

COMPUTATION AND ANALYSIS OF AIRCRAFT PROXIMITIES IN THE NAS USING ETMS DATA

Dr. Alexander Klein

George Mason University, Fairfax, Virginia

Florian Hafner

Embry-Riddle Aeronautical University, Daytona Beach, Florida

Abstract

Potential airborne conflicts or near-conflicts (jointly referred to as aircraft proximities) are an important National Airspace System activity indicator by itself and also as a component of the Dynamic Density metric. Proximity analysis in four dimensions using historical data (specifically, Enhanced Traffic Management System files) presents a number of challenges. If we use actual flight tracks, they have conflict resolution actions by Air Traffic Control already embedded into them. If we use flight-plans-as-filed, the time component of the 4D tracks needs to be constructed correctly. To simply use the filed departure time or the estimated/actual departure time as the start time for such tracks would distort the timing and location of the potential airborne conflicts because of the wide variance of out-to-off times for different airports and times of day. In this paper we propose a method that uses a combination of actual and planned tracks resulting in a more realistic timing of potential conflicts. An efficient scanning algorithm for computing 4D proximities is introduced. The results are displayed on a hexagonal grid over the National Airspace System. The method is also compared with the output from fast-time simulations using TAAM (Total Airport & Airspace Modeler), a well-known air traffic simulation model.

Introduction

The objective of this research is to compute aircraft proximities, actual and especially potential (“what could have happened had the controller not intervened”), as a factor contributing to controller workload. It is not our objective to assess safety. Besides, the accuracy of the Enhanced Traffic Management System (ETMS) data, as well as the accuracy of the proximity computation method

presented here, may not be sufficient for highly detailed safety analysis. But for our purposes – estimating total amounts of aircraft proximities in large portions of airspace as part of the Dynamic Density or workload assessment – the data and the method are deemed adequate.

We have attempted to make the proximity computation method as fast and efficient as possible, while keeping the fidelity at an adequate level for NAS-scale analyses.

The importance of correct timing adjustment in 4D conflict computations based on historical data can be illustrated by the following picture.

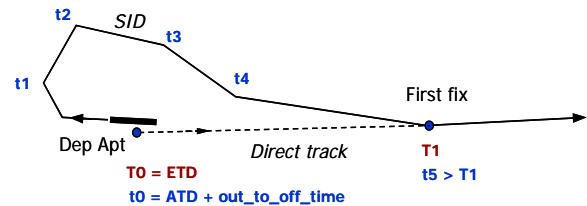


Figure 1. As-filed and as-flown tracks

If we use the ETMS scheduled departure time (STD) or the actual departure time (ETD) (T_0 in Figure 1) and a direct track from departure airport to the first fix/waypoint (T_1 in Figure 1), the aircraft would get there much sooner than it would have done in reality (t_5 in Figure 1). Ideally, we need to take into account the taxi-out time, runway queuing delay, and time on the instrument departure (SID). Without this, timing and location of potential conflicts and proximities on the day in question would not be computed correctly. Our paper describes a method addressing this issue.

Comparisons between different proximity computation methods, including output from a high-fidelity simulation model for calibration purposes, are presented further in this paper.

Background

Preventing potential separation violations – airborne conflicts or “proximity events” (pairs of aircraft that would have enough of a separation but not much of an added margin) – is a vital part of an air traffic controller’s job and a major contributor to the air traffic controller’s workload. As such, data on potential (hopefully not actual) separation violations are an important indicator of National Airspace System (NAS) activity and a major component of workload-related metrics.

A metric related to aircraft proximity, called Track Intersects, was developed for the Intersect Density Analysis Tool (IDAT) by MITRE CAASD [1]. From the description provided for IDAT, it seems that track intersects were computed in three dimensions and that filed flight plans were used rather than actual tracks. This provides a valuable insight into NAS structure and potential controller workload bottlenecks; and having the time component as the 4th dimension would further increase the usability of this tool.

Another approach, proposed by Donohue et al [2], used TAAM, a fast-time airport and airspace simulation tool developed by Preston Aviation Solutions (a subsidiary of The Boeing Company), to find potential conflicts in the NAS airspace divided into hexagonal prismatic “bins” with a 24-NM edge, rather than conventional airspace sectors. This computation was used as a key component of the Sector Workload indicator calculated by TAAM. This method can be very useful, especially for future traffic analysis or airspace redesign as TAAM uses a sophisticated 4D conflict probe.

An extension of the quoted research would be to add runways or use queuing models instead of just point-airports which in TAAM are simple objects with no queuing capability. When point-airports are used, aircraft depart exactly at their gate ETD, even from airports like Chicago O’Hare (ORD), New York Laguardia (LGA) or Dallas Ft. Worth (DFW). This reduces the accuracy of TAAM conflict detection in terms of timing and location of the proximities detected, as illustrated by Figure 1, although pre-processing the data by adding average historic taxi-out times at key airports to the ETDs from ETMS may improve this measure.

