EVALUATION OF NAUTICAL MINUTE DISCRETISATION FOR CONTROL OF CONTINENTAL ENROUTE AIRSPACE

Paul N. Simon, Cees Bil, Lachlan Thompson
School of Aerospace, Mechanical and Manufacturing Engineering, Royal Melbourne Institute of Technology, GPO Box 2476V, Melbourne, Victoria, 3001, Australia

Keywords: *Continental, ATM, FEM, discrete areas, parallel processing.*

Abstract

An air traffic simulator that divides computational load, in both a physical and virtual sense, may lead to increased parallel processing effectiveness and the ability to use them in real time. This paper presents a potential simulator methodology with that capability, and attempts to justify the system's safety and accuracy.

1 Introduction

Air traffic management (ATM) and Air Traffic Control (ATC) simulators use one or two airspace calculation methodologies to primarily understand and manipulate air traffic; path mapping [1] & conflict detection [2], or aircraft clustering [3] & sectorisation [4]. The former follows the dynamics and limitations of aircraft in reference with other airspace entities, and is designed with such in mind. The latter follows the limits of airspace itself in view of other situational limitations like dynamic airspace capacity and controller workload. It is possible, and preferable for complete situational awareness, that both methods are employed in any ATM system program [5]. The inevitable limit that these programs encounter comes from the need for increased computing power to handle the systems' level of variable fidelity, and the need for increased computational processing speed to handle the large number of complex iterations that safely managing multi aircraft pathway interactions requires on an entire ATM system scale.

Nautical Minute Discretisation (NMD) and Continental Enroute Airspace (CEA) are simple concepts that are easily confused. NMD in its truest form refers to the division of airspace according to the minutes (one sixtieth of a degree) of longitude and latitude that define world surface coordinates. For the purposes of this paper, NMD also refers to algorithms and processes shown here that are reliant on such division. Physical variable discretisation is common in numerical and simulation studies, even in ATM improvement studies. Such optimizations are often discretized in terms of time recognized iterative methods to solve their goals. With increased maturity of finite element methods, attention is now also on discretizing three dimensional volumes, and using the properties gained from doing so to assist in hastening computational calculation [6].

CEA, in its truest from refers to the airspace through which all aircraft, for a continental sized Flight Information Region (FIR), fly the en-route portion of their journey. Examples currently include Australia, the US, and recently Europe. Control of CEA is currently established by isolated sectors usually defined by placement of radar ground stations or expected high traffic regions, usually both [7]. A relevant property of CEA is they allow, and could benefit from, cohesive control of all aircraft within their boundaries; aircraft on opposite sides of CEA could have an impact on the optimizations and control of each other as well as of all aircraft within that region. However such combined control is still a developing concept due largely to the difficulty of accurately predicting aircraft movement, and the obstacles associated with integrating such capability with current ATC operations. For these reasons, isolated sector ATC continues as the only acceptable form of control in CEA.

This paper discusses the potential of NMD methods in supporting combined control and planning of CEA. Among all previous work in ATM known to the author, surface area discretisation similar to those shown here have only been used to drastically simplify aircraft traffic analysis, however in comparing the needs of CEA control with NMD method potential, significant planning and management capability was recognized. Before showing this potential, an investigation into the issues of using NMD methods in ATM had to be performed. To do this, NMD method origins, actual NMD methods with computational comparison of their functional alternatives, and the supportive correlations NMD methods have with future processing capability, are presented here.

2 NMD Origins

The reasons for creating this methodology is three fold. The original reason for it was as a low fidelity ATM simulator designed to predict pilot path preference in high traffic, Free Flight, scenarios. As the simulations were expected to cover decades in virtual time, the ATM simulator required a high virtual-real time ratio. Additionally, a request was made that required an algorithm that, with satellite based communication, and on the assumption of no air traffic control, optimizes cooperative flight trajectories. As the simulator was designed for control of CEA operations, the simulator in its entirety had to be used to satisfy the request. The last motivation for this simulator was the desire for a more distributed means of predicting flight path collision avoidance. Iterative methods that seek an optimum solution are fine for a limited number of aircraft in a relatively small area. However, the permutations that are created when thousands of aircraft are being considered make such optimizations unfeasible in real time without the use of a supercomputer, which themselves are difficult to procure. Parallel processing supercomputers however are not as difficult and fairly easy to scale up with need. However retrofitting analytical methodology to suit parallel processing, while possible, forces concessions on the methodology. Deriving a simulator that can readily interface with current analytical CEA control methods, as well as efficiently utilize capability allowed parallel super computers, would ease acquisition of systems capable of controlling CEA.

