FMS AUTOMATION ISSUES FOR FUTURE ATM INTEGRATION

George W. Flathers III*/, D.J. Allerton*, and Graham T. Spence*****
***University of Sheffield, **Air Force Institute of Technology,**
*****Aerosoft Ltd.**

Keywords: *Future ATM, UAS Integration, FMS, Automation*

Abstract

A key issue in attempts to increase the level of automation in future Air Traffic Management (ATM) systems is cooperation between flight management systems (FMS) and conflict detection and resolution (CD&R) systems. Trajectory prediction is integral to the process of air traffic conflict detection and resolution, and errors can be introduced if actual aircraft navigation performance deviates from these predictions.

This paper describes two algorithms that could potentially automate some manual flight management operations and are applicable to conflict resolution and Flight Management System navigation algorithms. Example operation of the algorithms are explored using simulations of flight plan rejoining, and of merging ATM generated flight plan modifications with an active flight plan. The results highlight the need for agreed navigation behaviour if future autonomous airspace management systems are to become a reality.

1 Introduction

1.1 ATM Modernization Expectations

In response to the projected growth of air traffic in the next decade and beyond, major Air Traffic Management modernization efforts have been instituted in Europe with the SESAR Programme and in the United States with the NextGen project. The goal of these programs is to increase the air traffic capacity while simultaneously improving safety and efficiency [1,2].

To accomplish this, both SESAR and NextGen envision a future ATM system with:

- Better system-wide information exchange, including the use of datalink as the primary mode of communication;
- More precise management of air traffic through trajectory based operations (TBO), including trajectory negotiation between aircraft and Air Navigation Service Providers (ANSP) and wide implementation of Area Navigation (RNAV) and Required Navigation Performance (RNP);
- An increased reliance on automation, including automated support of conflict detection, monitoring, and resolution.

The role of Flight Management Systems is to help reduce pilot workload and improve navigational performance and flight efficiency by managing many aspects of flight planning, navigation, and guidance [3]. With increasing automation, digital data exchange, and navigation performance requirements, it is likely that FMS will require a higher level of integration with ATM systems [1]. For example, we may see semi or fully automated conflict resolution systems generating and issuing short-term trajectory modifications (possibly 4D) to solve potential loss of safe aircraft separation events. Additionally, we may see non-segregated airspace shared with remotely-piloted and autonomous Unmanned Aerial Systems (UAS).

However, implementing this vision for increased reliance on automation is complex, requiring the close cooperation between ground automation tool and avionics developers. This cooperation will have to occur from the lowest to the highest levels of implementation so consistent and predictable navigation behaviour can be realised. Such consistency will be a key enabler to the success of future automated conflict resolution tools that rely on trajectory predictions.

1.2 UAS-ATM Integration Expectations

The widespread use of UAS, integrated with ATM systems, in non-segregated airspace has yet to be achieved. In this paper, we define UAS-ATM integration as the seamless inclusion of UAS in non-segregated airspace, supported by the necessary ATM, communication and avionics technologies to ensure safe separation of all airspace users. This definition does not cover UAS designed to avoid all other aircraft without the aid of ATM services. Some of the factors currently challenging UAS-ATM integration [4] include: equivalent level of safety as manned aircraft; certification of UAS avionics; ATC operational procedures when handling UAS; ATC-UAS FMS autonomous cooperation; semi or fully automated CD&R (to prevent controller workload increase); UAS-ATM datalink agreement, reliability and adoption.

If the expectation of UAS-ATM integration is to be realised [5,6], the onus is on UAS and avionics manufactures to converge with the ATM systems of the future. Currently, a wide variety of FMS/autopilot systems are being used for UAS navigation, ranging from simple direct-to navigation systems to full 4D trajectory management computers and adaptations of commercial transport aircraft FMS. It is likely that these systems will produce a large variety of navigational guidance responses to flight plan modifications uplinked by future CD&R systems. Before UAS can be safely integrated into non-segregated airspace managed by a semi or fully automated CD&R systems, the navigation systems and algorithms must produce predictable responses to ATM commands. The predictability of such responses is required to reduce the errors in trajectory prediction calculations so safe aircraft separation can be maintained. Autonomous systems (ground and airborne) will benefit greatly from agreed FMS navigation behaviour.