The experimental work by Wieland et al [3] delivered a substantial improvement of TAAM conflict detection performance and the results are planned to be included in the mainstream TAAM release. One issue regarding the use of sophisticated simulation models like TAAM is the ownership of, or access to, a TAAM license, as well as training, data preparation, simulation setup and simulation output post-processing. In the course of this research, we investigate if a simpler, faster software program can be used for the same purpose but with very little setup or post-processing effort and with a single data file (such as ETMS) as its input.

Conflict analysis as a key part of operational concepts validation was conducted using other simulation models, notably RAMS. Crook and Liang [4], for instance, examine workload and traffic complexity indicators in upper airspace (above FL350), distinguishing between different types of conflicts and their contribution to controller workload.

Dynamic Density is a widely accepted metric related to controller workload [5, 6].

Research conducted by Baart [7] indicates, among other things, that the FAA Technical Center’s dynamic density metrics and their calculation method seem to produce a better result than those obtained from other models including RAMS. This may be due to the already-noted fact that RAMS is a high-fidelity en-route traffic simulation model but, just like TAAM, it does not currently support queuing for point airports and its departure queuing and delay simulation capabilities for major airports are limited. As a result, 4D flight trajectories may be shifted in time compared to actual departure times.

NAS-wide airspace analysis toolsets such as NARIM feature 4D trajectory generation and conflict analysis. As reported by Weiss et al in [8], NARIM processing time for computing proximities in the entire NAS can be as fast as 15 minutes.

The FAA’s Sector Design & Analysis Tool (SDAT) also has a 4D proximity analysis feature [9]. Flight plan segments of each active aircraft are examined against flight plan segments of all other aircraft, with extensive use of 4D filtering to reduce the amount of computations. Proximity analysis for

one Air Traffic Control (ATC) Center can be performed in approximately one minute.

While ETMS planned tracks are being used for conflict analysis, in our literature search we have not come across a discussion on the time component of the 4D ETMS-sourced trajectories used in NARIM. Therefore, it seems worthwhile to compare actual, planned and simulation-model-generated 4D trajectories in terms of resulting conflicts and proximities.

Conflict detection methods developed by NEXTOR researchers (see, for instance, a report by Trani et al [10]) apply methodologies developed by member universities as well as used in tools like NARIM. In this report, the importance of correct timing in computing potential conflicts is underscored by the nature of tasks at hand, such as RVSM-vs.-Step-Climb analysis. This notion provided additional inspiration for our research.

ETMS Traffic Data

The ETMS air traffic data stream – also known as the Enhanced Traffic Management System – is a data feed available from the FAA and contains relevant flight planning and tracking information on aircraft in the National Airspace System (NAS). As part of the overall architecture, ETMS also provides a set of front-end tools, which are used by FAA facilities to plan traffic flows and other activities used in NAS planning and operation.

The ETMS data feed includes all relevant information for aircraft operating in controlled airspace, including flight plans, amended flight plans, cancellations, arrival/departure messages, as well as 3D track information where available. The latter is a stream of so-called “TZ” messages.

Data contained in the ETMS data feed is available to industry via the so-called ASDI (Aircraft Situation Display to Industry) feed from the Volpe center. The ASDI feed contains the vast majority of messages from ETMS but lacks any confidential government and military operations.

Computing Proximities from Archived Data: Challenges

It would be easiest if we could use the readily available actual 4D tracks from ETMS data for

aircraft proximity calculation. But actual tracks already have conflict resolution actions by air traffic controllers embedded in them, which is to be expected. Therefore, at best, actual tracks could give us just a “faint” indication of where the more serious potential conflicts might have been.

Using flight-planned tracks for proximity calculations is a better choice, especially if the last-updated (before departure) flight plans are used. The flight-planned track represents the desired 4D path of the aircraft; analyzing these paths can yield a better picture where the potential proximities and separation violations could have occurred had the air traffic controllers or pilots not taken the corrective actions.

However, this case presents the researcher with a number of challenges. First, as mentioned in the prior research overview, if we use as-planned departure times and times at all waypoints, then the timing (and, in part, location) of many 4D conflicts would be distorted. This is especially true for traffic flows from airports with long out-to-off times as it mixes with traffic from airports with short out-to-off times. This discrepancy will be illustrated later.

Second, we need to deal with different flight plan versions: historical flight plans (ETMS RT Route Expansion messages), possibly multiple versions of flight plans filed prior to departure (ETMS FZ messages), as well as flight plan amendments en route (ETMS AF messages), such as direct-to vectors, cruising altitude changes etc. In the latter case, it would be desirable to exclude conflict resolution actions (“ABC123, descend and maintain FL350 for traffic”).

Third, we need to decide if whole flight plan segments will be used for proximity calculations or the segments will be split, by interpolation, into “pseudo-TZ” 4D position updates with e.g. 1-min frequency.

Related to this is the computational challenge. Comparisons of vast numbers of flights, flight plan segments or position updates for NAS-scale traffic can be quite costly in computational terms.