2.1 Simulator Capability and Rationale

Pictures of the various levels of detail that the simulator manipulates are shown in figures 1 and 2. The question that paraphrases this simulator's methodology is; if it could be assumed that each square nautical minute (roughly a square nautical mile close to the equator) in a suitably large portion of airspace was treated algorithmically as an 'airport' with throughput time equivalent to how long it would take for any aircraft to pass through it, would the decrease in possible permutations and thus increase in iteration speed warrant the drop of fidelity that such discretisation implies? It is possible that the answer could be yes.

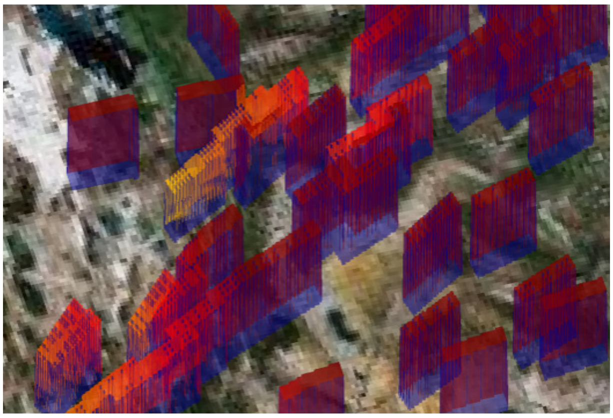

Fig. 1. NMD simulated US air traffic (2:56am UTC) showing aircraft entering squares at different heights.

Aircraft, due to their wake turbulence profile, have a volume of effect defined by their speed and the wake's rate of dispersion and dissipation. Assuming no relative wind, and general wake drop rates, this volume can be safely contained within an area thousand feet high and one nautical mile wide, with a length defined by aircraft speed. If you exclude the presence of the aircraft altogether, a nautical mile becomes the base unit for lateral separation. In essence, two aircraft cannot be allowed to exist within a single square nautical minute, within 1000 ft (reduced vertical separation minima), and within five minutes of each other, at any time, thus implying the lack of need of any other unit of lateral separation at a continental scale (this may not be the case for terminal airspace).

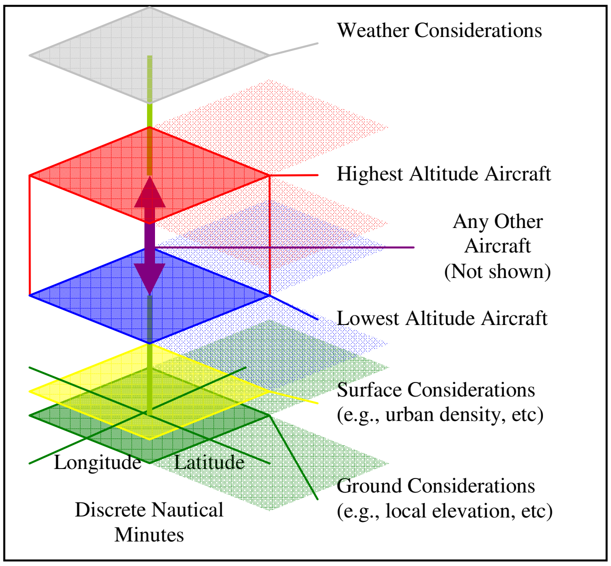

Fig. 2. Area Consideration Categories for data of each Nautical Square.

As for fidelity, constraining to a square nautical minute system does not prohibit optimization of a flight path within a sequence of square minutes, or of the sequence of square minutes itself. In fact, due to the distinction of external from internal forms of path optimization, each is easier to implement.

