Air Traffic Control

Team 358

April 26, 2000

1 Restatement of the Problem

In an attempt to improve safety and reduce workloads on air traffic controllers, the Federal Aviation Administration is considering software that would automatically alert controllers to potential collisions between planes.

Any useful algorithm must:

- Identify all potential flight-path conflicts in time for the controller to resolve them.
- Interface with the controller in a way that improves his efficiency or reduces his workload.

To avoid confusion, we will use the following definitions for ambiguous words.

- *Plane* will always refer to a geometric object defined by the span of two vectors, e.g. the xy-plane.
- *Airplane* will refer to that wonder of modern technology which we hope to protect.
- *Eventual danger* will refer to the seriousness of the flight-path conflict that will develop if an airplane-pair is ignored indefinitely.
- *Immediate danger* will refer to the degree to which a conflict must be dealt with immediately, in order to avoid disaster.
- *Danger*, when used without a modifier, will refer to some combination or nonspecific conglomeration of both senses above.

2 Further Considerations

As it seems likely that any worthwhile algorithm must consider all pairs of airplanes within a given volume of space, we cannot hope for complexity better than $O(n^2)$, where n is the number of airplanes. Thus, since our algorithm needs to run in real time, it must be able to determine swiftly whether a given pair of airplanes present a danger to one another. At the same time, it is not sufficient to consider only the airplanes' distance from one another. For example, a pair of airplanes heading directly at each other from a distance of five miles are in danger of colliding within about twenty seconds, while two heading away from each other at a distance of a mere 1500 feet present no danger at all (although they may have done so in the immediate past).

On a busy day, it may be no help to simply present an air traffic controller with a list of airplane-pairs which will have flight-path conflicts in the near future. For example, presenting a simple list of ten problems that will arise in the next ten minutes to an already-harried controller is likely to frazzle him even further, possibly leading to reduced efficiency and a greater chance of disaster. Prioritizing the problems (presumably by some method combining the proximity and immediacy of the projected encounters) is likely to go a long way towards reducing this human tendency: the controller now needs worry only about solving the problems in turn, rather than spend invaluable time trying to prioritize them himself. Such a prioritization may well improve efficiency, as an unaided controller would likely solve such pressing problems in the order that he noticed them, rather than the "correct" order in which the computer would present them. Thus it would be nice if our algorithm could sensibly rank the immediate dangers presented by different airplane-pairs.

3 Assumptions and Hypotheses

We make the following assumptions in attacking the problem:

- All collisions between airplanes are equally and fundamentally undesirable.
- A *near mid-air collision* (sometimes called a *near miss*) is defined by the FAA as an incident in which two airplanes pass within 500 feet of one another.[1] Near mid-air collisions are undesirable but preferable to actual collisions.
- The airspace may be represented by a convex subset of $\mathbb{R}^3$, in which we represent the vertical direction by z and orthogonal horizontal directions by x and y.
- Air-traffic controllers have established protocols to prevent airplanes from colliding when crossing airspace boundaries in opposite directions.
- We can measure the position and velocity of every airplane in the airspace, and the errors in these measurements are negligible compared to the effects of turbulence.
- Every airplane has sufficiently negligible acceleration that linear models for its movement will make sense over at least the next two minutes unless:

[1] see [2]

  - It is accelerating under the direction of the controller, in which case he has already determined any conflicts which this acceleration may cause.
  - It is taking off, in which case it is not in a part of the airspace used by cruising airplanes.
  - It is attempting to land, in which case it is not in a part of the airspace used by cruising airplanes.

- All airplanes are capable of accelerating through a 2-minute turn in both clockwise and counterclockwise directions parallel to the xy-plane.[2]

- The vast majority of airplanes are traveling within one of a few vertically well-separated planes parallel to the xy-plane. (These are often called "cruising altitudes".)[3] Two airplanes which are traveling within separate such planes pose no danger to one another.

- All airplanes are capable of accelerating at a given maximum rate in the direction of their velocity. (In particular, airplanes do not cruise at their maximum speed.)

- Two airplanes are said to present an eventual danger when, if their velocities are allowed to go unchanged indefinitely, any of the following will happen in the future:

  - They collide.
  - They pass near each other at some time.
  - They pass through nearby points in space at nearby times.

  Appropriate values of "near" will be discussed briefly in sections 5,7, 8, and 11.3.

4 Possible Solutions to the Danger Problem

In attempting to determine the danger that two airplanes present one another, two considerations seem dominant: the proximity they will attain to one another, and the time remaining until they do so. Other considerations, such as obstacles presented by bad weather, probably cannot be considered except in the most simplistic of ways: if we attempt to do otherwise, we risk losing the ability to determine danger in real-time.

We present several potential solutions to this problem. Later sections will describe them in more detail.

The first is a trivial model which attempts to provide yes or no answers to the questions "Will the airplanes collide?" and "Will they have a near mid-air collision?".

[2]see [3]
[3]see [1], 27

The second is the Probabilistic model, which determines risk based upon the probabilities of collisions and near misses.

The third is the Close-Approach method. It attempts to measure danger by calculating the closest approach two airplanes will make to each other, and the time until it will occur.

The next model is the Space-Time method. It attempts to measure the danger from the closest approach the airplanes will make in space-time, and the time until it occurs.

The last model is the Logarithmic Derivative. It attempts to measure the immediate danger by placing a lower bound on the time until a collision occurs.

5 A Trivial Model

5.1 How it works

The first model assumes that effects such as wind, measurement uncertainty, or piloting imperfection, which would make an airplane's actual course deviate from the linear model given by its current position and velocity, do not exist.

In this situation, the only question that matters is whether there will be a collision, a near miss, or neither.

Large commercial aircraft often have lengths and wingspans in the neighborhood of 200 feet.[4] Thus, a collision may occur only if the centers of the airplanes pass within 200 feet of one another, and a near miss when they pass within 700 feet. If we can determine how closely the airplanes will pass each other in the future, we will have an answer to the question above.

Suppose airplanes A and B have position and velocity vectors p_A, p_B, v_A, and v_B. Set $p = p_B - p_A$ and $v = v_B - v_A$, the position and velocity of airplane B relative to airplane A. Then the distance of closest approach is simply the altitude from A to v, as in figure 1. Its length is clearly equal to $|p| \sin\theta = \frac{|p\times v|}{|v|}$.

Thus our model predicts a collision if the distance of closest approach is less than 200 feet, and a near miss if it is less than 700 feet. Based on this model, we are now able to define a measure of the eventual danger as follows: the measure takes on three discrete values a, 1, and 0 ($a \gg 1$) corresponding to a collision, a near miss, and no danger. The value of a is best determined empirically; we consider this more in the next model.

5.2 Strengths and weaknesses

This model is a simple and efficient method of predicting whether a collision or near miss will occur in the future. However, this model assumes that no uncertainty is present in the environment (so airplanes always travel at a constant speed in a straight line). This assumption simply does not hold. Airplanes will, for example, be buffeted by changing winds, and their actual trajectories will

[4] see [4]

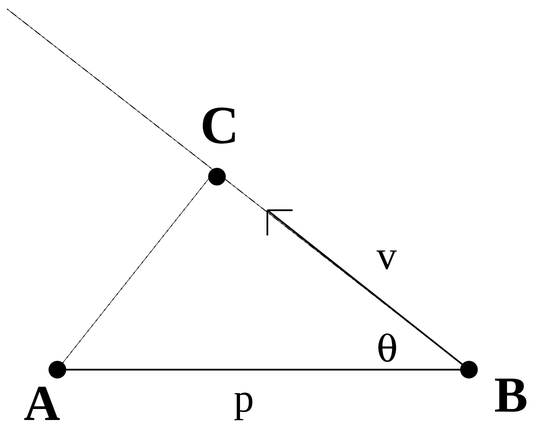


Figure 1: The position and velocity vectors of airplane B, relative to airplane A.

vary significantly and chaotically from those predicted by a linear model. Additionally, this model only considers the eventual danger resulting from a pair of airplanes, not how soon an immediate danger will be present. The model may be extended to rank collisions or near misses based on immediate danger using the following priority algorithm:

1. collisions within time period t (for example, t = 2 minutes)
2. near misses within time period t
3. collisions within time period 2t
4. near misses within time period 2t
5. collisions within time period 3t
6. etc.

Alternatively, a model may be developed which combines measures of eventual danger and temporal considerations into a single measure of immediate danger. The close-approach model discussed in section 7 is one method by which this can be achieved.

6 A Probabilistic Simulation Model

6.1 How it works

The probabilistic simulation model expands on the trivial model by taking environmental uncertainty into account. A C++ computer program was written: this program calculates the probability that a given situation will result in a collision or near miss, using a Monte Carlo simulation method. To do so, it performs a large number of random trials (each of which may result in a collision, near miss, or neither). Now let c denote the total number of collisions, let m denote the total number of near misses, and let n denote the total number of trials. Then the program calculates the probability of collisions (x) and near misses (y): x = P(collision) = c/n, and y = P(near miss) = m/n. Finally, it computes a measure of eventual danger: danger = c_1x + c_2y. As in the trivial model, we set $c_2 = 1$ and $c_1 >> 1$.

We now explain the simulation process in more detail. First, rather than assuming that each plane's speed and direction are fixed, we assume Gaussian distributions of these quantities, allowing the user to specify the mean and standard deviation of each. For each trial, a normally distributed random value of each quantity is chosen, then both planes' paths are extrapolated linearly (using the formula for minimum distance between the two planes as in the trivial model) to determine whether a collision, near miss, or neither occurs.