1.3 Paper Organisation

This paper focuses on some automation implications for flight management systems and automated aircraft separation management systems – particularly with relevance to autonomous UAS integration into the airspace, although equally applicable to commercial air traffic. Section 2 outlines the required FMS capabilities to enable the realisation of future ATM systems. Section 3 describes two algorithms aimed at providing automated solutions to processes that are typically completed manually. Finally, in Section 4, simulations are used to highlight the operation of each algorithm and the potential they offer for automation.

2 FMS Capabilities to Realise the Future ATM System

The TBO concept enables optimized flight routing by allowing aircraft operators to negotiate with the ANSPs a desired 4D trajectory, unconstrained the location of ground-based navigation aids. In return, aircraft will be required to navigate along the contracted trajectory with a given accuracy and precision. This will allow optimal routing, resulting in fuel savings, reduced emissions and flight time, while also enabling more accurate trajectory prediction by automated ATM tools.

For TBO to be realised, the FMS must be capable of 4D trajectory management, including the ability to define, communicate, predict, evaluate, modify, and execute 4D trajectories. Specifically, the FMS must be able to fully define 4D waypoints as well as other parameters such as the nominal bank angle, turn initiation point, and control mode; and it must be coupled with a digital data link so it can exchange these trajectories with other airborne and ground systems. The FMS must also be able to model and predict the future trajectory with a given accuracy, and to evaluate the safety, feasibility and efficiency of any proposed trajectory modifications. If acceptable, these trajectory modifications will need to be merged into the currently contracted trajectory in a consistent, predictable way. Additionally, the FMS must be able to execute these trajectories with consistent, predictable flight guidance including the ability to handle unpredicted scenarios such as rejoining the flight path following an unplanned deviation.

If autonomous UAS are to seamlessly integrate into future ATM systems, the role of the FMS would critically expand to managing all aspects of navigation and flight guidance, including automatic responses to clearances and instructions issued by the ANSP and to safety commands issued by on-board safety systems such as TCAS or sense-and-avoid systems [7]. Even for remotely piloted aircraft, safety considerations may still require the same level of FMS functionality as for autonomous aircraft in the event of failure, error, or unacceptably high latency in the command and control datalink.

The concept of operations for future airspace systems will necessitate significant increments in functionality than is currently available. The high level capabilities described in the previous paragraphs will rely on standardised low-level capabilities within an FMS and ATM automation tools. In this paper we highlight two particular important aspects of datalinked ATM-FMS automation (predictable, fully automatic flight plan rejoining and automatic in-flight flight plan route modification merging) that are currently not widespread, but require consideration amongst the aerospace community to ensure future automation systems collaborate efficiently and minimise unnecessary trajectory prediction errors.

The need for these particular automated capabilities became evident during the development of a fast-time airspace simulator at the University of Sheffield, and during flight trials of prototype automated airspace management systems being investigated as part of the Smart Skies project [8,9]. Smart Skies is a flight test program exploring future technologies that support the safe and efficient utilization of shared airspace by both manned and unmanned aircraft.

3 Description of Further Automated Capabilities

Within this section two algorithms are described that provide the flight plan related FMS-ATM automated functionality briefly mentioned in Section 2. The first algorithm enables either ground-based automation tools or an on-board FMS to be able to automatically determine an aircraft's active leg in a flight plan; in other words, determining a suitable leg in the event the aircraft is lost with respect to the flight plan. The second algorithm enables either ground-based automation tools or an on-board FMS to automatically merge new flight plan segments into the active flight plan, as may be the case when merging a commanded conflict resolution manoeuvre into the flight plan.

The authors acknowledge that alternative techniques to automating these manual processes are likely to have been developed over the course of other airspace automation and UAS integration research projects. However there is little published literature and no generally accepted solution to these automation problems. The authors' intention is that this paper will stimulate discussion on these and similar algorithmic issues.

3.1 Motivation

An automated flight plan segment discovery algorithm is useful in many circumstances, including:

- Ground automation tools that have received a flight plan and aircraft surveillance data, but lack correct knowledge of the active leg.
- Ground or airborne conflict resolution algorithms that need to determine a suitable leg to rejoin the flight plan after completing a conflict resolution manoeuvre. Future automated systems may require ATC clearances to rejoin a flight plan at a known fix on the current flight plan. This is a potential constraint on conflict resolution algorithm design, which an automated flight plan segment discovery algorithm may alleviate.