2.2 Impact of ATC procedural regulations

Most procedures used by the simulator are modeled after ATC procedures. While it may have been easier to model using stochastic models of optimized air traffic ideals, the more difficult obstacle lies in doing so for aircraft numbers experienced at a continental scale. The interaction complexities in those optimizations are usually no different than intersection calculation at an individual level using simple aviation formulary. As such they also would experience the exponential increase in calculation time required as imposed by needing to analyze all possible intersection permutations. However ATC have managed to do it without computational aid, so it was obvious that patterning air traffic handling based on discrete sectors could work around that exponential increase. Using square nautical minutes as minima for aircraft movement whilst replicating ATC procedures allowed the creation of the NMD methods shown here.

2.3 Air Traffic Differences due to Free Flight

In terms of air traffic, there is one fundamental difference between the two; the complexity of air route interaction. Currently, aircraft fly pre-defined routes to their destinations. This allows pre-defined intersection points which do not change in position under normal operating conditions. This allows sector control as sector borders can be drawn where high numbers of aircraft path intersection are unlikely to occur. ATC then becomes a matter of ensuring vertical separation in combination with one of lateral, longitudinal or crossing separation. Due to the pre-defined routes, safe separation in en-route regions requiring more than two of these types per set of separated aircraft would be rare.

Free Flight implies the ability to fly a direct route between an origin and destination. This means that pre-defined routes would not be adhered to, which means that intersection points could change and that simplification of aircraft separation work load can not occur easily. In terms of an individual sector it becomes likely that separation would require combinations of three or more separation types to handle the larger variety of aircraft intersections for a sector. For these reasons, Free Flight, especially for FIR where air traffic density is high, is likely to require computational assistance to facilitate FIR wide aircraft path planning to decrease the complexity that ATC handling an individual sector would have to face. Resolving conflicts as soon as possible would be more desirable then allowing them to be resolved within the sector where they could collide.

The simple complexity of current air route interaction is the reason behind why computation of intersection permutations or use of NMD methods for ATC is currently not common. The current level of complexity as enforced by having pre-defined routes does not need aid to be handled. However, as the NMD methods shown here were developed for situations of even greater complexity, it is hoped that using them in the current context reduces issues experienced from trying to apply an analytical variation of the same capability to current ATM systems.

3 NMD Methods

The following is a summary of the issues and potential as experienced while developing NMD methods. For comparisons sake, a functionally equivalent analytical method has been described to correspond with individual components of those NMD methods. In order to develop a functional simulation of en route air traffic and the airspace they define, three important functions need to be formed; recognition of aircraft flight path, conflict detection, and conflict resolution. The corresponding analytical and NMD methods, and the general theory and assumptions that support them have been described below.

3.1 Fundamental Theory and Assumptions

The fundamental theory, used by all methods shown here is standard aviation formulary; so calculation of great circle arcs (GCA, the most efficient two dimensional aircraft path, when cross winds are not an issue), true course, and position in terms of longitude and latitude. These are based on spherical triangle formula or ellipsoid parameterization and assume that the earth is spherical. As the area of concern is only continental sized, issues with variation in earth radius have not been given consideration. Also due to the parameterization, there is the usual fault of not being able to handle polar paths (i.e. paths that go directly over the geometric poles). While flights over poles are not common, paths elsewhere may, after ending, intersect with a pole, so the possibility is taken cared of.

Regarding the three functions a couple of procedural assumptions are made. First is that an aircraft's path is considered more static than any of its other variables; for two reasons. First is that there is common use of cross track deviation for the purposes of ensuring safe lateral distances [8]. Second is that of the dimensions that ATC can alter for an individual en-route aircraft, only its velocity and height are given priority for consideration; changing aircraft bearing and thus path, can cause other currently unknown intersections and the conflict resolutions they require. Additionally due to wake vortex dynamics, purely horizontal path alterations would only be more efficient when the time difference for aircraft crossing the same three dimensional point (longitude, latitude and height) is close to the wake vortex' time for dissipation (here assumed to be five minutes for any aircraft), thereby allowing a small increase in aircraft cross track position to 'sidestep' the non dissipated wake vortex. In any other situation, the large lateral distance to be covered to avoid conflict would be less efficient than a change in either height or speed to facilitate the same.