And, finally, the accuracy of the proximity computations needs to be analyzed. The required accuracy for NAS-level trend analyses is not necessarily the same as for mission-critical real-

time conflict detection and avoidance; but it is desirable to keep it as high as practical.

Point Sequences or Segments?

To use segments and calculate their 4D intersections would be a legitimate approach. One would, however, need to design an efficient segment sorting technique (placing them in 2D, 3D or 4D “bins” or combinations of 3D and time “bins”). For instance, one could start with min/max coordinate box enclosures for entire flights and sort them, in the U.S. case, into spatial “bins” such as North-East, South-East, California etc. Within each bin, time sort could be performed as well. Then, box enclosures of individual segments can be compared if the whole-flight-path boxes of the two flights intersect, and finally, detailed 4D intersections can be computed. Depending on segment lengths, great circle routes may need to be considered when box enclosures are generated, which could affect the performance of the algorithm.

A similar method was applied in TAAM in fast-time simulation environment. When a new aircraft was about to take off, TAAM conflict detection mechanism would check it against all other airborne aircraft using an extensive set of smart filtering techniques including 4D full-flight-path box enclosure checks. This method was quite efficient for small to medium size scenarios but less so for massive, NAS-scale simulations which could take many hours and in which the share of conflict detection could reach 50% of total run time. As mentioned in the Introduction, a more efficient method has since been developed by Wieland et al [3] but it relies on the TAAM conflict probe and the quad-tree computation rather than the flight segments.

Segment-based computations require that each aircraft’s flight plan is checked against all other aircraft’s flight plans; that is, the algorithm is essentially $O(N^2)$ vis-à-vis the number of flights or the time interval, even though the non-linearity can be greatly reduced through smart filtering. The algorithm presented in this paper, on the other hand, can be linear, at least in regard to the length of the traffic sample’s time interval. This will be discussed in the next section.

Additionally, unlike the en-route airspace, for terminal airspace maneuvers (including radar vectoring) it may be difficult to relate aircraft paths to flight segments, so position-by-position comparisons of point-sequence tracks may be more appropriate.

To conclude, the use of position-sequence tracks may be more efficient computationally for NAS-scale traffic analysis when the position update frequency is not too high. Also, when analyzing terminal airspace, position-by-position comparison may work better than the segment-by-segment comparison. And, when analyzing actual tracks for any conflicts or near-conflicts, e.g. for quality assurance, point-by-point computation must be used. On the other hand, for massive traffic samples with very-high-frequency position updates, segment-by-segment comparison may become more efficient.

In this paper, we will concentrate on position-by-position comparisons.

ETMS File Scanning Algorithm

The algorithm described below takes advantage of the fact that the ETMS file is sorted by timestamp, as illustrated in Figure 2. For convenience, the ETMS file sample is “compressed” in time, i.e. many records with timestamps between 001409 and 001509 have been removed to make the sample short enough. Also, only the TZ type records are shown.

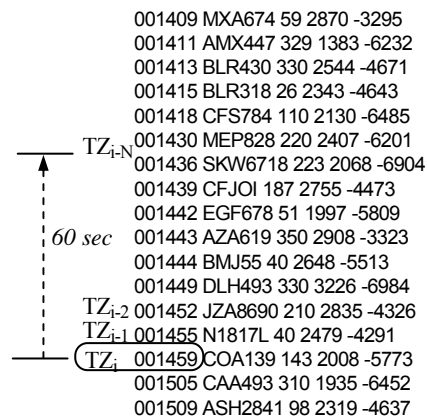


Figure 2, Scanning ETMS records

The process is as follows:

- a) The algorithm moves down the list of TZ records: $TZ_1, TZ_2, \dots, TZ_i, \dots, TZ_{Last}$.
- b) For each record TZ_i (such as, for instance, 001459, third from the bottom in Figure 2), it scans the file backward – *up to 60 seconds* – and searches for 4D proximities between TZ_i and each of the previous records, TZ_{i-j} , where $j = 1, \dots, N$, within the 60-sec time span.
- c) When a pair of TZ records, (TZ_i, TZ_{i-j}) is compared, the algorithm first checks their altitude difference.
 - If it exceeds the vertical separation (1000 or 2000ft), further checking is abandoned.
 - If the altitude difference is less than the required vertical separation, then the horizontal separation is checked.
 - For horizontal separation, a min-max box comparison is performed first for the two positions, and if it is obvious that they are far enough apart, further checking is abandoned.
 - Otherwise, full distance comparison is performed. Even then, we actually compare squared distances, avoiding the square root function as too computationally-expensive.
 - The proximity level is then calculated (see next section).
- d) When the time difference between TZ_i and TZ_{i-N} exceeds 60 sec, the process for our base record, TZ_i , is completed. The algorithm then increments i by one, i.e. takes the next record down the list, TZ_{i+1} , and again looks backward within the 60-sec time interval, repeating steps (b) – (d).