- If an aircraft temporarily suspends the automatic sequencing of waypoints to execute a route deviation due to weather, or ATC vector, and subsequently needs to rejoin the original flight plan.

Most of the airborne circumstances can be solved by a pilot manually sequencing the waypoints until a suitable leg is found. Manually sequencing waypoints is common practice in the event of flight plan discontinuities. However, human intervention increases pilot workload. Furthermore, in the context of ground-based conflict resolution tools, a controller may incorrectly predict the active leg chosen by a pilot to rejoin the flight plan. Algorithms could perform this function and offer a standard solution for consistent automatic flight plan segment discovery by ground automation systems and the aircraft FMS. In the context of a future airspace with integrated UAS, manual waypoint sequencing either increases the workload of a ground station operator or remote pilot. Additionally, to account for the possibility of temporary loss of datalink communication with ATC (for both remotely-piloted and autonomous UAS), automatic segment discovery of a suitable active leg is highly desirable, not only from a navigation perspective, but also from a predictability perspective.

Similar reasons can be given for the usefulness of an automated flight plan merging algorithm. For example, flight plans for UAS missions may involve many waypoints, and as a result it would be an inefficient use of bandwidth to transmit an entire modified flight plan when only several waypoints are affected. The problem of merging flight plans is simplified if the final waypoint of the route modification is along the original route of flight. However, this forms a constraint on the generation of conflict resolution solutions. Furthermore, not all conflict resolvers are designed to incorporate this principle, and as a result will require the resolution manoeuvre to be merged into the original flight plan in an intelligent way.

3.2 Automatic Segment Discovery Algorithm

This automatic flight plan segment discovery algorithm is based on an adaptation of a straightforward two-dimensional *point–to-segment* distance computation. When applied to flight plan segment discovery, the point is represented by the coordinates of the aircraft and segments are represented by the individual legs in a flight plan. Each segment is constructed from starting and ending waypoints as shown in Figure 1.

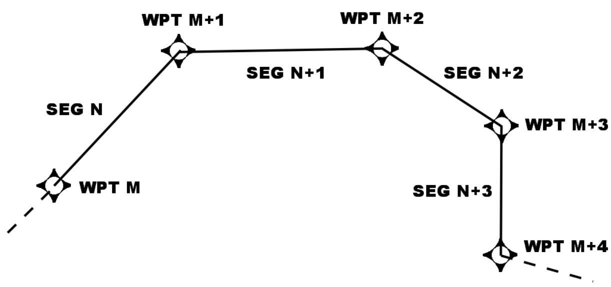

Fig. 1. Flight plan broken into segments for the automatic segment discovery algorithm.

Implementation of a *point-to-segment* distance calculation can be found in computational geometry reference books [10]. The distance algorithm is used not only to return the distance from a segment, but also to determine whether the point (aircraft) is behind, within, or in front of a particular segment bounds. This information is available because such geometrical calculations are typically based on projecting a point onto an infinite line defined by the starting and ending points of the segment. The projected point can either lie behind the segment (aircraft not yet reached the segment), within the segment, or passed the segment. The following equations use vector notation to illustrate how to determine where the aircraft is relative to the segment boundaries.

$$\bar{t} = \left((\boldsymbol{P}_{end} - \boldsymbol{P}_{start}) \bullet (\boldsymbol{P}_{aircraft} - \boldsymbol{P}_{start}) \right) \quad (1)$$

Where the value of $\bar{t}$ represents the projected aircraft location on the segment defined by the starting and ending waypoints. If $\bar{t} < 0$, then the aircraft is *before* the segment. If $\bar{t} > \sqrt{((\mathbf{P}_{end} - \mathbf{P}_{start}) \bullet (\mathbf{P}_{end} - \mathbf{P}_{start}))}$, the length of the segment interval, then the aircraft is beyond the segment.

Otherwise, the aircraft is considered within the segment boundaries. In this segment discovery algorithm, only segments that the aircraft is potentially within, or has yet to reach, are of interest.