The next assumption is that multiple aircraft on the same GCA path, rather than being considered as multiple individual aircraft with different paths, are treated as having the same path. For intersection determination this allows an intersection between two particular routes to be calculated once, and then reused whenever aircraft, that are present on the routes that define that intersection, have to check for conflict. Overtake and reverse direction possibilities can be handled by assigning different cross track positions (e.g. really fast traffic between 0 and +6NM, slow traffic between +6NM and +12NM, reverse flow traffic would be the negative amounts of these), and a middle 'no movement' cross track region can be established. In other words, all these actions can be defined using the same path. The methodology can handle multiple aircraft on the same route as multiple aircraft on different routes; however this reduction of possible paths, and therefore complexity, is possible now and even in free flight theory, so there should be no issue of using it.

3.2 Computational Assumptions, Limits, and Issues with Comparison.

The simulator was developed using MatLab; a numerical computing environment that uses a C based programming language. One of its recognized areas of capability is in indexing and matrix manipulation. Thus, wherever variables would benefit from having been discretized, as usually required for numerical analysis, MatLab would confer optimizations to computation time unavailable to other computing environments. With regards to the work done here, where a processing method allows either an analytical or index method of handling, the index method is always chosen, so as to benefit from the efficiencies from MatLab. Obviously, as analytical methods, by definition, do not use discretized data, their indexing potential is limited to data that is already discrete in nature.

While powerful, MatLab adheres to the limits of the system it's on, thus issues that would affect any program's performance also effect applications within the MatLab environment; processing speed and memory limits do affect application speed. This inference of an application's use of either processing speed or memory allows the justifications made in this paper. Additionally, as MatLab was developed to take advantage of parallel processing capability, inferences to the potential of NMD methods when parallel processing can also be made.

3.3 Flight Path Recognition

As previously mentioned, an aircraft's two-dimensional path is considered more static than any of its other variables. Additionally, it's this characteristic that determines where a collision will occur, if at all. Thus the first step for continental scale conflict detection and resolution is to plot on a map where that aircraft has to travel, and then determine whether any aircraft on those paths can collide given the start and end point of those paths.

For the analytical form of GCA calculation, creating the path simply requires a two dimensional start and end point, and an expected amount of cross track deviation. That sufficiently covers the area an aircraft could travel over: a point inside this area would have a total along track distance from it to start and from it to end equal to the GCA distance from start to end, and a cross track distance less than the deviation specified.

In order to determine intersections, their existence must be proven. As any two GCA would automatically have two intersections (a.k.a. antipodal points), this existence is determined by whether or not the intersection exists between both paths' start and end points. Further, as there's no other conclusive way to determine intersections analytically, it means that consideration of each combination of paths must be performed. In other words, the check for intersections must be performed for ${}_NC_2$ possibilities where N is the number of aircraft paths. Additionally, as aircraft intersections can have considerable area, each of these intersections must also undergo a similar check to determine whether or not they intersect or overlap; the process for determining intersection overlap continues until all possible intersections are eliminated by virtue of the placement of their start and end points. The total number of intersections to be computed, assuming no terminations of possibilities before full consideration all possibilities, would be:

$$Total\ Intersections = \sum_{1}^{N-1} N_i\ ,\ where \qquad (1)$$

$$N_i = {}_{N_{i-1}}C_2\ ,\ for\ i=1 \ldots N-1$$

$$N_0 = Total\ Number\ of\ Aircraft\ Paths.$$

(1) defines the escalating issue in simple permutations. The only apparent way of mitigating this issue is to prematurely (i.e. using non-path dependant data before consideration of any $i+1$ intersections) determine if an intersection exists. However, doing so eliminates possible comparison with NMD and is therefore not done here.