If the two aircraft are more than 60 sec apart in time, no matter where they are location-wise, they are not in conflict at this time. They may get closer later, or may have been closer earlier, but that would be ascertained by the algorithm checking other 4D position pairs for these two aircraft. This is because true air speeds of aircraft above FL180 are about 5-8 NM / minute, so chances of encounters less than 5 NM apart are practically nil

for respective pairs of TZ positions that are more than 1 minute apart.

This time span has been selected for ETMS file type where aircraft location report frequencies are usually 1 minute; if this mechanism is used on higher-fidelity data (e.g. using the 4.7-sec radar position update frequency) and with smaller separation distance (e.g. 3 NM instead of 5 NM), the 60-sec interval can be reduced.

As can be seen, the above algorithm is linear vis-à-vis the length (time interval) of the input data sample of the same frequency. It is, however, of the $O(N^2)$ type in respect of the frequency of position reports, as well as in respect of the amount of traffic (e.g. a Center vs. NAS).

One way to reduce computation time is to conduct a secondary – geographical - sort of TZ positions, in addition to timestamp sorting, and to divide the airspace into “bins”. TZ positions in geographical “bins” can then be processed in parallel.

The base algorithm has proved to be very fast: the computation of all proximities for the entire day of NAS takes about 5-6 minutes on a mid-range Windows laptop. This does not include pre-processing of an ETMS file (such as sorting TZ and FZ track points by flight). Embry-Riddle Aeronautical University’s software performs this type of pre-processing with good efficiency.

Proximity Levels

We distinguish between three proximity levels. Level 1 proximity means essentially a real potential conflict while proximities Level 2 and 3 are less severe. Table 1 summarizes the settings for a pair of 4D TZ hits.

Table 1. Proximity Level definitions

Proximity Level	Vertical separation	Horizontal separation	Time separation
1	< 2000 ft at or above FL290, <1000 ft below FL290	< 5 NM (<100% std)	< 10 sec
2	Same as previous	5 - 7.5 NM (100-150%) OR < 5 NM	< 10 sec < 20 sec

3	Same as previous	7.5 – 10 NM (150-200%) OR 5 – 7.5 NM OR < 5 NM	< 10 sec < 20 sec < 30 sec
---	------------------	--	--

As mentioned above, these parameters can be adjusted if required, e.g. when the target separation is smaller and the position update frequency is higher.

Experiments with Time Synchronization for Pairs of TZ Hits from Different Aircraft

We have investigated two methods aimed at increasing the accuracy of proximity computation.

Let the first timestamp actually recorded for Aircraft 1 be T_1 and the other timestamp (for Aircraft 2) be T_2 , so that $T_2 - T_1 = \Delta T$. If the algorithm determines that the two positions are separated by less than the specified distance and don't have enough vertical separation, it checks how far apart their respective timestamps were. If the timestamps were more than a few seconds apart, the algorithm attempts to reduce uncertainty by creating additional estimated positions where the two aircraft would have been at the *same* time respectively (shown as grey images in Figure 3).

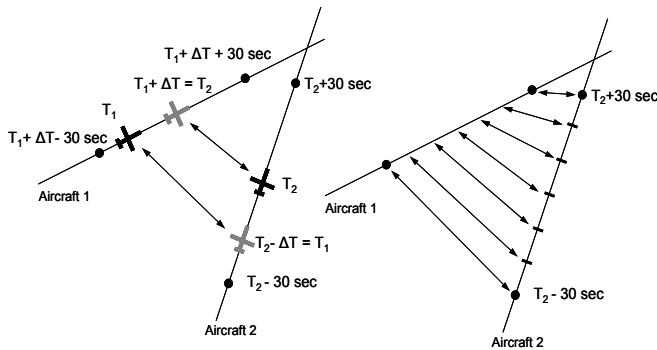


Figure 3. Synchronizing aircraft positions

The first method compares two pairs of TZ hits (Aircraft 1 black with Aircraft 2 grey and Aircraft 1 grey with Aircraft 2 black, respectively, see the left-hand-side drawing in Figure 3).

In the second method, we use interpolation. This is illustrated by the right-hand-side drawing in

Figure 3. We use the estimated-velocity vectors (obtained during pre-processing the ETMS file from current/previous TZ hits for each aircraft) to calculate where these aircraft would be at N synchronized moments in a ± 30 second interval, counting from T_2 . For Aircraft 1 this means that we are looking at the $[T_1 + \Delta T - 30, T_1 + \Delta T + 30]$ interval.

Pairs of these adjusted positions of two aircraft are compared and the closest-distance pair is selected.

Results from these experiments show that the proximity numbers and locations were rather similar to those with no synchronization. Observed differences with the original non-synchronized comparison method were about 5-7%. However, the computation time increases by 200% for the first method (two pairs instead of one) and by more than 400% for the second method (multi-point interpolation), due to the need to use trigonometric functions which are very expensive computationally when we deal with vast numbers of comparisons.

