separation conflicts even when an aircraft has its landing STA frozen, by trying different turn angles that yield the same time-to-fly until touchdown. It is the only maneuver available that can modify a trajectory spatially without changing its landing time.

E. Horizontal Vector Turn

The horizontal vector turn (H.V.T.) was originally designed to resolve short-range conflicts such as in TSAFE where it is important to include the effects of turn rate limits in the generation of the resolution maneuver. The algorithm has also been adapted for use in the Auto resolver to resolve conflicts over longer time ranges while retaining its ability to resolve short-range conflicts. The H.V.T. algorithm determines the minimum heading change required to achieve a specified separation distance, assuming both conflict aircraft are flying at constant airspeed. The algorithm gives an explicit solution for a specified minimum separation without trial planning. It also generates a return path for the aircraft via an auxiliary waypoint and a return waypoint, as illustrated in Fig. 3. The algorithm provides the coordinates of a point on the straight-line segment that follows the vector turn where minimum separation of the resolution maneuver is reached. Then, the auxiliary waypoint is located on the straight-line segment an incremental distance beyond this point. The incremental distance, taken to be the equivalent of about 30s of flight time, ensures that a turn back maneuver starting at the auxiliary waypoint will not cause the original conflict to reappear. Although the trajectory segment to the auxiliary waypoint is designed to be conflict free, trial planning followed by a conflict check is still needed to ensure that the entire trajectory is free of secondary

F. Delayed Turn Back

Similar to the trombone maneuver for arrival aircraft, the *delayed turn back* (DTB) algorithm is designed to maneuver climbing aircraft whose departure route inside the TRACON involves a large change of heading after take-off; see Figure 8 for an example. Specifically, eligible for this maneuver are aircraft whose ultimate heading toward its destination requires the aircraft to make a large heading change (more than 90 degrees) from its initial heading following takeoff. For those aircraft, the algorithm creates an auxiliary waypoint, located on the line segment that connects their current position with the next waypoint on their original route and an incremental distance beyond this point. In this way, the aircraft delays the first turn on its planned route. A second auxiliary waypoint, located at a 90 degrees angle from the first auxiliary waypoint, is used as a turn back point where aircraft can resume their flight towards a downstream fix on their flight plans. The DTB algorithm increases the set of candidate solutions for departure aircraft. It is used as an alternative to path-stretch resolutions, when the latter either fails to find a conflict-free resolution trajectory or generates a trajectory that imposes a large delay to the departure aircraft.

G. Extension of Final Approach (“Fanning”)

In certain situations, when an aircraft is on a heading to intercept the final approach path, an alternative to path stretching is to change the heading of the aircraft so that it intercepts the final approach path further upstream (see Fig. 3). In this way, the need for specifying an auxiliary waypoint is avoided and the entire maneuver can be communicated more easily to the pilot. The pilot is simply given a new heading and the instruction to intercept the final approach. Again, the aircraft’s current speed is reduced to the lower limit of its speed envelope prior to trying a fanning maneuver.

A somewhat restricting factor for this maneuver is the interception angle with the final approach path. In order to ensure a smooth transition into the final approach, the algorithm requires that the interception angle is not greater than 60 degrees. It turns out that in most of the cases, that translates to an extension of the final approach by only a few miles which is typically less than five nautical miles. That limits the amount of delay that can be absorbed by this maneuver to relatively small values. However, it is still a useful tool for situations where speed reduction is not sufficient to absorb all delay, since it has the advantage of not requiring the specification of auxiliary waypoints.

H. Compound Horizontal Maneuvers

A combination of two horizontal maneuvers can also be employed to resolve spacing conflicts that require a large amount of delay to be absorbed. When extension of final approach does not resolve the conflict, a symmetric or elliptic path stretch can be applied in conjunction with extending the final; Fig. 3 illustrates an example. This type of resolution can absorb more delay compared to what a symmetric path stretch or extension of final can absorb separately.

There are situations where the return waypoint of a path-stretch maneuver lies on the final approach. Depending upon the position of the turn back point, it may happen that the aircraft must perform a turn of more than 90 degrees to align on final approach. Such a large heading change to capture the final approach is generally considered undesirable by pilots, and the Resolution Generator will discard such maneuvers. To address this issue, a special type of compound maneuver has been created that consists of a path-stretch and a constructed base-leg segment that provides a more acceptable transition to final approach; Fig. 3 provides an example. At the end of the path stretch maneuver the aircraft transitions to a base-leg segment, which ensures a smoother turn to final approach segment.