Several other methods for representing uncertainty also exist. For example, each trial could have been divided into a large number of discrete time steps, with random alterations to each plane's velocity being performed after each time step. Alternatively, a more analytical approach could be used to calculate the probability distribution of each plane's location as a function of time, than compute the overlap between these regions. These methods also are potentially useful, but proved too difficult to implement in the available time. The probabilistic simulation method chosen for this project has several additional advantages over these methods. First, it is relatively simple to calculate the results of each trial; other methods may be too computation-intensive to allow real-time simulation of a large number of potential conflict pairs. Second, the amount of uncertainty present in a plane's ground speed tends to be significantly higher than the amount of uncertainty in its path, perhaps because a pilot's goal is to fly in a given path at the maximum safe speed. This difference in uncertainty is evident by comparing the standards for horizontal separation and longitudinal separation based on time: according to Mahalingam[5], two airplanes should not cross the same point within 15 minutes (this corresponds to about 120 nm), yet planes in different paths are allowed to remain 3-5 nm apart. Finally, the linear extrapolation of an initial uncertainty value may be thought of as a "worst case" in terms of deviation from the mean velocity, since in models with discrete time steps, total uncertainty is less than sum of the momentary uncertainties due to the laws of averages. Thus our model errs in the direction of caution, providing a built-in safety buffer for the simulation results.

6.2 Strengths and weaknesses

The probabilistic simulation model significantly improves upon the trivial model by accounting for uncertainty in the environment. Rather than simply extrapolating the given values to predict whether a collision, near miss, or neither

[5]see [1] 26-7

occurs, a large number of trials are run with Gaussian distributions centered on the given values. This allows us to construct a danger measure that, instead of taking on only a few discrete values, gives intermediate measures based on a weighted average of these values, allowing us to gain a better idea of the relative danger present in distinctly different situations.

Like the trivial model, the probabilistic simulation model considers the eventual danger (not the immediate danger) of a given situation. Though a minimum time to collision or near miss is computed, this time is not taken into account in the final measure of danger. One possible solution would be to design an alternative danger metric taking this into account. We have chosen instead to design the simulation program with an optional user-specified maximum time (so that, if two airplanes do not reach their minimum distance by time t, the distance at time t is considered instead of the minimum distance). This allows us to ignore conflicts that occur far in the future, focusing on more immediate dangers. MICA [6] states that short term conflict analysis tools which extrapolate based on current aircraft trajectories "operate over a short time horizon, generally less than two minutes". This suggests that limiting the time horizon is a reasonable method of dealing with immediate danger under uncertainty.

One notable weakness of this simulation is its reliance on a two dimensional model of the airspace: the model assumes that the locations of the two airplanes either have approximately identical altitudes (in which case they are treated as being in the same z-plane) or substantially different altitudes (in which case the danger is assumed to be negligible). This creates a problem when the model is faced with two airplanes with moderate differences in altitude, or when one or both airplanes change altitude substantially over time, but the simulation could be extended to deal with these cases with relatively little trouble.

A C++ implementation of this is available in section B.

7 The Close-Approach Method

7.1 How it works

Intuitively, we expect the eventual danger to be inversely related to the closest approach the airplanes will achieve with each other, and the immediate danger to be inversely related to the time until that closest approach. Thus, we have, as a first approximation,

$$\text{Eventual danger} \approx \frac{1}{(\text{distance of closest approach})^{\alpha}}$$

$$\text{Immediate danger} \approx \frac{1}{(\text{distance of closest approach})^{\alpha}\,(\text{time until closest approach})^{\beta}}$$

Since danger would be averted by accelerating the airplanes away from each other, the extra separation achieved should be proportional to the square of the

[6] see [7],p.124

time during which they are accelerated. Since the time in which the airplanes can accelerate is bounded by the time until their projected close approach, it seems reasonable to set $\beta = 2\alpha$. As raising to any positive power won't affect ordering, we can set $\beta = 2$ and $\alpha = 1$. Such a simple formula, while useful in the general case, runs into trouble in boundary situations. No matter how far away the airplanes will be at their closest approach, the computation above shows the immediate danger going to infinity as they approach that closest approach. Also, if the two airplanes are on a collision course, this computation gives infinite immediate danger, no matter how much time remains until their collision. Finally, if the airplanes have nearly identical velocities, this rates the immediate danger as very close to zero (unless the aircraft are practically on top of one another), when it should intuitively be inversely proportional to their current separation. We fix the formula as follows:

$$\begin{aligned}\text{Immediate danger} &= \frac{1}{(\text{distance of closest approach}+c_1)(\text{time until closest approach}+c_2)^2} \\ &+ \frac{c_3}{\text{current separation}}\end{aligned}$$

where c_1, c_2 and c_3 are positive constants, probably best determined empirically.

Now we must compute the factors above. Suppose we are given positions p_A and p_B and velocities v_A and v_B for airplanes A and B. Then $p = p_B - p_A$ and $v = v_B - v_A$ are the position and velocity vectors of airplane B in a reference frame where airplane A is unmoving at the origin, as in figure 1. The vector corresponding to the closest approach of the two airplanes is the altitude from A to v, which has length equal to $p \sin\theta = \frac{|p\times v|}{|v|}$. If this distance is sufficiently large, say greater than five nautical miles[7], then we may conclude that this pair of airplanes pose no meaningful eventual danger to one another, and move on to the next pair. If the airplanes will pass close to one another,then we now have a rating for eventual danger: $\frac{|v|}{|p\times v|}$.

The time that will elapse before the closest approach is attained is simply the time until airplane B reaches point C, or $\frac{|\overrightarrow{BC}|}{|v|}$. The numerator here is equal to $|p|\cos\theta = \frac{p\cdot v}{|v|}$, so the time is equal to $\frac{p\cdot v}{|v|^2}$. If this time is negative, then these two airplanes have already survived their closest approach, and present no danger to one another in the future. If the time is positive, we can calculate the immediate danger by plugging everything in and simplifying:

$$\text{Immediate danger} = \frac{|v|^5}{(|p\times v| + |v|\,c_1)\left(p\cdot v + |v|^2\,c_2\right)^2} + \frac{c_3}{|p|}.$$

This yields a $\frac{0}{0}$ in the first summand when $v = 0$, i.e., when the two airplanes are flying exactly parallel to one another. In this case, we set the immediate danger equal to $\frac{c_3}{|p|}$.

[7] see [1], 26-7

7.2 Strengths and Weaknesses

The main strength of this model is that it is extremely swift: the danger presented by a pair of airplanes can be computed with just over fifty basic operations on known real numbers if eventual danger exists, and about half that to conclude that it does not; thus even a modern personal computer should be easily able to handle this computation for several thousand airplane-pairs every second, or about 500 airplanes every two minutes.

The most serious weakness of this model would appear to be that it does not attempt to worry about airplanes that pass near each other in time but not in space. For example, it cannot distinguish between two airplanes flying the exact same route through space with a time-separation of fifteen seconds and two airplanes flying parallel to one another with a physical separation of two miles in some direction orthogonal to their velocity; the first situation appears to be much more dangerous. The next model will attempt to differentiate between these and similar situations.

8 The Space-Time Method

8.1 How it works

The Space-Time method uses similar reasoning to the Close-Approach method, but considers the airplanes' proximity in space-time, rather than simply in physical space. Thus, intuitively, we have:

$$\text{Eventual danger} \approx \frac{1}{\text{closest approach in space-time}}$$

$$\text{Immediate danger} \approx \frac{1}{(\text{closest approach in space-time})\,(\text{time until closest approach})}$$

The same corrections for boundary conditions will apply, leaving

$$\begin{aligned}\text{Immediate danger} &= \frac{1}{(\text{closest approach in space-time}+\gamma_1)(\text{time until closest approach}+\gamma_2)^2}\\ &+ \frac{\gamma_3}{\text{current space-time separation}}\end{aligned}$$

These quantities are harder to compute than in the Close-Approach method. Suppose airplane a is positioned at the origin, and p gives the position of airplane b with respect to airplane a, and that they have velocity vectors v_a and v_b, respectively. Then we may represent the future of airplanes a and b by rays in $\mathbb{R}^4$, parametrized by $\left(v_{a_x}t_a, v_{a_y}t_a, v_{a_z}t_a, kt_a\right)$ and $\left(v_{b_x}t_b, v_{b_y}t_b, v_{b_z}t_b, kt_b\right), t_a, t_b > 0$, where k is a constant chosen so that a distance of one unit in the time coordinate is equally dangerous to a distance one unit in one of the horizontal coordinates. Mahalingam[8] equates a fifteen-minute separation in time with a five-nautical-mile separation in space; we will assume that this scales. Then k equals 5 nautical miles per 15 minutes, or about 34 feet per second.