Computing Proximities from Planned Tracks (ETMS FZ Records)

For this analysis, we have used the EXP (Export Library Format) files built by Embry-Riddle Aeronautical University's Center for Applied Air Traffic Management Research (EARL). This file is created by processing an ETMS/ASDI file, sorting it by flight ID, and recording separate TZ and FZ tracks (the actual, or TZ, track is stored as 4D data; the planned, or FZ, track – in the form of an RT Route Expansion – is stored as 2D data in the EXP file). The filed flight plan (FZ) 2D expansions used for the EXP data files are the first observed for each flight. Subsequent re-filing or amendments are recorded in the dataset but not expanded for use.

The algorithm augments the 2D FZ track data and turns it into 4D data. It assigns the time and altitude of the first TZ record to the first FZ track point (the origin airport). The algorithm then takes each subsequent FZ track point, finds two TZ track points that are closest to it, and interpolates between the time and altitude of these two TZ track points to set the time and altitude of the FZ track point (waypoint).

The approximate locations of two additional points, Top-of-Climb (TOC) and Top-of-Descent (TOD) are calculated and inserted into the FZ track.

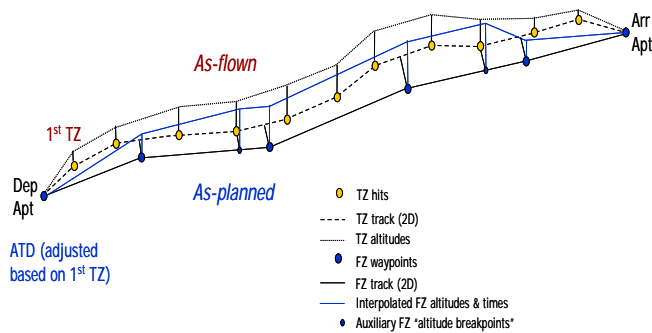


Figure 4. Creating 4D filed track from actual radar hit points (TZ hits)

The result is a time-adjusted, four-dimensional FZ track whose time component conforms rather well to the actual timeline of the flight but whose trajectory is still the flight-planned, not actual, track. Note that airborne holding and other sequencing actions may be embedded into the times of FZ track waypoints, which seems reasonable because times at subsequent waypoints will include this airborne delay and will therefore conform more closely to the actual timeline of the remaining flight. But the positions of the FZ waypoints do not change (they remain as in the filed flight plan), regardless of whether the aircraft was in a holding pattern or was vectored. Since in this study we are only concerned with flight tracks above FL180, most airport specific sequencing, vectoring, and holding effects on the FZ-TZ position (not time) synchronization should be factored out of this analysis, whereas strategic en-route flow control measures are implicitly included.

In order to use our proximity computation algorithm, we need to generate pseudo-TZ tracks from the FZ data. This is easy to accomplish by simple interpolation since the FZ tracks are now four-dimensional. The result is recorded into an ETMS-like file using the same format. The FZ (planned) track data is now ready for proximity analysis. The ETMS file scanning algorithm can be applied, as described earlier in this paper.

Extracting Proximity Information from TAAM Output

Proximity calculations from an ETMS file using the method presented above were compared to the results generated from TAAM simulation runs.

We have run a NAS-scale TAAM simulation scenario using ETMS/ASDI FZ records which TAAM converts to its flight plan format. Only point airports were used and in the initial test, no ETD adjustments were made and no TAAM sequencing algorithms were used in the exercise. The day chosen was the same as for ETMS-derived flights and the number of flights in all tests was approximately 56,000.

In addition to setting up a NAS-wide TAAM simulation scenario using the same flight plans as in our algorithm tests, we have developed a post-processor that extracts the necessary information from a TAAM report file in the same format as our proximity calculation application.

TAAM uses a 4D conflict probe, shown on the next picture, with user-selectable look-ahead time, typically 5-7 min. Predicted positions (shown as red vertical bars) are computed continuously every 6 seconds in fast time during the simulation. The probe works equally well in en-route and terminal airspace where extensive 4D maneuvering is required. Note that TAAM is a fast-time simulation tool and is not used in operational, “frontline” conflict detection or resolution.

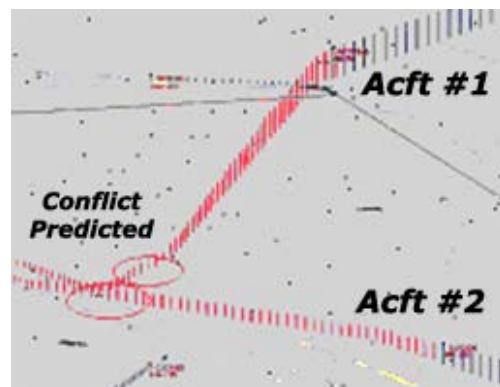


Figure 5. TAAM conflict probe

When TAAM reports conflicts, it grades them by potential severity and records approximately 30 parameters for each conflict, including aircraft positions, altitudes, speeds, rates of change of altitude, and also the location and time of the closest proximity, etc. This information is stored in an ASCII-text form in the TAAM report file. For our purposes, we extract the following parameters: the IDs of the two aircraft, as well as the timestamp, location and altitude of the point of closest proximity.