I. Temporary Altitude during Climb

In this case the aircraft assigned by the Resolution Generator to perform the altitude resolution maneuver is currently climbing toward its assigned cruise altitude. The time of first loss with the conflict aircraft may occur during the climb segment or after the aircraft has leveled out at its cruise altitude. The specification of the parameters for the trial resolution trajectory is the same for both situations.

A temporary altitude maneuver for an aircraft in climb is defined by two parameters. The first specifies the flight level at which the maneuvering aircraft must level out (also referred to as the temporary altitude level). Only flight levels at least 1500 ft. above the current altitude of the climbing aircraft and 1000 ft. below the flight level of the altitude where first loss occurs are eligible for temporary altitude levels. The second parameter that defines the temporary altitude maneuver is the time at which the maneuvering aircraft can resume its climb to capture the originally assigned flight level.

Conclusion:

The paper describes direct and efficient method of collision detection and avoidance for cooperative unmanned aircraft systems based on negotiation and predefined maneuvers. MDP formulation is flexible enough to accommodate a variety of sensor modalities, intruder behavior, aircraft dynamics, and cost functions. Both state estimation and policy execution are quite efficient for the state spaces considered. Current state-of-the-art solvers, using a simplified representation of the aircraft dynamics, can generate useful collision avoidance behavior. Improvements to the problem formulation may further improve performance.

The predefined maneuver will depend directly on the identification of relative collision angle between colliding aircraft. This angle is computed by comparing flight plans of computing and conflicting agents in the near-future and estimate the angle at a time when aircraft's protected zone overlapping is reported. The flight plan information sharing is considered because of the enhancement in the technology. Moreover, the utilization of flight plans in the awareness of collision detection allows perfect estimation of the collision parameters and thereby more robust conflict resolution.

The developed functional architecture allows each aircraft to negotiate with each other to determine a safe and acceptable resolution when a potential conflict is detected. Important factors, such as computational time and simplicity of implementation have been taken into consideration to resolve conflicts. The proposed approach uses simple negotiation peer to peer protocol to solve the conflicts between two aircraft. This peer to peer approach can be extended to consider multiple collisions among more than two aircraft through iterative utilization of the approach.

ACKNOWLEDGMENT

We would like to thank the Department of Computer Science, SVP College of Engineering, Visakhapantam especially those members of post graduate research committee for their input, valuable discussions and accessibility. In particular, we would like to thank Sri A. Phani Sridhar, Head of the Department and Sri Kalyan Chakravarthy, Assistant Professor and the rest of department for their expertise, patience and unrelenting support.

REFERENCES

[1] Mykel J Kochenderfer, Jessica E Holland and James P Chryssanthacopoulos, "Next Generation Airborne Collision Avoidance System," Lincoln Laboratory Journal, vol. 19 Number 1 2012.

[2] Laurent Jacolin and Robert F Stengel, "Evaluation of a Cooperative Air Traffic Management Model Using Principled Negotiation Between Intelligent Agents", American Institute of Aeronautics and Astronautics, AIAA-98-4103, 1998.

[3] Andrew. R. Lacher, David R Maroney and Dr. Andrew D Zetlin, "Unmanned Aircraft Collision Avoidance – Technology Assessment and Evaluation Methods," The MITRE Corporation, McLean, VA, USA, Case No. 07-0095, 2007.

[4] Sarit Kaus, Institute of Advanc ed Computer Scicence, University of Maryland, "Negotiation and Cooperation in Multi Agent Environments," Elseiver Science B.V., 1997

[5] Dong Xue, Jing Yao and Jun Wang, "H00 Formation Control and Obstacle Avoidance for Hybrid Multi Agent Systems,"
Journal of Applied Mathematics, Volume 2013, Special Issue (2013), Article ID 123072,

[6] Zhicheng Hon, Isabelle Fantoni and Arturo Zavala – Rio, "Modeling and Deceentralized Control for the Multiple UAVs Formation based on Lyapunov Desigg and Re-design," HAL Archives - auvertes, HAL id : hal -00932861, Jan, 2014.

AUTHORS PROFILE

Ravi Vikranth DABBIRU, is an Mechanical Engineer from Andhra University. He is a having active membership in Institute of Engineers, ISTD etc. He is pursuing M.Tech.(CST) from SVP College of Engineering, Visakhapatnam.

Kartik SIDDHABATHULA, is working as Assistant Professor at the Department of Computer Science, SVP College of Engineering, Visakhapatnam. His research interest is in the area of Sensor Network. He completed his M.S. from Univ. of Texas, Arlington, USA.