[8] see [1],26-7

For any t_a, t_b, the space-time distance between the position of airplane a at time t_a and the position of airplane b at time t_b is given by
$\delta(t_a, t_b) = \left|\left(v_{a_x} t_a, v_{a_y} t_a, v_{a_z} ta, kt_a\right) - \left(v_{b_x} t_b, v_{b_y} t_b, v_{b_z} tb, kt_b\right)\right|$. This yields
$\left(\delta(t_a, t_b)\right)^2 = At_a^2 + Bt_a t_b + Ct_b^2 + Dt_a + Et_b + |p|^2$, where

$$\begin{aligned} A &= k^2 + |v_a|^2 \\ B &= -2k^2 - 2v_a \cdot v_b \\ C &= k^2 + |v_b|^2 \\ D &= -2p \cdot v_a \\ E &= 2p \cdot v_b \end{aligned}$$

Now the minimum space-time distance occurs at t_a and t_b which minimize this expression, that is, where $\nabla f^2 = 0$. This occurs at $t_\alpha = \frac{2CD-BE}{B^2-4AC}, t_\beta = \frac{2AE-BD}{B^2-4AC}$. We observe that this is well-defined whenever the velocities are not equal, as $B^2-4AC = 4\left(\left(v_a \cdot v_b\right)^2 - |v_a|^2 |v_b|^2 + 2k^2 v_a \cdot v_b - k^2 |v_a|^2 - k^2 |v_b|^2\right)$. The first two terms add to less then zero by Cauchy-Schwarz, and the last three by AM-GM, with equality in both cases only when the velocities are equal. (The case when the velocities are equal is handled more simply: For every t_α, there is a unique t_β satisfying $Bt_\alpha + 2Ct_\beta + E = 0$ (since C is always positive) which yields the minimum space-time separation.) The minimum space-time separation is given by $f(t_\alpha, t_\beta)$. The time until this separation is given by $\min\{t_\alpha, t_\beta\}$.

The current space-time separation would appear to be the minimum of the space-time separations between the current position of airplane a and the future of airplane b, and the current position of b and the future of a. The first of these occurs when $t_b = t_\beth = -\frac{E}{2C}$, the second where $t_a = t_\aleph = -\frac{D}{2A}$. (Note that, in the case where the velocities are equal, $t_\beth$ is the t_β associated to $t_\alpha = 0$.) If one of these times is negative, we replace it with 0, which is not in the past. Thus we have:

$$\text{Current space-time separation} = \min_{t_\aleph, t_\beth \geq 0} \{\delta(t_\aleph, 0), \delta(0, t_\beth), \delta(0,0)\}$$

Actual determination of danger is done as in the close-approach model, except that the times associated to closest approach must be computed first. If either is negative, then any danger posed by this airplane-pair has already been avoided, and so the eventual and immediate dangers are set equal to zero. Then the space-time separation of the closest approach is computed; if it is too large, say greater than five nautical miles, we conclude that these airplanes pose no danger to one another. If, on the other hand, they do pose significant eventual danger, the eventual and immediate dangers are computed by plugging everything in.

8.2 Strengths and Weaknesses

It is clear that this is a more timid measure of danger than the close-approach method: Every airplane-pair receives at least as high immediate- and eventual-danger measures from the space-time method as from the close-approach method, while it recognizes as dangerous some cases which the close-approach method does not. This suggests that it is probably more useful as a predictor of danger, so long as the immediate-danger rankings continue to make sense. If one accepts the premise that danger arises when two airplanes pass through nearby points in space at nearby points in time, as seems eminently reasonable, then these measures in fact make more sense than those generated by the close-approach model. It is somewhat but not significantly slower to implement, requiring only about 100 basic operations to compute the eventual and immediate dangers: Thus it should be able to handle about half as many airplanes as the close-approach model in the same amount of time; this is easily fast enough to work in real-time.

On the other hand, this model is much more opaque to any human who must try to work with it. An air-traffic controller attempting - in real time - to plot a course correction which minimizes the danger measures given by this model could easily find himself frustrated and confused very quickly. Human beings, no matter how great our intellectual understanding of such things, are often simply not equipped to think in terms of extra dimensions.

9 The Logarithmic Derivative

The logarithmic derivative model is different from those which have gone before in that it originates from different observations and intuitions. It is probably the model that most closely approximates what an air traffic controller would do, in the absence of predictive software.

9.1 How it Works

The model arises from the observation that, if the velocities of airplanes A and B remain constant, the function $\frac{d}{dt}$ (distance between airplanes A and B) is monotonically increasing with time (unless the airplanes are traveling in the same line, in which case it is constant) and is bounded both above and below. Thus, $-\frac{d}{dt}$ (distance between airplanes A and B) is monotonically decreasing, and $-\frac{\text{distance}-\ell}{\frac{d}{dt}(\text{distance})}$, evaluated at t_0, gives a lower bound on the time between t_0 and any future time t at which the airplanes are separated by a distance less than ℓ. Thus the reciprocal of this quantity, $-\frac{\frac{d}{dt}(\text{distance})}{\text{distance}-\ell}$[9] might work as a measure of immediate danger. Let us investigate the behavior of this function.

Suppose that airplane B has position and velocity vectors p and v, respectively, in some frame of reference where A is stationary at the origin, as in figure 2. Now

$$\frac{d}{dt}\left(|p|^2\right) = \lim_{h\to 0}\frac{|p+hv|^2 - |p|^2}{h} = \lim_{h\to 0}\frac{2hp\cdot v + h^2|v|^2}{h} = 2p\cdot v$$

[9] This suggests the name "logarithmic derivative", as the logarithmic derivative of a function f is defined to be the derivative of its logarithm: $\frac{1}{f}\frac{df}{dt}$.

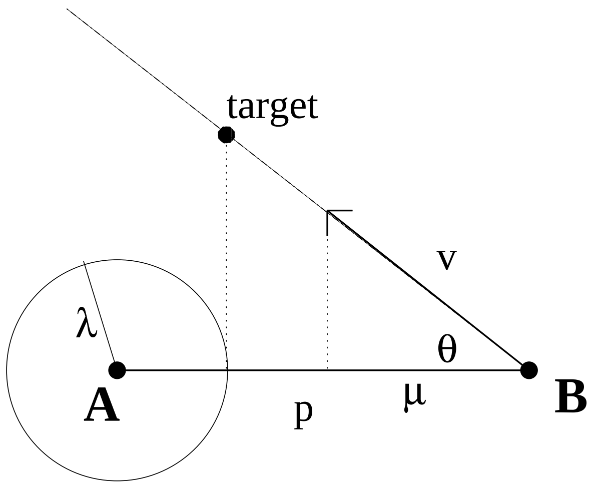


Figure 2: The logarithmic derivative

But $\frac{d}{dt}\left(|p|^2\right) = 2\,|p|\,\frac{d}{dt}\,|p|$, so $\frac{d}{dt}\,|p| = \frac{p \cdot v}{|p|} = |v|\cos\theta$. This is represented in figure 2 by μ, the projection of one time-unit of velocity onto p. Dividing by $|p| - \ell$ gives the number of time-units necessary before the projection onto p intersects the circle of radius ℓ about A. If ℓ is chosen to represent some danger threshold, this makes intuitive sense. Consider also the behavior of this function as time passes: If B will pass through the circle of radius ℓ, the target point converges to the intersection of B's trajectory with the circle, and the measure goes to infinity as B approaches the circle, then becomes negative once it passes inside. If ℓ is sufficiently small (like, say, 700 feet, the threshold for a near miss), this is fine: Once the two airplanes are within ℓ of each other, it is in some sense already too late. If, on the other hand, B will not pass within ℓ of A, the target point goes to infinity as the B approaches the point of least separation from A, and the measure simultaneously goes to zero, then becomes negative as that point is crossed. This is also fine; all danger is past once the point of least separation is reached.

9.2 Strengths and Weaknesses

The greatest strength of this model is its simplicity; the immediate danger measure can be computed with only about ten basic operations; this is even faster than the close-approach model. Furthermore, it offers other useful information: the reciprocal of the measure is exactly how much time the controller has to act, before this pair of airplanes will have played out whatever danger they face from one another.

Another strength is that this measure behaves correctly in two of the situations where the previous two measures required ugly fixes. If the two aircraft

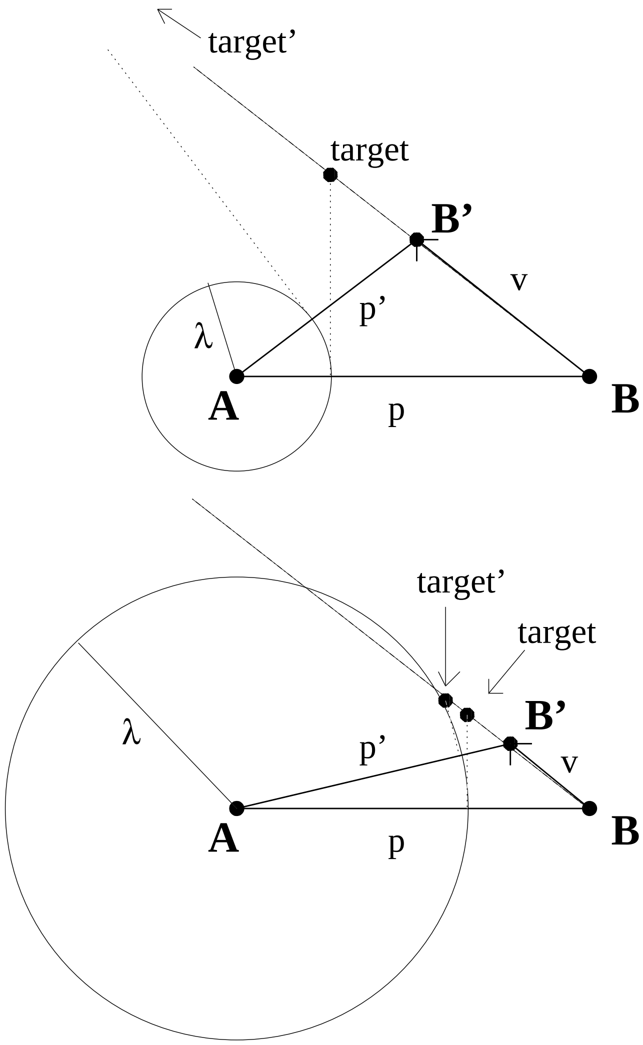


Figure 3: Motion of the target point over time; two cases

are on or very near a collision course, it acts as a countdown to that collision, which is correct behavior. If the airplanes are near their point of closest approach, it gets large only if they are actually close to one another, and remains appropriately small otherwise.