Figure 2 visualises the segment boundaries defined in the flight plan shown in Figure 1. Note that the boundaries of each segment are identified by perpendicular lines emanating from the waypoints defining a particular segment. These lines extend infinitely as shown by the dot-dash lines. This results in an aircraft potentially existing in multiple segments simultaneously. Furthermore, note that the solid triangles show the regions in which an aircraft is not within any segment. In such an event, we use the *before-within-passed* information and assume that if an aircraft has passed a segment, and before the next segment, then we allow the algorithm to consider this next segment as within the next segment bounds and a potential active leg. The pseudocode for the segment discovery algorithm is presented in Figure 3.

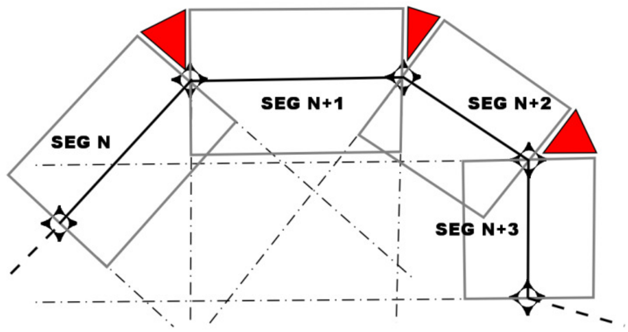

Fig. 2. Flight plan segment boundaries, extended infinitely (dotted lines).

Initially, the search for a potential rejoin leg begins with the last known active leg. The remaining flight plan legs are then considered sequentially by testing the cross track error (XTE) of the aircrafts current position with the segment under consideration. If the XTE is less than a previously defined minimum (MIN_XTE), then the *best leg estimate* is updated to the considered leg and MIN_XTE updated. Once all of the flight plan legs have been considered, the *best leg estimate* is returned. The search can be stopped early without considering the remaining flight plan by exiting the loop if the sum of the lengths of the legs so far considered is less than a distance threshold (for example, double the distance of the aircraft to the starting waypoint of the last known active leg).

```
1. active_leg_estimate  = current_active_leg
2. best_leg_estimate = current_active_leg
3. LOOP over remaining legs in flight plan
        Calc XTE from active_leg_estimate
        IF aircraft within segment bounds
                IF XTE < MIN_XTE
                        MIN_XTE = XTE
                        best_leg_estimate=↵
                        active_leg_estimate
        Increment active_leg_estimate
4. Return best_leg_estimate
```

Fig. 3. Pseudocode outlining the segment discovery algorithm.

Previous studies [11] have reported typical commercial flight plans include an average of 6.2 turns per flight during the portions of flight greater than 10,000 ft, with the majority of turns being 20 degrees or less. However, UAS missions may require more complex flight plans, such as those incorporating orbits, intersecting and collinear segments. This algorithm is successful for both simple and complex flight plans, and as a result can be applied to both commercial air transport and UAS flights.

3.3 Automatic Flight Plan Merging Algorithm

A method has been developed to merge the lateral components of two flight plans based on the principle of wayline leg sequencing. It should be noted that this algorithm has been designed assuming track to fix legs and fly-by turns, which are expected to be the most common path-terminators in future TBO airspace since they result in a fully defined ground track.

A flight plan leg terminating at a fly-by waypoint can be sequenced when the aircraft crosses an infinite-length wayline at the bisector of current leg and next leg [3 Sptizer]. Although variations exist (such as waylines perpendicular to segment at the turn initiation point), the authors consider this the most general.

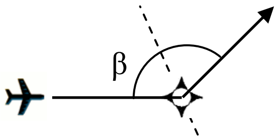

Fig. 4. Plan view of bisecting wayline (dashed line) across the interior course change angle, β.