Defining a path for an aircraft for a NMD grid requires that the discrete nautical minute squares that fall within (even just partially) a path's two dimensional area are all tagged as being so. What information is placed in each square is a matter for later consideration (sections 3.4-5); the key requirement for defining the path is how the squares that represent that path are found. The analytical method for determining whether or not a point exists inside an aircraft's two-dimensional path can be applied here. However this is computationally intensive by itself; each component calculation (along track distance to start, along track distance to end, and cross track distance) is repeated for each of the nautical minute squares that exist inside the area. For an area of continental size, this number could be in the millions, and this is only one path. While the number of calculations to facilitate this for all aircraft would be smaller than then the amount in (1), this number is still substantial and required reduction.

After going through various different methods, all of which had issues in terms of using too much memory, having holes in the path, or having inappropriate ends, an appropriate method was defined. This method showed itself to be the fastest whilst satisfying all known criteria. It uses the steps below and can be visually defined using the corresponding parts of Figs. 3-7 (appearance of non linearity caused by earth curvature warping to two-dimensional representation):

1. Create a numerical matrix just big enough to cover the path. Use the maximum and minimum values of the path's longitude and latitude (with an appropriate buffer) to create a discrete rectangular area of the same longitudinal width and latitudinal height.
2. Place all discrete longitude values between the start and end point (including them, and some outside of them to form a buffer) and place them in a vector array.
3. For each element of this array, determine the corresponding latitude (unrounded) and aircraft bearing at that latitude, using the start point and its bearing to the end point.
4. For each bearing, determine the latitudinal distance required to cross a path of that bearing [9], then double it to ensure all discrete areas within that longitude are covered.

$$Latitudinal\ Dist = \frac{2s_y}{\sin(\Delta\theta)},\ where \qquad (2)$$

s_y=*allowable cross track deviation*

$\Delta\theta$ =*Bearing difference between path and Meridian*

While (2) is a flat surface approximation, this distance is acceptable as it is defined along a meridian and thus does not suffer the spherical warp to coordinate position that distances in other directions would incur.

5. Centre this distance on the corresponding latitude for each bearing to determine the maximum and minimum latitude experienced for each discrete longitude.
6. For each discrete longitude, place the cell index of all cells between and on the maximum and minimum latitude for that longitude into a cell list.
7. For the start and end discrete longitudes, check all cells within a distance (half of the latitudinal height, for that longitude), from its calculated latitude, for distance from the opposite end point.
8. Discrete areas in the vector array near the end point, with a distance greater than the path's total length (plus appropriate buffer) are removed from the cell list.
9. Discrete areas not already in the vector array, but have distance to the opposite point less than the total path distance, are then added to the cell list.