One flaw is that this measure always gives an immediate danger near zero when the airplanes have nearly identical velocities. As before, we'd like the danger in this case to be roughly inversely proportional to their separation; we can solve this problem by adding a term of $\frac{c}{|p|}$ to the measure; unfortunately, doing so eliminates the nice relationship between this measure and the time left for the controller to act.

A potentially more serious problem is the inability of this algorithm to project far into the future. It will detect almost no difference between, for example, two pairs of airplanes with the same relative velocities, the first of which are on a course to collide in five minutes' time, and the second to pass each other with a mile of separation.

10 Testing the models

10.1 Some sample situations

We postulate several airplane-pairs in a variety of situations which are somewhat exceptional in terms of the quality of peril posed. Since our various systems for determining immediate danger all purport to be measuring the same thing, we hope that they will agree on which situations pose the greatest immediate danger. It would also be nice if they agreed with our intuitions on the subject.

In the following situations, all airplanes move with speed equal to 480 knots, or about 810 feet per second. Airplane a's initial position is always at the origin. Airplane b's position, in feet, and the angles associated to the velocities will be given below.

1. **Impending head-on collision:** Airplane a heading 0°. Airplane b heading 180° from (6000,0).
2. **Impending oblique collision:** Airplane a heading 60°. Airplane b heading 120° from (3000,0).
3. **Tailgating:** Airplane a heading 0°. Airplane b heading 0° from (2400,0).
4. **Flying alongside:** Airplane a heading 0°. Airplane b heading 0° from (0,2400).
5. **Same point, nearby time:** Airplane a heading 0°. Airplane b heading 90° from (2400,-3200).
6. **Same point, nearby time:** Airplane a heading 0°. Airplane b heading 120° from (4400,-2100).

7. **Passing at a distance:** Airplane a heading 0°. Airplane b heading 180° from (18000,-6000).
8. **Far-future head-on collision:** Airplane a heading 0°. Airplane b heading 180° from (600000,0).
9. **Flying parallel:** Airplane a heading 0°. Airplane b heading 0° from (0,18000).
10. **Right angles:** Airplane a heading 0°. Airplane b heading 270° from (18000,0).
11. **Receding:** Airplane a heading 0°. Airplane b heading 180° from (0,6000).
12. **Receding:** Airplane a heading 120°. Airplane b heading 60° from (3000,0).
13. **Receding:** Airplane a heading 180°. Airplane b heading 0° from (6000,0).

10.2 Ranking the Samples

We can put an intuitive partial ordering on the immediate danger presented by these situations. Obviously the two impending collisions are worst, with the head-on collision being even worse than the oblique collision, as maneuvering in the former case will avail the airplanes less. The next most dangerous situation is the one titled Tailgating; if something causes airplane b to slow down significantly, there is very little that airplane a can do to avoid a conflict. The fourth situation is similar to the third, except with more maneuverability. The next two situations feature a single point through which both airplanes pass within one second of each other; if both speeds and directions are altered in the wrong way, there could be serious trouble. Situation 8 will transform into situation 1 if it is ignored for too long. Trouble could happen in situations 7 and 9 only if both airplanes are allowed to veer significantly in the wrong directions. In situation 10, airplane b would nearly have to turn around to make trouble, and in the last three situations, both airplanes would have to turn around for any danger to arise (i.e., there is no danger at all.)

We also evaluated the immediate danger presented by each situation, using all of the algorithms presented in the preceding sections, except for the space-time algorithm, for which it was infeasible to find appropriate constants in the allotted time. The rankings they produced are reported in table 1. The probabilistic method here uses the metric

$$\text{danger} = \frac{\text{collisions}}{10000 \text{ trials}} + \frac{1}{20}\frac{\text{near misses}}{10000 \text{ trials}}.$$

The close-approach method here uses the constants $c_1 = 50$ feet, $c_2 = 5$ seconds, $c_3 = .05\ \text{Hz}^2$, (and displays the immediate danger value in units of $\frac{10000\text{Hz}^2}{\text{ft}}$.) These constants seem to give reasonable results.

We note immediately that both the probabilistic and the close-approach models match up very well with our intuitive rankings, though not particularly

Situation	Intuition	Triv	Prob	Close-App	Logarithmic
1	1	2 (a)	1 (9946)	2 (7.77)	3.5 (.27)
2	2	2 (a)	2 (7370)	1 (8.60)	3.5 (.27)
3	3	9.5 (0)	4 (860)	3.5 (2.08)	10.5 (0)
4	4	9.5 (0)	7 (98)	3.5 (2.08)	10.5 (0)
5	5.5	4.5 (1)	5 (479)	5 (1.31)	2 (.28)
6	5.5	4.5 (1)	3 (1017)	6 (1.12)	1 (.29)
7	8.5	9.5 (0)	11 (0)	9 (0.26)	5 (.08)
8	7	2 (a)	6 (263)	10 (.001)	7 (.003)
9	8.5	9.5 (0)	8 (13)	8 (0.277)	10.5 (0)
10	10	9.5 (0)	11 (0)	7 (0.280)	6 (.05)
11	12	9.5 (0)	11 (0)	12 (0)	10.5 (0)
12	12	9.5 (0)	11 (0)	12 (0)	10.5 (0)
13	12	9.5 (0)	11 (0)	12 (0)	10.5 (0)

Table 1: The dangers in the situations presented in section 10.1, as ranked by the various algorithms.

so with one another. The trivial method and the logarithmic derivative method both compare much less favorably. This is no great surprise, as we have already noted that the trivial method is a rather poor measure and that the logarithmic derivative does poorly when both airplanes have nearly the same velocities; in three of the thirteen situations under examination, both airplanes have exactly the same velocity. When those situations are removed, the logarithmic derivative corresponds quite well to the intuitive ranking.

We note as well that the close-approach model agrees with our intuition almost exactly, except for switching the rankings of situations 8 and 10. As we expect very little eventual danger from situation 10, and very little immediate danger from situation 8, this is perhaps not so big a deal. In these situations, the close-approach model does very well by our intuition. We note in passing that the space-time model would probably do even better: It would certainly rank situation 3 as more dangerous than situation 4 (agreeing with our intuition), and, as it is very closely related to the close-approach model, might well rank all the other situations identically.

11 Recommendations

11.1 Which danger model should we use?

We have presented, in the preceding sections, five separate models that can be used for predicting the danger posed by a given airplane-pair, and given brief discussions of their strengths and weaknesses. In order to obtain a quantitative measure of the accuracy of each model, we created thirteen sample cases and compared each model's ranking of the relative danger of these cases to our own intuitive ranking of the danger of each situation. The RMS error between the model rankings and intuitive rankings was computed for each model. The Trivial and Logarithmic Derivative models both had RMS error values above 3.0, suggesting that these models have significant differences from our intuitive notion of danger.

We do not recommend the trivial model, due to the manifest falsehood of its most basic assumption (an environment with no uncertainty) and its failure to perform satisfactorily on the sample set.

We also do not recommend the logarithmic derivative model, but our reasons are more complex. The logarithmic derivative model in its present form does not perform well on the sample set. Although we have briefly discussed ways that the logarithmic derivative model could be improved, it can be argued that the model would still be less desirable than others. The logarithmic derivative model appears to behave in much the same manner as would a human operator watching only a graphical display; thus any improvement would still result in a model that essentially behaved like a very fast human operator. We expect any model to have some "blind spots", kinds of danger which it will fail to detect for longer than many other models. In general, an alert human operator could complement the model by detecting some dangers in these blind spots. The blind spots of the logarithmic derivative model, unfortunately, will be shared by the human operator, and so he cannot complement it in this manner. However, we do recommend study of the logarithmic derivative model, particularly in relation to other models. Understanding its flaws should help lead to an understanding of human operators' weaknesses.

We do recommend any of the other three models, however. They are all based on sensible assumptions, and both the probabilistic and the close-approach models matched well with our intuition when we tested them on the sample set, yielding RMS error values below 1.5. We have no reason not to believe that the space-time method would have performed at least as well, had it been feasible to find appropriate constants in the given time.

In particular, we most highly recommend the close-approach method. Its rankings of immediate danger match up particularly well with our intuition, yielding an RMS error value of only 1.28. In addition, this method computes danger measures at blindingly fast speeds, allowing it to constantly update the danger measures for every airplane-pair in real-time. Finally, its mechanisms are intuitive and clear, so that controllers may intuitively take it into account when plotting courses.

11.2 When should the tower intervene?

We have presented several methods for measuring danger; in general, when the danger is too high, the tower must intervene. Threshold values for danger which should force the tower to intervene will obviously vary based upon which measure is used. Some of these models offer ways to measure both eventual danger and immediate danger, others only one or the other. Whenever both measures exist, the eventual danger is the one that should be used: If the eventual danger is higher than some threshold value, then the tower will - eventually - have to intervene. This does not necessarily mean that the tower must intervene immediately, however. If the immediate danger is low enough, the tower may (and should) take the time to deal with more pressing problems first. When a measure exists for only one of the two kinds of danger, then obviously that measure must be used.