TAAM conflict severities are reported in a similar fashion to the one adopted in this paper: potential conflicts (< 100% horizontal separation) and near-conflicts (100-120%, 120-150%, and 150-200% horizontal separation observed; insufficient vertical separation in all cases). We combined the 100-120% and 120-150% intervals into one so TAAM results can be compared directly with our algorithm's.

Proximity Computation Comparisons

We have compared four methods of proximity computation in NAS Class A airspace (FL180-600):

- From actual (TZ) tracks;
- From planned (FZ) tracks, departure and waypoint times not adjusted;
- From planned (FZ) tracks, departure and waypoint times adjusted using TZ tracks;
- From TAAM, point airports only (no sequencing), departure times not adjusted.

Visualization

As might be expected, proximities computed from actual ETMS TZ tracks are much lower in number and density than proximities computed using other methods. Even so, there are still a considerable number of proximities (most of which are unlikely to result in separation violations) where the two aircraft were relatively close to each other in time and space (note that proximities were recorded and accumulated up to 200% required separation for these images).

The next series of pictures illustrates the results for all (Level 1, 2 and 3) proximities, weighted as follows. Level 1:2:3 proximities are assigned weights in 1:0.5:0.25 proportion, which

would reflect their share as a controller workload factor. Proximities are shown on a hexagonal grid with cells about 20 NM in diameter. Brighter shades of purple indicate higher proximity counts, the brightest indicating at least 32 proximities in a cell over the 24-hour period.

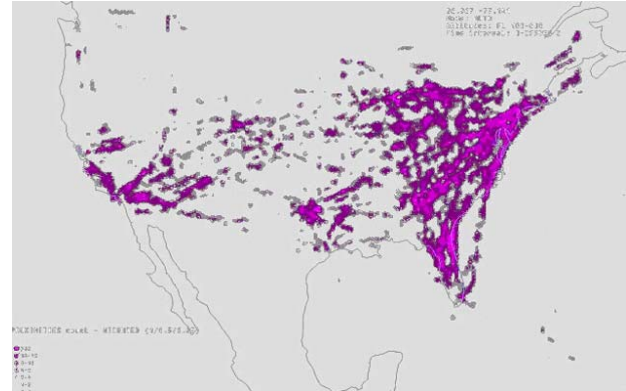


Figure 6. Proximities - actual tracks

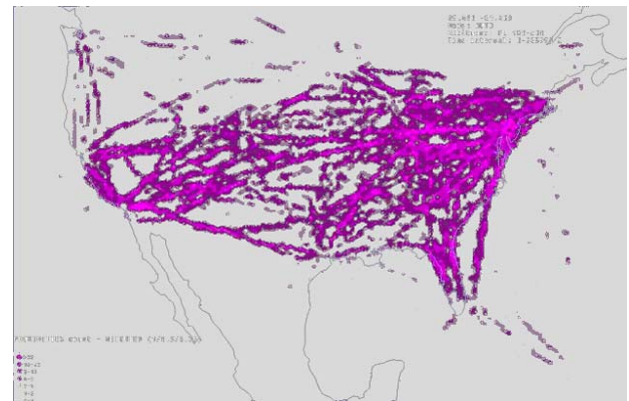


Figure 7. Proximities - planned tracks, NOT adjusted

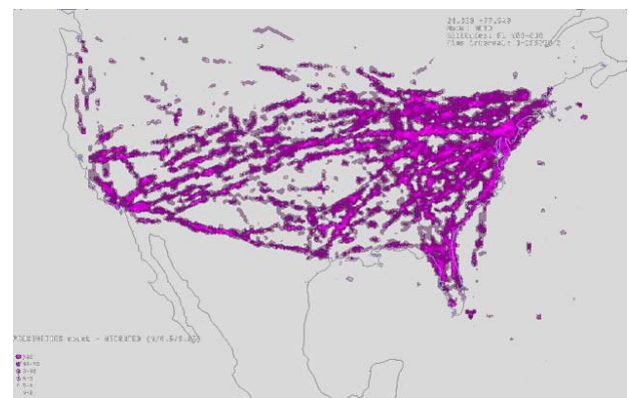


Figure 8. Proximities - planned tracks, ADJUSTED

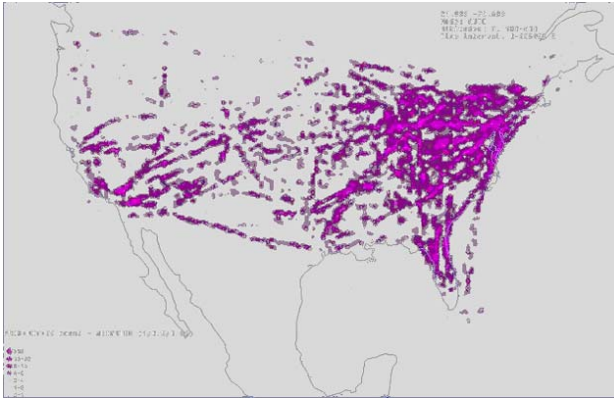


Figure 9. Proximities - TAAM

Direct Proximities Computation from ETMS vs. Processing TAAM-Reported Conflicts

Figures 8 and 9 show similarity between TAAM and direct-from-ETMS proximity reporting results but there are also some differences. We must, however, be cognizant of the specifics of our methodology, ETMS data collection, and TAAM conflict computation:

- TAAM uses continuous simulation at 6-second time step and applies the same time step for conflict probes. We use ETMS data with (currently) 1-minute position updates.
- In this TAAM scenario, only point airports were used. This allowed us to complete a full-NAS simulation (56,000 flights, full day) with conflict analysis in about 3 hours: a good result for a microscopic simulation tool. Flights departed at their scheduled departure time (STD), rather similar to non-adjusted ETMS-derived conflict computation.
- TAAM reports conflicts in two ways. There is the Proximity Monitoring report using 1-minute frequency (so that multiple proximity events can be reported for a pair of aircraft over some period of time). There is also the Conflict Count report where only the point-of-closest-proximity is counted *once* and reported for a given pair of aircraft. In this paper, we have used the TAAM Conflict Count report because only this report contains 4D information; but our own ETMS processing algorithm may contain multiple proximity reports for the some aircraft pairs.

The above differences and ways of accounting for them can be explored more in depth in the course of further research.

Comparison of Proximities Using Adjusted vs. Non-Adjusted Flight Plan Times

Overall, both related pictures (Figures 7 and 8, respectively) are similar but the underlying route structure is more visible in the non-adjusted case (Figure 7); that is, more proximities were identified than would have really been the case. Closer examination (Fig. 10 below) shows this clearer.

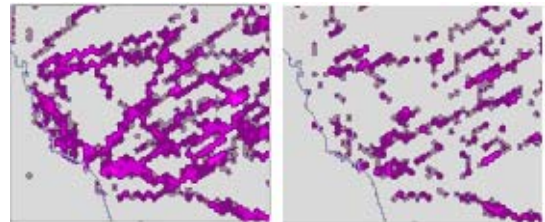


Figure 10. Proximities - non-adjusted vs. adjusted times

The differences between the two computation methods are therefore noticeable. Without the proper time adjustment (Fig. 10, left), more conflicts would be reported, and possibly in a different time and place than where they would have actually been likely to occur.

TAAM Climb/Descent Rates Calibration

Proximity visualization from our first experiments with TAAM data was very different from ETMS-derived data. TAAM showed many more conflicts in terminal airspace and much fewer conflicts in en-route airspace (see Figure 11).

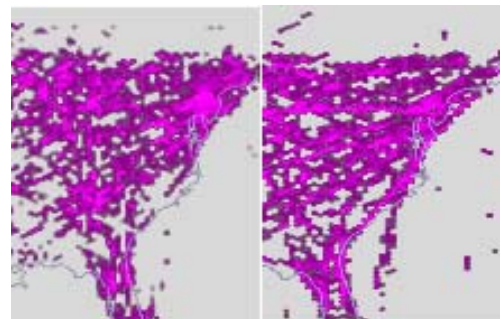


Figure 11. Initial TAAM (left) vs. ETMS (right) comparison

We then realized that aircraft climb and descent rates used in TAAM are for unrestricted climb and descent. If they are applied e.g. to aircraft climbing from a point airport representing a busy airport, the aircraft will reach its cruising altitude much sooner than it would in reality (with SID-required and ATC-imposed climb rate limitations). For instance, it is quite common for aircraft to level their climb for some minutes when transition from the Terminal area into En-Route center airspace. As a result, proximity reports from TAAM simulation using point airports and non-adjusted climb/descent rates are likely to show too many conflicts in terminal airspace near large airports.

This is not a drawback of TAAM as a model but merely a scenario setup issue. It highlights the need to either use runways, SIDs and en-route ATC restrictions (e.g. standard operational procedures) even in very-large-scale simulations, or at the very least, the need to adjust climb/descent rates if using point airports. We opted for the latter and reduced these rates by about 50-60%, depending on the aircraft type. This has resulted in a much-improved picture (see Figure 12):

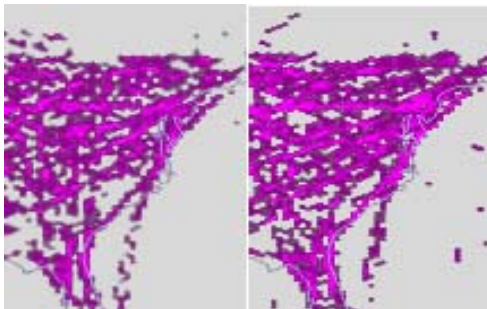


Figure 12. TAAM, adjusted, vs. ETMS (right) – US East Coast

Future Work

An acceptable way of increasing the accuracy of proximity computations is to interpolate between the 1-minute aircraft position updates. This can be done for both actual TZ hits and the position updates generated from planned tracks. For instance, we could increase the frequency of updates to once every 15 seconds and at the same time narrow the time window for 4D proximity comparisons. As higher definition (and frequency)

RADAR sites are added to the ETMS/ASDI data stream, proximity calculations will improve incrementally at the cost of processing times.