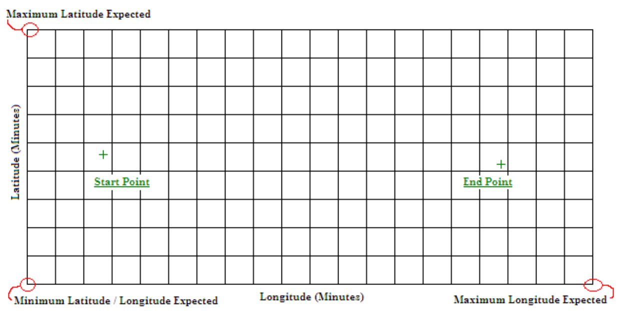

Fig. 3. Step 1 - Determination of Grid Size

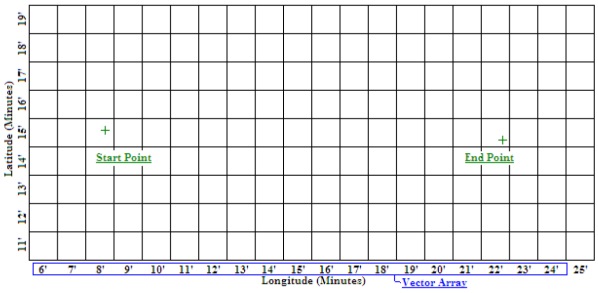

Fig. 4. Step 2 – Collecting Longitudes

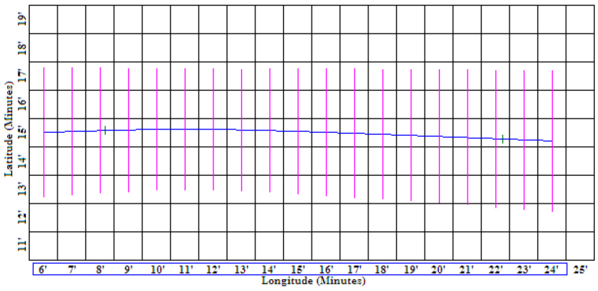

Fig. 5. Step 3-5 – Determining Relevant Discrete Areas by Latitude

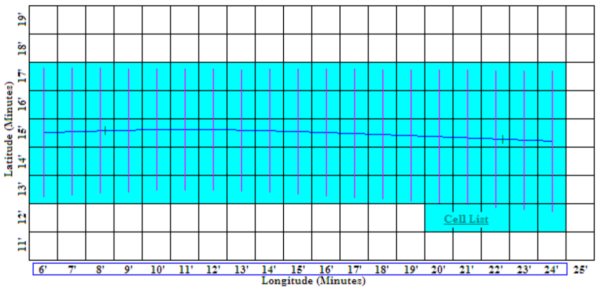

Fig. 6. Step 6 – Collecting Relevant Discrete Areas

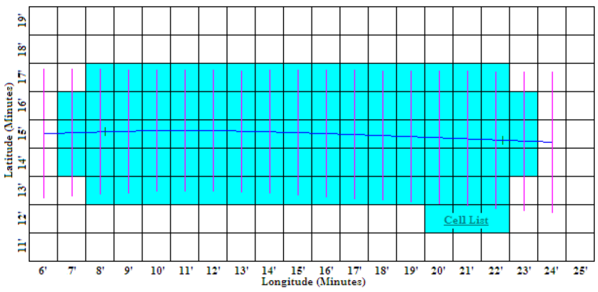

Fig. 7. Step 7-9 – Correcting for start and end points.

While not shown here, all buffers have analytical equations to determine their size to ensure that all relevant cells are included in the vector array.

Several differences between this method and the straight forward application of the analytical method make the former significantly faster. First and foremost, the only areas that require a position query are cells close to the start and end points; everything else is straight application of indexing capability. Aviation formulary is used to determine relevant GCA path cell positions, and the small circle assumption allows considerable savings wherever it's applied due to a lack of need to recognize earth curvature. The above improvements thus infer two notable efficiencies. First is that computation time is, as desired, proportional to the area considered; this implies that smaller paths can be created faster which is important when considered against the computation/surveillance time horizon issues experienced by ATC today [10].

Second, and going from the first efficiency is that total computation time is now more proportional to the longitudinal distance between the start and end points. This is not always true, but provided the longitudinal values for cells near the endpoints do not overlap, it would be. This implies that increases to computation time due to having a larger allowable cross track deviation, would incur smaller increases than what would be incurred by increasing a path's length. This is particularly important when considering sources of unpredictable cross track deviation; the greater the variation, the more allowable cross track deviation needed to cater for possible intersections.

There are path types where computation times are less affected by the efficiencies gained from using an acceptable small circle assumption: near polar paths. Of the steps outlined previously, most of the apparently pointless actions were created to handle this scenario. While it is possible to simplify, and possibly optimize, the procedure to handle these kinds of paths, doing so would imply different methods to handle different paths. This would increase system complexity and there would be no easy way of gauging whether or not this increase would cause a decrease in computation time, thus this issue has been avoided for now.

3.4 Conflict Detection

This method was taken directly out of ATC procedure guidelines for manual determination of conflicts [9]; remember however that as the only true consideration for conflict is of wake field dissipation, those procedures can be simplified even further. Simply put; aircraft that pass the same three-dimensional intersection area within five minutes of each other are in conflict. For pure analytical, intersections are already defined; determination of conflict is performed on all intersections of a particular order before consideration of a higher order, this ensures that higher order intersections are disallowed from consideration if a lower order intersection that defines it does not exist.

For NMD, each discrete area has the potential to be an intersection; this has two important implications. First, the total number of intersections to undergo consideration is always equivalent to the number of discrete areas in the entire NMD grid, however as an intersection requires an aircraft presence greater than one, and that this presence is in itself an integer, defining which areas require determination of conflict is purely a matter of indexing and is trivial in terms of computation time; computation of conflict detection would therefore be proportional to the number of intersections as defined by the analytical.