For each of our models, we must determine the relevant threshold values.

In the trivial model, the only values are 0,1,and a. Obviously the tower cannot be expected to intervene for every airplane-pair with danger measure 0. On the other hand, it must intervene to prevent a near-miss, thanks to the second assumption in section 3. Thus the threshold danger value is 1.

In the probabilistic model, we only have a measure for eventual danger. Thanks to the simulations in section 10.1, we can guess that most situations with no intuitive danger give a danger rating less than $\frac{1}{10000}$, and all situations with high intuitive danger give a danger rating well greater than $\frac{1}{10000}$. Thus $\frac{1}{10000}$ seems like a good "safe" threshold value; it will catch all high-intuitive-danger cases, as well as cases such as situation 9 in section 10.1.

In the close-approach and space-time models, there is a natural map from the eventual danger to distance (given by taking the reciprocal). If the corresponding distance is less than some reasonable length, then the tower should intervene. Mahalingam[10] argues that airplanes should be horizontally separated by 3 nautical miles. Taking him at his word, we can set the threshold danger value for both of these models equal to $\frac{1}{3\text{ nm}}$. This corresponds to intervening if the airplanes will eventually come within 3 nautical miles of one another. (Note that if they come together in the past, then the danger measure will be negative and we won't have to intervene.)

In the logarithmic derivative model, we have only a measure of immediate danger. There is a natural map from the immediate danger to time (also given by taking the reciprocal). Thus if the corresponding time is less then some reasonable span, we should intervene. MICA[11] argues that 2 minutes is a natural time interval, so we set the threshold equal to $\frac{1}{2\text{ min}}$. This corresponds to intervening only if the airplanes will come within 700 feet of their closest approach within the next two minutes.

11.3 How close is too close?

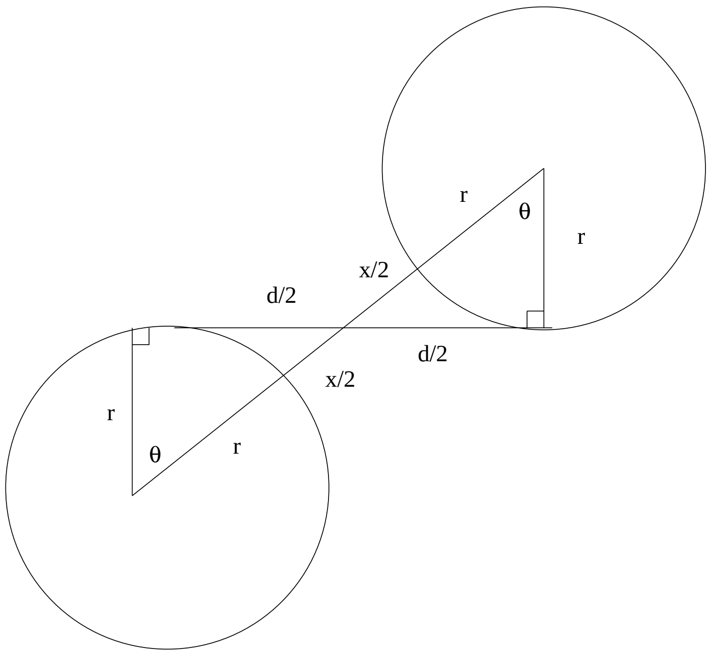
At some point, we must face the question, "How close can two airplanes get before intervention from the tower cannot save them from at least a near-miss?". The answer to this question is clearly related to the paths on which they approach one another. We will put an approximate bound on the answer by requiring that the airplanes' original trajectories predict a head-on collision. While there may be some trajectories for which the answer is larger, it can never be much larger.

[10] [1],26-7
[11] [7],125

Assume that each airplane has velocity v feet per second, and that each can turn z radians per second. We find the distance d (in feet) at which each airplane must start turning, to ensure that the aircraft do not pass within each x feet of each other at any time.

To do so, we first note that each turn (assuming constant turning rate) forms an arc of a circle. Let r denote the circle's radius. Since the length of an arc subtended by an angle θ is given by s=rθ, we take the derivative to obtain r = $\frac{ds/dt}{d\theta/dt}$ = v/z.

Next we note that x (the shortest distance between the two circles) is equal to the sum of the distance k between the centers of the two circles and the two radii: k = x + 2v/z.



Finally, we observe that the initial line of flight of the two planes is tangent to both circles, and hence two right triangles are formed. For each triangle, the lengths of the two legs are r = v/z and d/2, and the hypotenuse is k/2 = v/z + x/2. Thus we apply the Pythagorean theorem to obtain $(d/2)^2 = (v/z + x/2)^2$ - $(v/z)^2$. This gives us d = $\sqrt{x^2 + \frac{4xv}{z}}$.

Now we consider two airplanes with velocity 480 knots = 810.67 feet per second, and turning rate 3 degrees = $\pi/60$ radians per second (a standard "two-minute turn"). In order to ensure that the aircraft do not pass within 500 feet of each other (the standard FAA definition of a "near miss"), we assume a wingspan of 200 feet, and thus the centers of the airplanes must be 500 + 200 = 700 feet apart to ensure that a near miss is avoided. Given these values of x, v, and z, we calculate d = 6621 feet = 1.09 nm. Thus in order to avoid a near miss, both pilots must start a turn at a distance of 1.09 nm from each other. This implies that the controller must identify the problem and communicate his command to the pilots significantly before this point. Assuming a maximum delay of 15 seconds between when the controller's discovery of the problem and the pilot's response to it, this gives us a "safety distance" of approximately five nautical miles, or 19 seconds, until a head-on collision.

12 Measuring Complexity

In order to measure the complexity of the workload faced by an air traffic controller (ATC), we need a basic understanding of the tasks performed by the ATC and how his ability to perform these tasks is affected by the number of airplanes in the airspace sector. To that end, we present the following algorithm as a model for the decision process undertaken by an ATC in detecting and solving conflicts:

ATC Decision Algorithm

1. Scan the radar screen (and other sources of information) for airplanes that are located close to each other or currently at a safe distance but whose projected paths will cross.

2. If a pair/group of airplanes at a given time instant appear close to each other, evaluate velocity and heading information to determine whether the airplanes will move below a minimum separation distance of each other within the near future (2 minutes?).

3. If a potential conflict is detected, scan briefly to see if there are any more pressing conflicts that need to be taken care of first.

4. If there appear to be no other conflicts that need to be taken care of first, then alert the pilots of the airplanes detected in step 2 of their situation and formulate alternate routes for them to take.

5. Assess whether the alternate routes designated in step 4 will cause conflicts with the projected routes of nearby aircraft.

6. If the alternative routes will cause conflicts, reformulate alternative routes for the airplanes in conflict.

7. If there are no impending conflicts, or if the most recent conflicts have been resolved successfully, then take care of other tasks such as landings, takeoffs, assessment of weather conditions, and clearance of airplanes to enter/exit airspace sector.

8. When the secondary items in step 7 have been adequately dealt with, return to step 1.

The order of the steps in the preceeding algorithm is drawn from a synthesis of reports on the factors identified by experienced air traffic controllers as relevant to conflict prevention. [12]

As we stated in our assumptions, the actual role of the factors mentioned in step 7 must be subordinated in order to simplify the task of determining complexity as a function of air traffic and the potential for conflicts to develop.

With this model for the decision process undertaken by the ATC, we can examine how the number of planes in the airspace, the number of potential conflicts, the number of planes entering and exiting the airspace at any given time, and the volume of the air space affect the complexity of the air traffic workload.

12.1 Complexity of Step 1

Step 1 in the ATC Decision Algorithm involves scanning the airspace for potential conflicts. The ATC, or a conflict detection algorithm, might do this by examining every pair of airplanes in the airspace to see if their distance apart is less than some prespecified safe value. If there are currently n airplanes in the airspace, then there will be $\binom{n}{2} = \frac{n(n-1)}{2}$ airplane pairs for the ATC to consider. Additionally, the task of identifying airplanes whose paths will probably cross at some later time requires making tentative projections for every pair of aircrafts, so this operation also has complexity of order $O(n^2)$.

Naturally, measuring complexity strictly by the number of pairs of airplanes the ATC will have to evaluate for proximity violations and projected path intersections does not completely capture the process by which the ATC gathers information or the uncertainty of extrapolating current paths a significant amount of time into the future. Specifically, it might be more realistic to divide airplanes in the airspace at a given time into clusters and analyze the complexity associated with each cluster. The process of dividing the airspace into clusters can be accomplished automatically by one of several algorithms, which are particular forms of the following general template:

Clustering Algorithm

1. Number the airplanes from 1 to n.
2. Define $close(i, j)$ to be true iff the distance between airplanes i and j is less than a pre-specified distance threshold.
3. For each airplane a, perform one of the following operations:
    (a) If no clusters have been formed yet, or if $close(a, j)$ is false for all airplanes j that have been placed in clusters, then create a new cluster with the airplane as sole member.

[12] see [9]

(b) Otherwise, while $close(a, j)$ and $close(a, k)$ hold for airplanes j and k in different clusters, combine the clusters containing airplanes j and k.