Wayline crossing can be determined as follows:
First, the interior course change angle, β, between the current leg course, θ_k, and the next leg course, θ_{k+1}, can be calculated.

$$\beta = \pi - (\theta_{k+1} - \theta_k) \tag{2}$$

However, β must be normalised between +π and - π to correct for course changes greater than 90 degrees.

$$\beta_{norm} = norm(\beta, \pm\pi) \tag{3}$$

Then, a radial, R, from the waypoint along the bisector in the direction of the turn can be defined.

$$R = \theta_{k+1} + \frac{\beta_{norm}}{2} \tag{4}$$

The perpendicular radial to R on the side containing the next leg, $R_\perp$, is

$$R_\perp = \begin{cases} R + \pi, for\ norm(\theta_{k+1} - \theta_k, \pm\pi) < 0 \\ R - \pi, for\ norm(\theta_{k+1} - \theta_k, \pm\pi) > 0 \end{cases} \tag{5}$$

Given $R_\perp$ and the bearing from the waypoint to the aircraft, $\theta_{k,ac}$, the normalised angular difference between $\theta_{k,ac}$ and $R_\perp$ can be found, and the wayline check can be completed:

$$\Delta\theta_{norm} = norm(\theta_{k,ac} - R_\perp, \pm\pi) \tag{6}$$

The wayline is crossed if $\Delta\theta_{norm}$ is less than π/2, as shown in the wayline crossing geometry in Figure 5.

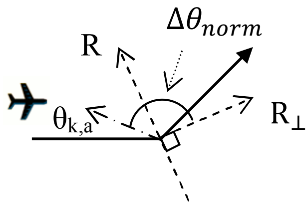

Fig. 5. Wayline crossing geometry.

The algorithm, illustrated in Figure 6, operates by sequentially evaluating each new waypoint that is to be merged into the flight plan as a *virtual aircraft*. Given the original flight plan, the current active leg, and a sequence of new waypoints to be merged, the first new waypoint to be merged can be evaluated as a *virtual aircraft* against the terminating wayline of the current active leg. If the *virtual aircraft* has crossed the wayline, the next leg in the original flight plan is flagged and evaluated in the same way. This is repeated until the *virtual aircraft* fails to cross the wayline of the flagged leg. Then, the next new waypoint to be merged can be considered the *virtual aircraft,* and evaluated against the flagged leg.

Once all new waypoints have been checked in this manner, the original flight plan legs from the current active leg, up to but not including the flagged leg, are removed and replaced with the new waypoints. The first of the new waypoints is then set as the terminator of the active leg, completing the merge.

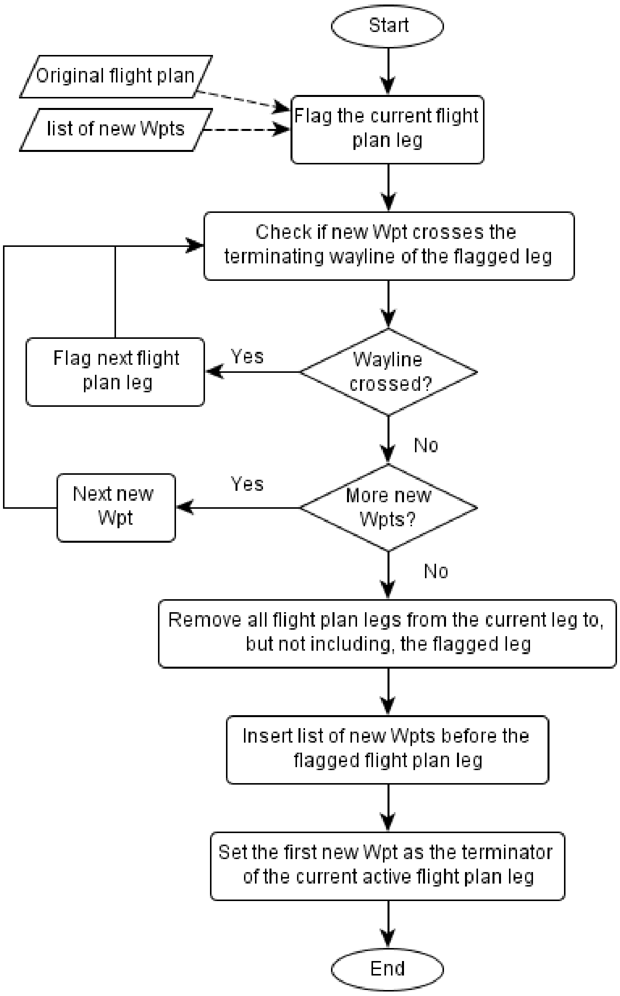

Fig. 6. Illustration of an automated algorithm for inserting and merging trajectory modification commands into a flight plan.