Second, comparative order for NMD would be defined by the number of aircraft that pass through a discrete area; the *N-1* order limit that applied to the analytical also applies here, though as a single aircraft presence in an area is itself one, the maximum number that would be seen would be *N*. Contrary to analytical, consideration of conflict detection does not need to follow this order in any way. This is due to the discrete nature of the individual areas; nearby cells though derived from the same aircraft do not share the same conflict information and thus can not effect computation for nearby cells for the purposes of conflict detection. While this would imply lesser computation time per intersection, a requirement for conflict resolution does define an order for consideration thus denying NMD this possible improvement.

Detection of conflict for either an analytical or NMD intersection requires that, for each intersection, a time for arrival at the intersection point be determined from each aircraft's speed profiles for their flight; each of these would then be checked for difference to ensure that it was less than 5 minutes plus a bit more time to cover the size of the intersection. For a straight analytical approach, the additional time would be defined by where the intersection was entered, which would be a value already obtained. For NMD, the additional time would consistently be based, no matter the difference in bearing, on the maximum length of a cell's diagonal (distance from cell centre to corner); this is acceptable as these distances would be smaller than the minimum unit of consideration in NMD and thus would be arguably negligible, the maximum diagonal is therefore chosen as an appropriate buffer and not for the purposes of accuracy. For both NMD and analytical methods, if an intersection shows that a conflict does exist, it would get flagged for conflict resolution.

Overall a comparison of analytical and NMD would point to the greater number of intersections as required by analytical ((1) would be greater than the number of discrete areas in a NMD grid) to indicate that computation of all analytical intersections would take longer than computation of a NMD grid. Should (1) be less than the number of discrete areas, or if intersection areas are larger than the continental area (algorithmically possible, especially as the NMD grid decreases in size), this statement may not remain true. Otherwise, even with the potential to minimize the effective number of possible intersection for analytical methods, unless that minimization limits the number of intersections to less than the number of NMD discrete areas, conflict detection using NMD could still be faster.

3.5 Conflict Resolution

The procedure for conflict resolution for both analytical and NMD is the same; going in decreasing order of conflict, resolve the conflict as desired for each intersection, ensuring to leave a marker on the aircraft involved indicating where the conflict was resolved (to prevent alteration due to lesser order conflicts). The method of resolution can alter any number of variables; speed and height are the most frequently seen alterations, however it is possible to use acceleration and cross track deviation to ensure separation. Of these only cross track alterations could change how an analytical or NMD method would work, as the rest define time for arrival for an intersection where cross track deviation does not. As for individual conflict resolution, the method of resolution is irrelevant so long as the method used results in consistently separated aircraft within the intersection. This implies that NMD, like analytical methods, can work with conflict resolution procedures of any that could be directly applied now. Obviously this would be on the assumption that alterations to aircraft do not veer significantly from their intended path in such a resolution; but this is true for both analytical and NMD so further development of either framework would need to occur if such resolutions methods are desired.

After resolution of each conflict, conflict detection must be performed again so that conflicts that would have been solved by higher order conflicts are removed from consideration. Additionally, if a recent resolution causes an unsolvable conflict (due to the locked in nature of higher order conflicts on involved aircraft), an alternative resolution method may be used to generate a solvable conflict. Unfortunately this functionality was tested using only resolutions that altered only height and speed; resolution methods with more complex procedures may not experience the same capability.

Overall, because of the need to compute conflict detection once per resolution, computation time for conflict resolution for all aircraft would be highly dependant on conflict detection computation time; something for which NMD methods already looks promising. On the possibility that the number of intersections for analytical and NMD are of the same magnitude, i.e. that computation time for conflict detection is the same, there is a good possibility that NMD methods would be better still, as NMD intersections are a greater than one multiplier on physical intersections (the additional intersections experienced by analytical being intersection overlap). Resolving one NMD intersection has the potential to resolve nearby intersections with the same aircraft, so even with similar numbers of intersections, the number of resolutions that would be performed would be significantly less.