(c) Then, once clusters have been combined such that $close(a, j)$ holds for only airplanes j in a single cluster, add airplane a to the cluster.

If the aircraft in a given airspace can be divided into C distinct and independent clusters, then the total number E of proximity evaluations the ATC would have to perform is given by

$$E = \sum_{i=1}^{C} \frac{n_i(n_i - 1)}{2},$$

where n_i is the number of airplanes in cluster i. The motivation for the cluster approach is that, when scanning a radar screen, an ATC often mentally groups certain sets of aircrafts together in order to reduce the amount of mental work he/she has to do to assess the number of potential proximity violations. It is reasonable, therefore, to consider the cluster approach as a more accurate representation of the number of comparisons the ATC must make when performing an initial scan of the airspace sector. Note that, although the cluster approach is $O(n^2)$, the values taken on by E for an airspace with n airplanes will be consistently lower than the values of $\binom{n}{2}$. This follows from the fact that

$$\begin{aligned}
\binom{n}{2} - E &= \frac{n(n-1)}{2} - \sum_{i=1}^{C} \frac{n_i(n_i - 1)}{2} \\
&= \frac{n^2}{2} - \frac{n}{2} - \left(\sum_{i=1}^{C} \frac{n_i^2}{2} - \sum_{i=1}^{C} \frac{n_i}{2} \right) \\
&= \frac{n^2}{2} - \sum_{i=1}^{C} \frac{n_i^2}{2} \\
&= \frac{(\sum_{i=1}^{C} n_i)^2 - \sum_{i=1}^{C} n_i^2}{2} \\
&> 0.
\end{aligned}$$

For an airspace with C groups of equal size, for instance, the ratio of the two measures is

$$\frac{E}{\binom{n}{2}} = \frac{n - C}{C(n-1)} < \frac{1}{C}$$

for $C > 1$. This illustrates the difference in the magnitudes of these two measures for step 1 complexity.

12.2 Impact of Group Interaction on Step 1 Complexity

Although the measure E is a reasonable metric for step 1 complexity, it can be improved further by taking into account additional factors that affect the number of step 1-level comparisions performed by the ATC in the ATC Decision Algorithm. One such factor is the effect of interactions between separate clusters. Contrary to the simplifying assumption made above, clusters of airplanes are not completely independent of each other, and decisions made to avert problems in one cluster by rerouting flight paths might potentially affect the airplanes in another cluster. One way of accounting for such interactions would be to add a term to E that represents the number of pairs of groups:

$$E_{\text{revised}} = \sum_{i=1}^{C} \frac{n_i(n_i-1)}{2} + \frac{C(C-1)}{2}.$$

This additional term encodes the fact that, for each pair of groups, the ATC must consider the likelihood that rerouting traffic from one will affect the other.

12.3 Measuring Step 2 Danger

Up to this point, we have focused only on the component of complexity that arises from step 1 of the ATC Decision Algorithm, where the ATC identifies airplanes that appear too close together at the current time or whose projected courses might lead them to an eventual conflict if not corrected. The primary part of step 1 is the former task of identifying aircraft whose current separation distance is less than some approximate safe value, since presumabably any such pair has the potential to become a dangerous situation. Given any pair of aircraft identified in step 1 as being too close to each other, step 2 of the ATC Decision Algorithm takes heading and velocity information into account to generate some estimate of the potential for a collision, near miss, or lesser airspace violation caused by that airplane pair.

In the case of step 1, we argued that complexity depends on the number of pairs the ATC had to evaluate for proximity violations and eventual path conflicts. For step 2, the complexity of the task facing the ATC depends in part on the level of uncertainty involved in predicting the location of the members of an airplane pair or cluster in the near future. Also, a measure of step 2 complexity should take into account the level of danger implicit in the various situations identified in step 1. The more situations there are that represent a high level of danger, the more complex the task of the ATC becomes because he is required to make more crucial decisions per unit time. The increased demand on mental resources, in turn, requires the ATC to pay less attention to secondary situations that could develop into problems if left unattended. This problem is compounded by the fact that severe problems, such as potential collisions or near misses, demand almost total attention until they are resolved, and are not resolved completely until (in parts 4-6) the ATC determines that the alternate paths he has directed the pilot to take do not cause any other conflicts.

In order to construct a good metric for the complexity of step 2, we need a good metric for the danger represented by a given aircraft pair, given the position and velocity of both airplanes. The first portion of our paper is devoted to creating several workable alternatives for such a metric, and we use those results here.

A natural way to incorporate the danger presented by each aircraft pair into a danger metric for the airspace as a whole would be to compute the measure

$$D = \sum_{1 \le k < l \le n} f_{kl}, \tag{1}$$

where the airplanes in the airspace are numbered from 1 to n and $f(k,l)$ represents the value of one of the danger metrics established earlier (trivial, close approach, space-time, logarithmic derivative, or probabilistic) for the aircraft pair (k,l). The higher the sum of the values of the individual metrics is, the more dangerous the airspace clearly is. Moreover, an advantage of this danger metric is that it can be compared across airspaces of different sizes because the individual danger metrics take into account the distance between aircraft and some measure of the time until the point of maximum danger.

Ideally, one would want to determine the range of values for the measure D that correspond to airspaces with high, medium, and low danger. This would require further empirical study comparing the frequency and type of accidents that occur in different airspaces with the values obtained for D in those airspaces. One way of matching values of the D metric with the occurence of accidents or operational errors by ATCs would be to measure the latter and examine the extent to which higher accident and error rates match higher multiples of the standard deviation above the mean for the metric D.

A disadvantage of this metric is that, by themselves, values taken on by D have no easily interpretable meaning. This can be remedied in two ways. The first is to determine, for whichever indicator f is being used, a function $\Psi_{k,l} = \Psi(f(k,l))$ that, given a value of the metric f, returns the probability that an undesirable event such as a collision or near miss will occur. Given Ψ, we compute the measure

$$D_{\text{exp}} = \sum_{1 \le k < l \le n} \Psi(k,l),$$

which represents the expected number of undesireable conflicts that will occur in the airspace within the time period (the near future) for which values of Ψ are specified. This measure of danger is appealing from the standpoint that, since it is desireable to specify a maximum expected number of conflicts that should occur per unit time in any airspace, we have a workable method for determining whether real measures for airspace safety are violated.

The second way to improve on the measure given by Equation 1 is a variation on the preceeding idea. Instead of determining the expected number of conflicts that will occur in the airspace over the time interval for which Ψ is computed, we could define an indicator variable

$$\begin{aligned} I(k,l) &= 1 \text{ if } f(k,l) > T^* \\ &= 0 \text{ if } f(k,l) \leq T^* \end{aligned}$$

where T^* represents a threshold value for the danger metric f which, when exceeded, requires intervention by the ATC. With this formulation, the number of situations in which the ATC must intervene is given by

$$D_{\text{int}} = \sum_{1 \leq k < l \leq n} I(k,l).$$

An advantage of this measure over D_{exp} is that the values of T^* can be determined in part by the ATC based on judgement and experience, whereas D_{exp} at the very least requires a realistic probabilistic simulation to establish some relationship between the probability of mishap and the value of the danger metric f. Also, this measure takes a more direct approach to determining the number of cases where the ATC will have to intervene, since the ATC decides whether or not to intervene on a case by case basis rather than by computing the expected number of mishaps.

Now that we have posited several reasonable ways to measure danger for an entire airspace as a function of the danger present in each individual aircraft pair, we turn to the central question: how is the complexity of the ATC's task related to the danger posed by potential accidents?

12.4 From Step 2 Danger to Step 2 Complexity

The principal measure of danger we will use in this section is D_{int}, because this metric most accurately caputures the number of interventions the ATC must make in the near future. As discussed earlier, the complexity of the situation facing the ATC in step 2 is closely related to the danger of the situation. Now that we have presented a workable metric for step 2 danger, we will explicate how the complexity of step 2 depends on the measure of danger D_{int} we have defined.

To assess the number of decisions the ATC has to make in steps 2 through 6, we must determine the number of possibilities open to the ATC to resolve a given conflict between n airplanes. To do so, we consider the following reasonable algorithm for solving conflicts: for each pair of conflicting airplanes in the group, solve the conflict while making sure the solution does not conflict with the constraints generated by any previously solved conflicting pair. Thus we have a problem in which each of the $\frac{n(n-1)}{2}$ conflicts constrains what choices we can make to resolve each of the other conflicts; in some cases, after the first k conflicts are solved, no solution for conflict $k+1$ may exist under the constraints, and thus backtracking may be necessary. In other words, this problem is a form of the general constraint satisfaction problem, which is known to be NP-complete.[13] Thus we postulate that the worst case complexity of the problem

[13] [8]

varies exponentially with the number of pairs: $O(k^{n(n-1)/2})$ for some constant k. To determine this constant, we consider the problem from a slightly different perspective.

Under the assumption that the altitude level of the planes is not open to modification, it seems fairly reasonable to suppose that the ATC can either tell both pilots to bank to the right (from their point of view) or to the left. Then for every pair of aircraft that is in conflict, there are two possible solutions open to the ATC for averting a collision or near miss.