4 Simulation of Automated Capabilities

Having described two algorithms that could potentially aid FMS and ATM automation (Section 3), this section uses aircraft and FMS simulations to show the operation and usefulness of the automated capabilities. Two simulation cases are discussed; one case for each of the presented algorithms. In both simulation cases, a walkthrough of how each algorithm arrives at the desired solution is outlined. The following simulations have been designed to show scenarios in which FMS's without the discussed automated features would require pilot intervention or give rise to the possibility of inconsistent or unpredictable navigation behaviour.

4.1 Case 1: Automatically rejoining a flight plan after a temporary deviation.

Case 1 simulates a scenario in which an aircraft is executing a flight plan, but FMS managed navigation is suspended due to a disruption causing the aircraft to navigate significantly off-track (For example, avoiding certain weather conditions, executing an ATC heading vector or executing the commands of an automated conflict or collision avoidance system). Once the disruption has passed, the FMS should ideally be able to automatically update the active leg so that subsequent navigation is desirable (correct leg selected) and predictable. This scenario is illustrated in Figure 7.A. The aircraft, depicted by the filled arrow, has deviated off the flight plan and wishes to automatically rejoin the flight plan. The last active leg before the deviation was *Segment N* and for this simulation, it is assumed that the active leg has not been updated during the flight plan disruption. Three approaches to rejoining the flight plan were simulated: 1) Rejoin using the last known active leg, with no further automated action; 2) Rejoin by enabling the existing auto-sequencing function to attempt to select a suitable active leg (Figure 7.B); 3) Finally, rejoin by executing the segment discovery algorithm to select the active leg. The desirable result is that *Segment N+3* is selected.

Ground tracks of Approach 1 have not been shown because it is likely that a discontinuity will be detected, or that subsequent behaviour will be very implementation specific. Either way, any approach that attempts to return the aircraft back to the last active leg is undesirable.

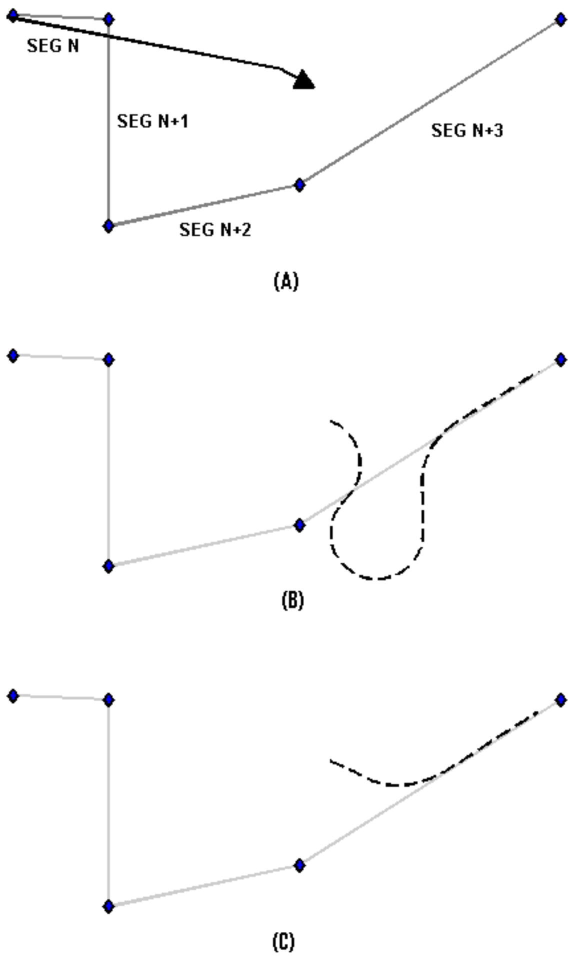

Fig. 7. Aircraft diverted from a flight plan (A) and the subsequent ground tracks using segment auto-sequencing (B) and the segment discovery algorithm (C).