4 NMD Implications for Parallel Processing

While it is possible for the analytical method described to be coded to work on parallel processing framework, NMD does lend itself more readily. In consideration of the flight path, the analytical method, by virtue of the combinations to be developed does need to be centralized first; it could be divided, but the determinant for doing so would require some clarification. The NMD method is not as centralized; if the grid properties are known to everyone, computation could occur at the point of origin, then distributed to the discrete cells individually (a grid representation of an aircraft's path is no different to code utilized in image storage). The consideration of conflict detection is area based, but the same can be said for its analytical counter part; no further advantage for NMD methods there. Centralized processing is required for CEA level conflict resolution for both methods; so the state of distribution to and from the central processor and other entities is important. The more that is already distributed in terms of path planning, the easier centralized control becomes, as the amount that needs to be distributed decreases. This means that benefit as allowed by the greater distributable form of NMD, in consideration of path planning, is further increased in due to lesser effort needed for centralized control. Obviously, as an analytical central processor has to decide far more in terms of which processor determines what, its processing capability is comparatively diminished.

5 Conclusions

The method presented in this paper was developed in order to allow cohesive control of CEA traffic under free flight assumptions. The method is easily distributable and works readily with parallel processing capability and therefore with future cost effective supercomputing potential. The selection of discrete areas that define an aircraft path is purely analytical and thus incapable of causing unpredictable variation. Additionally as this selection process was derived from standard aviation formulary, it can receive input from other methods of aircraft position prediction and conflict detection, which uses the same theory, to define more complex interactions. With additional testing using currently available supercomputing parallel processing capability, NMD methods could well become the step between current ATM methodology and future free flight capability.

References

[1] Xue M, and Atkins EM. Three-Dimensional Segmented Trajectory Optimization for Runway-Independent Aircraft. *43rd AIAA Aerospace Sciences Meeting and Exhibit*, USA, AIAA-2005-909, 2005.

[2] Chaugule V, and Pant RS. Investigation of Strategies for Optimal Collaborative Airborne Separation in Free Flight, *7th AIAA Aviation, Technology, Integration, and Operations Conference (ATIO) Proceedings*, Northern Ireland, AIAA-2007-7730, 2007.

[3] Bilimoria KD and Jastrzebski M. Aircraft Clustering Based on Airspace Complexity, *7th AIAA Aviation, Technology, Integration, and Operations Conference (ATIO) Proceedings*, Northern Ireland, AIAA-2007-7728, 2007

[4] Tandale MD, and Menon PK. Genetic Algorithm Based Ground Delay Program Computations for Sector Density Control, *AIAA Guidance, Navigation, and Control Conference Proceedings*, South Carolina, AIAA-2007-6358, 2007.

[5] Alam S, Abbass HA, and Barlow M. ATOMS: Air Traffic Operations and Management Simulator, *IEEE Transactions on Intelligent Transportation Systems*, location, Vol. 9, Issue. 2, pp 209-225, June, 2008.

[6] Massink M, and DeFrancesco N. Modelling Free Flight with Collision Avoidance. *Proceedings for the Seventh IEEE International Conference on Engineering of Complex Computer Systems*, pp 270-279, 2001.

[7] Federal Aviation Administration, *NextGen Implementation Plan*, U.S. Department of Transportation, Federal Aviation Administration, Washington, 2008.

[8] International Civil Aviation Organization. *European Region Area Navigation Guidance Material*. 5th Edition, European and North Atlantic Officer of ICAO, 2003.

[9] International Civil Aviation Organization. *Procedures for Air Navigation Services in Air Traffic Management*, 15th Edition, International Civil Aviation Organization, 2007.

[10] Lope, J, Vilaplana M, Bayraktutar I, Klooster J, Asensio JM, McDonald G, Kappertz P. Towards an open test bed for the study of trajectory synchronization in the future ATM system: The ASIS initiative. *Integrated Communications, Navigation and Surveillance Conference*, ICNS '09, pp 1 - 14, 2009

Contact Author Email Address

paul.simon@rmit.edu.au

Copyright Statement