Using the the metric D_{int} to determine the number of situations in which the ATC will have to intervene, and supposing that for each of these situations there are two possible solutions for altering the headings of the aircraft involved, we get that there are

$$S = 2^{D_{int}}$$

possible solutions for the system. Of course, in the near future, we assume that D_{int} refers to the measure of conflict within a given cluster, and consider the total complexity as the sum of the complexities within each cluster. More precisely, we let $S(i) = 2^{D_{\text{int}}(i)}$, where $D_{\text{int}}(i)$ is the danger measure for cluster i, and define

$$\text{Complexity} = \sum_{i=1}^{C} S(i) = \sum_{i=1}^{C} 2^{D_{\text{int}}(i)}, \tag{2}$$

where C is the total number of clusters. Given our assumptions, this provides a reasonable initial measure for the complexity involved in resolving the conflicts detected in step 2 of the ATC Decision Algorithm.

Nevertheless, it is obvious that the feasibility of the ATC's decision hinges upon the effect his preposed solution will have on nearby airplanes in the short term and afterwards. To improve our measure of step 2 complexity, we must take into account the conclusions of the ATC in steps 4 through 6. In other words, we must estimate how many operations are involved in checking whether a tentative rerouting decided upon in step 4 (based on conclusions made in step 2) will affect the routes of other airplanes in the near future.

The main factor we need to determine is the number of airplanes that should be considered "close" to an aircraft pair that needs to be rerouted. We use the characterization of "close" defined as part of the Clustering Algorithm we presented earlier. Given an airspace divided into C clusters, with n_i representing the number of planes in cluster i, it follows that there are $n_i - 2$ other airplanes the ATC must consider when resolving a conflict between two airplanes in cluster i. Thus there are (approximately)

$$2(n_i - 2)(D_{\text{int}}(i))$$

interactions that the ATC must consider in a given cluster i. At this point, it becomes clear that the number of interactions added by this measure does not change the fact that that the complexity of step 2 is $O(k^{n(n-1)/2})$. Thus, we argue that the metric for step 2 complexity given above by Equation 2 is a satisfactory proxy for our purposes.

12.5 Additional Factors Contributing to Workload Complexity

Up until this point, we have focused on defining two types of metrics for complexity: first, a metric related to the number of comparisons required of the ATC in step 1 and taking into consideration the processes involved in steps 4 through 6, and second, a metric that quantifies the role of danger and in creating complexity. The latter metric stems mostly from step 2 considerations.

These metrics capture the most important aspects of the air traffic workload complexity. Nevertheless, complexity is also affected by factors such as the rate of airplanes entering and exiting the airspace, the volume of the airspace, and the presence of additional software tools to automatically predict conflicts and alert the ATC.

The first of these factors depends on the rate of entry and exit of planes from the airspace at any given time. Under the (reasonable) assumption that it takes the ATC a certain block of time to hand over every aircraft exiting the airspace and receive every aircraft entering the airspace, we have that the complexity of this task can be represented by

$$\text{Distraction} \approx T\left|\text{Entry Rate} + \text{Exit Rate}\right|,$$

where T represents the amount of time (assumed approximately equal) that it takes the ATC to handle entries and exit from the airspace. The complexity of this task is linear.

The volume of the airspace, which has been taken into consideration implicitly by our measures of step 2 complexity, affects the job of the ATC in obvious ways. For a fixed number of planes, smaller airspaces will be harder to handle than larger airspaces because the number of planes per unit volume will be higher, so the danger metrics defined in the previous sections will tend to be higher. Naturally, dealing with eventual conflicts (as discussed earlier in the paper) will take even more of a secondary role to resolving more imminent conflicts. Given that many of the operational errors made by ATCs result from ignoring secondary conflicts for too long [14], this means that the potential for accidents is higher when the number of planes per unit volume of airspace is higher. The larger the volume of an airspace is, the more uncertainty the ATC faces in predicting eventual conflicts but the more leeway he has to reroute airplanes that are on conflicting paths.

One of the main secondary factors that deserves consideration is the effect of software to aid the ATC in predicting conflicts. The advantage of such software is that, by identifying conflicts and ordering them by the amount of danger they pose, it reduces the element of complexity that arises in step 1 of the ATC Decision Algorithm. Nevertheless, the primary complexity of the ATC's job comes from trying to figure out how to solve conflicts once they arise, a fact which follows from the discussion in the previous sections ($O(n^2)$ for step 1 versus $O(2^{n(n-1)/2})$ for step 2). This suggests that, while helpful in reducing one

[14] see [10]

type of complexity, programs designed to detect conflicts might not be able to combat the primary source of complexity faced by the ATC. The primary danger of software designed to identify conflicts for the ATC is that it could cause the ATC to take a more passive attitude in searching for potential conflicts not identified by the software. Ignoring secondary (or eventual) conflicts until too late is a primary source of operational errors in air traffic control, and there is a possiblility that software aids could worsen this problem. As a point of synthesis regarding the use of software in air traffic control, we suggest that programs designed to aid the ATC in identifying conflicts be designed as a guide to the ATC's own judgement rather than a way of automating any of the functions performed by the ATC.

12.6 Conclusion

In this section, we have derived several metrics for the complexity of the problems facing the ATC in his attempt to identify and correct potential problems between aircraft in a dynamic environment. To clarify the problem, we introduced the ATC Decision Algorithm as a model for the different types of tasks confronting the ATC. We then examined the two primary types of complexity that were suggested by this algorithm, namely step 1 and step 2 complexity. The first of these corresponds to the difficulty involved in identifying aircraft between which a potential conflict might occur and aircraft whose current trajectories suggest the possibility of an eventual conflict. The second type of complexity refers to the difficulty involved in determining, given that two airplanes are near each other in the airspace, whether they present a situation in which the ATC should intervene to avoid a collision, near miss, or lesser airspace violation. We determined that step 1 complexity is of order $O(n^2)$ and that step 2 complexity is of order $O(2^{n(n-1)/2})$, where n represents the number of airplanes in a given cluster of the airspace. It is possible, of course, that we consider the airspace itself to be one cluster, in which case the above measures represent the step 1 and step 2 orders of complexity of the entire airspace. If the airspace can be divided into more than one cluster, we have shown that from the perspective of the ATC (and the number of required decisions/assessments required in the ATC Decision Algorithm) the complexity of the entire airspace is the sum of the complexity measure for the individual clusters.

The primary consideration for step 2 complexity follows from extending the results relating to assessing the danger between the airplanes in an aircraft pair. Three metrics for assessing danger were presented and discussed, and we argued for D_{int}, the number of interventions required of the ATC based on the exceeding of certain danger criteria, as the best measure of danger for determining the compexity of step 2. As for step 1, we considered the analysis of step 2 complexity as it relates to demands placed on the ATC by steps 4 through 6, and modified both measures to take the interactions between clusters and the interactions between the conflict pair and other airplanes in the cluster into account.

The connection between the number of airplanes in the airspace at any one time and the complexity of the airspace is implicit throughout our analysis. The number of airplanes present during any given interval of time affects our complexity analysis in much the same way as the number of aircraft present at any given time, in the sense that the average number of airplanes in the airspace at any given time that are part of a larger cluster is the primary factor in determining complexity. The only real difference that this distinction makes comes from considering a situation where the average time each airplane spends in the airspace is small, since in this case step 2 (and 4 through 6) complexity factors (identifying planes with short term path conflicts and rerouting them) will be relatively more important since the flight paths of other airplanes are harder to predict and the airspace is potentially smaller. Also, the order $O(n$ per unit time) complexity due to clearing planes to enter and exit the sector will be higher because the rates of entry and exit are likely to be higher in this situation. Fluctuations in the air traffic over a 24 hour period[15], judging by patterns exhibited in major European areas, consist of low night time plateaus, a sharp rise, and a higher midday plateau with the number of airplanes in the aircraft sector. The complexity of each of these periods depends in the manner discussed earlier on the number of planes present in the airspace at a given time and the extension of the danger measures from the first portion of the paper to the complexity metric of part 2 in the ATC Decision Algorithm. In the period of sharp rise and sharp decline, clearance effects will be more important than usual, as will step 2 effects when it is more difficult for the ATC to assess how to deal with potential conflict situations. For the higher plateau, step 1 effects will be higher than the other periods of traffic. In sum, we believe the combination of the complexity metrics we have derived takes the potential complexity from short and long term conflicts into account reasonably well for all of these periods of intraday variation by considering the primary factors affecting the specific stages of the ATC's decision process over a given period of time.

A Summary for presentation to FAA administrator

In an attempt to improve safety and reduce workloads on air traffic controllers, the FAA is considering software that would automatically alert controllers to potential collisions between planes. To assist in the development of this software, we first created and tested five models, each of which gives a metric for the danger presented by two airplanes flying in space. For all metrics, higher measures correspond to more imminent danger between the two planes, and thus each metric suggests a priority ordering by which the controller should attend to problems; additionally, controller intervention for a particular conflict is necessary when the situation's danger measure becomes higher than some predefined threshold value. We propose five models for solving this problem: the trivial model (which simply calculates whether a collision or near miss will occur,

[15] see [6]

assuming linear trajectories with no uncertainty), the probabilstic simulation model (which finds the probabilities of a collision or near miss, by assuming Gaussian parameter distributions and performing a large number of random trials), the close-approach model (which calculates danger as a function of the two airplanes' minimum distance and the time until that minimum distance is reached), the space-time model (which extends the close-approach model to four dimensions by calculating a minimum spacetime "distance" between two airplanes), and the logarithmic derivative model (which calculates a lower bound on the time until closest approach).