Simulation results of Approach 2 (auto-sequencing) are shown in Figure 7.B. From the last known active leg, the FMS attempts to automatically sequence the legs, but does not sequence beyond *Segment N+1* immediately. In this simulation, sequencing does not advance until the aircraft passes a wayline defined as perpendicular to the current leg at the point of turn initiation to the next leg. After a short period of turning clockwise to intercept *Segment N+1*, the automatic sequencing advances to *N+2*, then *N+3* immediately. Unfortunately, the initial incorrect selection of *Segment N+1* as the active leg to rejoin results in an undesirable and inefficient flight plan rejoin. The authors acknowledge that, in reality, there is likely to be a variety of resulting auto-sequencing behaviours for the variety of available FMS systems. However, this highlights the need for consistency in FMS behaviour if trajectory prediction errors are to be reduced in future automated ATM systems.

Approach 3 used the automated segment discovery algorithm with the resulting simulated ground tracks shown in Figure 7.C. The algorithm selected the desired active leg, resulting in the aircraft intercepting and tracking *Segment N+3*. The result was obtained by the following steps:

1. The aircraft was not in the boundary of *Segment N*, so not considered.
2. The aircraft was in the boundary of *Segment N+1*, so the XTE was recorded.
3. The aircraft was not in the boundary of *Segment N+2*, so not considered.
4. The aircraft was in the boundary of *Segment N+3*, so the XTE was recorded.
5. The XTE to *Segment N+3* was the smallest, so it was selected.

In summary, although Approach 2 is likely to work in many cases, some scenarios and flight plans may result in different behaviours being executed by different vendors FMS's. Future automated ATM systems should seek to remove likely inconsistent behaviours. The automated algorithm of Approach 3 yields not only the desirable active leg, but also consistent automated behaviour that could be incorporated into FMS's or trajectory predictors within ATM automation tools.

4.2 Case 2: Automatically merging ATC generated flight plan modifications.

The second set of simulations demonstrates a scenario where an automated tactical separation management system has generated a flight plan modification consisting of three waypoints (squares in Figure 8.A) to correct a previous resolution issued by a strategic separation management system that failed to resolve the conflict. These new segments must either be merged with the active flight plan (diamonds in Figure 8.A) and then up-linked to the aircraft, or up-linked to the aircraft and then merged with the active flight plan.

Figure 8.B traces the ground tracks of two incorrect outcomes, illustrating the challenge of automatically modifying flight plans. The first case (dashed grey track) demonstrates the need for an intelligent method of merging flight plans. The flight plan segment was inserted directly into the active flight plan before the original active waypoint, resulting in a flight path that backtracks to that point following the resolution manoeuvre. This behaviour is undesirable and could be prevented using a more intelligent merging routine.

The second flight (solid black track in Figure 8.B) illustrates the limitations of overly-simplistic merging routines. In this case, the final new waypoint was merged with the closest leg in the original flight plan. This again results in an undesirable flight path that bypasses a significant portion of the original flight plan.

To avoid these two errors, the flight plan merging algorithm described in Section 3.3 was applied to this scenario. The algorithm processed as follows:

- Point *ATC 1* did not cross the *WPT M* wayline.
- Point *ATC 2* crossed the *WPT M* wayline but not *WPT M+1*.
- Point *ATC 3* crossed the *WPT M+1* wayline, but not *WPT M+2*.

Thus, *WPT M* and *WPT M+1* were removed from the flight plan, and replaced with *ATC 1…3*.

The resulting flight plan and ground track are shown in Figure 8.C, demonstrating a predictable and desirable merge between the two flight plans. This consistency is desirable in a more highly automated ATM environment.