The same methodology could be applied to other, non-ETMS traffic data sources, such as radar tracks (at 4.7 sec frequency), and could be used in conjunction with airspace design and analysis tools such as SDAT, PASSUR, TARGETS and others. Post-operation position-by-position proximity checks could possibly be used in quality assurance applications if data fidelity is high enough.

To further increase the quality of proximities calculated from the ETMS data, flight plan re-filings and amendments should be considered alongside of initial filed flight plan records. The effect of using historical taxi-out times (e.g. from [11]) instead of TZ-record-derived corrections can be considered also.

Regarding TAAM proximity calculations, a definite improvement would result from the use of runway configurations within the simulation for the busiest US airports. TAAM currently uses SIDs without the knowledge of these runways, which minimizes travel distances in the terminal area. Both functionalities should improve the accuracy of TAAM proximity counts. Furthermore, the addition of operational restrictions from, for example, Letters of Agreements (LOAs) and Flight Level Allocation Schemes (FLAS) would allow TAAM to generate better profiles. All of these aspects may be incorporated into the TAAM model to produce a more accurate set of proximities at the cost of computational performance.

Conclusions

We have presented an efficient method of scanning ETMS or other timestamp-sorted air traffic archive files for computing potential proximities.

We have also developed a method to improve the accuracy of 4D flight profiles extracted from ETMS flight plans. This method applies times from actual aircraft position reports to the time component of the otherwise-unchanged filed 4D flight plans.

The comparison of aircraft proximity computations using the non-adjusted and time-

adjusted 4D profiles shows noticeable (though not dramatic) differences in results, which makes the use of such time adjustments justified.

Comparisons between the proximities computed using our method with those computed by a fast-time simulation model such as TAAM show that one must be cognizant of the different methods and different fidelity of such computations. Still, the resulting overall proximity distribution picture in the NAS is very similar for both methods.

Our research also highlights the importance of proper calibration of fast-time simulation models when trade-offs between simulation fidelity and the speed of computation are considered.

The methods developed in this paper could be applied to a variety of data sources and used in conjunction with many airspace design and analysis tools.

References

- [1] MITRE CAASD Airspace Management - Intersect Density Analysis Toolset (IDAT). At: www.caasd.org/work/project_details.cfm?item_id=134.
- [2] Donohue, G., A.Yousefi, K.Qureshi, 2003, Investigation of Enroute Metrics for Model Validation and Airspace Design Using TAAM, 5th Eurocontrol/FAA ATM R&D Seminar, Budapest, Hungary.
- [3] Wieland, F., D.Carnes, G.Schultz, 2001, Using Quad Trees for Parallelizing Conflict Detection in a Sequential Simulation, Proceedings of the 15th Workshop on Distributed and Parallel Simulation, Arrowhead, CA, p.117-123.
- [4] Crook, I., and D.Liang, 2003, FAA Operational Concept Validation – High-Altitude Airspace Analysis. 5th Eurocontrol/FAA ATM R&D Seminar, Budapest, Hungary.
- [5] Masalonis, A., M.Callahan, and C.Wanke, 2003, Dynamic Density And Complexity Metrics For Realtime Traffic Flow Management. 5th Eurocontrol/FAA ATM R&D Seminar, Budapest, Hungary.
- [6] Kopardekar, P., and S.Magyarits, 2003, Measurement and Prediction of Dynamic Density. 5th Eurocontrol/FAA ATM R&D Seminar, Budapest, Hungary.
- [7] Baart, D., 2001, *An Evaluation of Dynamic Density Metrics Using RAMS*. FAA Technical Center Report.
- [8] Weiss, W., S.Mondoloni, T.Thrasher, 2003, CSSI's NARIM Suite of Airspace Analysis Tools, a PowerPoint presentation by CSSI, Inc., April 2003, © CSSI, Inc. Internet link at: <http://ffp1.faa.gov/approach/media/modwork/CSSI.htm>
- [9] Sector Design & Analysis Tool (SDAT), 2004, Functional Description, FAA ATM Lab, at: www.faa.gov/ATS/ata/ata_lab/sdat_info.htm
- [10] Trani, A. et al, 1998, Development of Airspace Sector and Encounter Models to Support the Analysis of Aircraft Separation and Collision Risk, NEXTOR Research Report RR-98-16.
- [11] Bureau of Transportation Statistics Aviation Database. At: <http://www.transtats.bts.gov>

Acknowledgements

The authors are very grateful to Dr. George Donohue and Dr. Lance Sherry of GMU for their valuable comments.

Email Addresses

The authors can be contacted using the following e-mail addresses:

- Dr. Alexander Klein (aklein1@gmu.edu)
- Florian Hafner (hafnerf@erau.edu)

*24th Digital Avionics Systems Conference
October 30, 2005*