While each of the five proposed models has advantages and disadvantages, we demonstrate that the close-approach model is particularly successful in proposing a danger metric which is both accurate and efficient to calculate. When the models were tested on various sample cases, the close-approach method computed relative danger values that corresponded closely with our intuitive understanding of the situations, obtaining the lowest RMS error with respect to an intuitive ranking of test cases by difficulty. The probabilistic simulation model also performed well on the test cases (though slightly worse than the close-approach model), and also has the advantage of dealing with some of the uncertainty present in the environment; it is more computationally intensive than the close-approach method, but not unreasonably so. The space-time method may be more accurate than the close-approach method, but is also slower to compute, more difficult to implement, and more opaque to the controller. Finally, the logarithmic derivative model is likely to closely approximate a human observers's intuitions of how fast two planes are approaching, but as a result it loses an accurate measure of whether the two planes will actually collide. Finally, we take factors such as plane maneuverability and response time into account in order to obtain another measure of minimum safe distance for two approaching planes.

Next, we expand on the two-plane case by considering the complexity of an entire airspace sector with respect to the number of aircraft and the number of potential conflicts. To do so, we present an algorithm as a model for the decision process undertaken by an air traffic controller in detecting and solving conflicts. We examine each step of this algorithm individually to obtain a measure of its complexity. In particular, the step of scanning for potential conflicts was found to vary quadratically in complexity as a function of the number of airplanes, either based directly on the number of airplane pairs, or more accurately as a sum of complexities of individual clusters of airplanes. Next we present several methods for calculating a danger metric for the airspace as a whole, based on the sum of some measure (danger, number of conflicts, etc.) over all possible pairs or clusters of airplanes. We next compute the complexity of conflict resolution for a given cluster of n planes: this is an NP-complete problem with worst case complexity $2^{n(n-1)/2}$. Finally, we discuss other factors contributing to controller workload complexity, including airplanes entering and exiting the airspace, weather conditions, and the presence of software tools for conflict prediction.

B Probabilistic Simulation Program

This program implements the Probabilistic simulation model described in 6

```
#include <iostream.h>
#include <math.h>
#include <stdlib.h>

#define PI 3.14159265359

int BAD=18240;
int WORSE=700;
int WORST=200;
int n=10000;
float c1=50;
float c2=5;
float c3=0.05;

float phi(float x)
{
   // Abramowitz & Stegun 26.2.19
   float
      d1 = 0.0498673470,
      d2 = 0.0211410061,
      d3 = 0.0032776263,
      d4 = 0.0000380036,
      d5 = 0.0000488906,
      d6 = 0.0000053830;

   float a = fabs(x);
   float t = 1.0 + a*(d1+a*(d2+a*(d3+a*(d4+a*(d5+a*d6)))));

   // to 16th power
   t *= t;  t *= t;  t *= t;  t *= t;
   t = 1.0 / (t+t);  // the MINUS 16th

   if (x >= 0)  t = 1 - t;
   return t;
}

float phiinv(float p)
{

   // Odeh & Evans. 1974. AS 70. Applied Statistics. 23: 96-97
   float
      p0 = -0.322232431088,
```

```
      p1 = -1.0,
      p2 = -0.342242088547,
      p3 = -0.0204231210245,
      p4 = -0.453642210148E-4,
      q0 =  0.0993484626060,
      q1 =  0.588581570495,
      q2 =  0.531103462366,
      q3 =  0.103537752850,
      q4 =  0.38560700634E-2,
      pp, y, xp;

   if (p < 0.5)  pp = p;  else  pp = 1 - p;

   if (pp < 1E-12)
      xp = 99;
   else {
      y = sqrt(log(1/(pp*pp)));
      xp = y + ((((y * p4 + p3) * y + p2) * y + p1) * y + p0) /
               ((((y * q4 + q3) * y + q2) * y + q1) * y + q0);
      }

   if (p < 0.5)  return -xp;
   else  return  xp;
}

float getrand(float mu,float sigma)
{
  float r=rand()+0.5;
  return mu+phiinv(r/32768)*sigma;
}

int main()
{
   float x,y,mv1,mv2,sv1,sv2,ma1,ma2,sa1,sa2;
   float v1,v2,a1,a2,vx,vy;
   float dot,normp,normv,mindist,mintime;
   int bad=0,worse=0,worst=0;
   float min1=-1,min2=-1,min3=-1;
   int limit;
   cout << "Enter initial x,y coordinates of plane 2: ";
   cin >> x;
   cin >> y;
   mv1=mv2=811; //cout << "Enter mean velocities of planes 1 and 2: ";
   //cin >> mv1;
   //cin >> mv2;
   sv1=sv2=40; //cout << "Enter sigma velocities of planes 1 and 2: ";
```

```
//cin >> sv1;
//cin >> sv2;
cout << "Enter mean angles of planes 1 and 2: ";
cin >> ma1;
cin >> ma2;
sa1=sa2=1; //cout << "Enter sigma angles of planes 1 and 2: ";
//cin >> sa1;
//cin >> sa2;
limit=-1; //cout << "Enter time limit (-1 for none): ";
//cin >> limit;
normp=sqrt(x*x+y*y);
for (int i=0;i<n;i++)
{
    v1=getrand(mv1,sv1);
    v2=getrand(mv2,sv2);
    a1=getrand(ma1,sa1);
    a2=getrand(ma2,sa2);
    vx=(v1*cos(a1*PI/180))-(v2*cos(a2*PI/180));
    vy=(v1*sin(a1*PI/180))-(v2*sin(a2*PI/180));
    dot=x*vx+y*vy;
    normv=sqrt(vx*vx+vy*vy);
    mindist=normp*normp-(dot*dot/(normv*normv));
    if (mindist<0) mindist=0; else mindist=sqrt(mindist);
    mintime=dot/(normv*normv);
    if (mintime<0)
    {
       mintime=0;
       mindist=normp;
    }
    if ((limit!=-1) && (mintime>limit))
    {
       mintime=limit;
       mindist=sqrt((x+limit*vx)*(x+limit*vx)+(y+limit*vy)*(y+limit*vy));
    }
    if ((mindist<BAD) && (mindist>=WORSE))
    {
         bad++;
         if ((min1==-1) || (mintime<min1)) min1=mintime;
    }
    if ((mindist<WORSE) && (mindist>=WORST))
    {
         worse++;
         if ((min2==-1) || (mintime<min2)) min2=mintime;
    }
    if (mindist<WORST)
    {
```

```
            worst++;
            if ((min3==-1) || (mintime<min3)) min3=mintime;
      }
   }
   cout << "In " << n << " flights there were:" << endl;
   cout << worst << " collisions" << endl;
   cout << worse << " near misses" << endl;
   cout << bad << " violations" << endl << endl;
   cout << "Minimum time to collision: " << min3 << endl;
   cout << "Minimum time to near miss: " << min2 << endl;
   cout << "Minimum time to violation: " << min1 << endl;
   vx=(mv1*cos(ma1*PI/180))-(mv2*cos(ma2*PI/180));
   vy=(mv1*sin(ma1*PI/180))-(mv2*sin(ma2*PI/180));
   dot=x*vx+y*vy;
   normv=sqrt(vx*vx+vy*vy);
   mindist=normp*normp-(dot*dot/(normv*normv));
   if (mindist<0) mindist=0; else mindist=sqrt(mindist);
   mintime=dot/(normv*normv);
   if (mintime<0)
   {
         mintime=0;
         mindist=normp;
   }
  if ((limit!=-1) && (mintime>limit))
  {
         mintime=limit;
         mindist=sqrt((x+limit*vx)*(x+limit*vx)+(y+limit*vy)*(y+limit*vy));
  }
  cout << "Danger measure 1: " << (mindist<700) << endl;
  cout << "Danger measure 2: "
       <<  1/((mintime+c1)*(mintime+c1)*(mindist+c2)) + c3/normp << endl;
  cout << "Danger measure 3: "
       << dot/(normp*normp) << endl;
  return 0;
}
```

References

[1] Mahalingam, S. Air Traffic Control. New Delhi: Kaveri Books, 1999.

[2] Federal Aviation Administration,
http://www.asy.faa.gov/asy_internet/safety_data/default.htm

[3] Denker, John S., *See How It Flies*,
http://www.monmouth.com/~jsd/how/htm/maneuver.html

[4] Airbus Industries, `http://www.airbus.com/imperial.html`

[5] Wiener, E. and D. Nagel. Human Factors in Aviation. San Diego: Academic Press, 1988.

[6] MAICA: modeling and analysis of the impact of changes in ATM, Transport Research #71.

[7] MICA: MET improvement for controller aids, Transport Research #72.

[8] Vardi, M. "The descriptive complexity of constraint satisfaction", Implicit Computational Complexity 1999,
`http://www.cs.indiana.edu/icc99/vardi.html`

[9] Endsley, M. and M. Rodgers. "Situation Awareness Information Requirements for En Route Air Traffic Control", DOT/FAA/AM-94/27, 1994.

[10] Endsley, M. and M. Rodgers. "Distribution of attention, situation awareness, and workload in a passive air traffic control task", DOT/FAA/AM-97/13, 1997.

[11] H. Kremer et al. "Probabilistic vs, Geometric Conflict Probing", National Aerospace Laboratory NLR, 1997.

[12] L.V. Lauderman et al. "Dynamic Density: An Air Traffic Management Metric", NASA, NASA/TM-1998-112226, 1998.

[13] M. Rodgers et al. "The Relationship of Sector Characteristics to Operational Errors", DOT/FAA/AM-98/14, 1998.