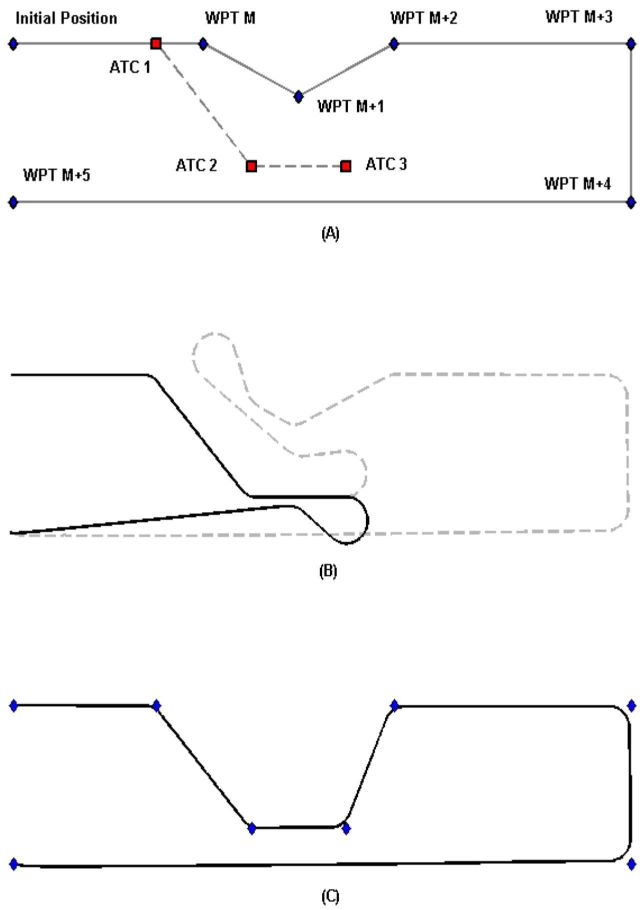

Fig. 8. Resulting ground tracks after incorrectly (B) and correctly (C) automatically merging points *ATC 1…3* into a flight plan defined by points *WPT M…M+5*.

5 Conclusions

This paper has described two algorithms that could potentially provide steps towards increased airspace automation: An automatic flight plan segment discovery algorithm and an automatic flight plan merging algorithm. Both of these algorithms can be applied in either airborne or ground automated ATM systems.

Simulation results demonstrate that these two typically manually processes can be achieved automatically with relatively straightforward algorithms. The algorithms offer a route to reduce pilot and controller workload, increase automation, and provide both aircraft and ATM automation tools with consistent awareness of the current active leg and route of flight in the event this information is lost.

The work in this paper highlights the need for consistent and predictable FMS and ATM automation tools behaviour to decrease trajectory prediction errors in the system. Such behaviour needs to be agreed to support the safe integration of automated technologies and UAS into future ATM systems.

Acknowledgements

The authors wish to acknowledge the support of Boeing Research and Technology, especially Richard Baumeister, Regina Estkowski and Ted Whitley. G.W. Flathers acknowledges the support of the Marshall Aid Commemoration Commission and the Air Force Institute of Technology. Additionally, we acknowledge the Smart Skies project (a joint research and development project involving Boeing Research & Technology (BR&T), Boeing Research & Technology Australia (BR&TA), the Australian Research Center for Aerospace Automation (ARCAA)) for providing a test platform on which many of the ideas in this paper have been tested.

Disclaimer

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

References

[1] Joint Planning and Development Office. Concept of operations for the next generation air transportation system. Version 3.0, 2009.

[2] SESAR Consortium. Deliverable 3 - The ATM target concept. Doc. No. DLM-0612-001-02-00, 2007.

[3] Spitzer C.R. *The avionics handbook*. 1st edition, CRC Press, 2001.

[4] DeGarmo M.T. Issues concerning integration of unmanned aerial vehicles in civil airspace. MP 04W0000323, MITRE, Virginia, 2004.

[5] Vos D.W. F*ive steps to facilitating the convergence of manned and unmanned aviation*, Rockwell Collins, 2009.

[6] Vos D.W. *Awareness: The most critical step to facilitate the convergence of manned and unmanned aviation*, Rockwell Collins, 2010.

[7] Weibel, R and Hansman R.J. Safety considerations for operation of unmanned aerial vehicles in the national airspace system, MIT International Center for Air Transportation Report, No. ICAT 2005-01 ICAT 2005-01, 2005.

[8] Baumeister R, Estkowski R and Spence G.T. Automated aircraft tracking and control in class g airspace. *Proc. of ICAS 2010 Congress*, Nice, France, September 2010.

[9] Baumeister R, Estkowski R and Spence G.T. Test architecture for prototyping automated dynamic airspace control. *Proc. of CEAS European Air and Space Conference*, Manchester, UK, 2009.

[10] Schneider P.J and Eberly D.H. *Geometric tools for computer graphics*. Morgan Kaufmann Publishers, pp 192-193, 2003.

[11] Mondoloni S. Aircraft trajectory prediction errors: including a summary of error sources and data. FAA/Eurocontrol Action Plan 16: Common Trajectory Prediction Capabilities, 2006.

Contact Email:

gts@aerosoft.co.uk

Copyright Statement