

Introduction

For the Air Traffic Management course, a plugin was designed that adds functionality to the BlueSky Air Traffic simulator. This plugin is an Arrival Manager (AMAN) that can assist in sequencing arriving aircraft. The plugin periodically outputs the predicted arrival sequence to the BlueSky command window. This can help Area Controllers or perhaps Approach controllers to efficiently sequence the incoming aircraft. The AMAN support tool predicts the arrival sequence based on the estimated arrival times at the transition point. These times are calculated based on the trajectory the aircraft will presumably fly. Furthermore, the AMAN plugin will filter aircraft that are relevant for the controller and color all aircraft, such that the controller can see what traffic he/she has to work with.

Working principles

The arrival manager works based on trajectory estimations. However, before a trajectory can be estimated, the aircraft on the radar should be filtered. This is done by a few criteria. First, all aircraft with their destination corresponding the selected airport are taken. These aircraft flight plans are then examined. The arrival manager works for one Initial Approach Fix (IAF), hence this IAF should be present in the flight plan. All aircraft that have the selected IAF in their flight plan will be considered in the AMAN. When no IAF's are in the flight plan, the aircraft will be colored yellow as it's flight plan is incomplete. Lastly, the remaining set of aircraft will be checked whether or not they are heading towards the IAF, or are already past this waypoint (and thus under approach control). This yields a set of aircraft that should be sequenced by the air traffic controller of the specified sector. Aircraft in this set are called active and will be colored lime. The other aircraft will be blue, and aircraft that have incomplete information (and thus might potentially be sequenced by the ATCo) are coloured yellow. Lastly, aircraft that are active but might be too close to another at the merge point will be marked red. The different markers are shown in Figure 2.



Figure 2: blue: Non-active aircraft. Lime: Active aircraft in the AMAN sequence. Red: Active aircraft, but potentially no clean merge. Yellow: incomplete aircraft, action needed as no IAF is assigned.

After filtering, a set of active aircraft is parsed into the trajectory calculation block. The input this block requires aircraft parameters (primarily velocities), wind data and the flight plan. This allows to calculate an estimated arrival time. This is done by segmenting the data into legs of the flight plan. These legs of the flight plan have a certain distance, and might have speed and altitude constraints. The speed constraint and altitude constraint at the IAF are fixed at 250kts IAS and 10000ft. For each leg of the flight plan up to the IAF, the flight time is calculated. This sum gives a final estimated arrival time. The segment flight time can be calculated by dividing the distance of the leg by the ground speed profile. Calculating the distances can be done easily by the relation given in Equation 1. In this equation,

d is the distance between the coordinates, R is the earth radius, lat_A, lon_A are the coordinates of the current aircraft position, or in case of the following legs they represent the position coordinates of the previous waypoint. lat_B, lon_B are the coordinates of the destination waypoint.

$$d = R * \arccos((\cos(lat_A) * \cos(lat_B) * \cos(lon_A - lon_B)) + (\sin(lat_A) * \sin(lat_B))) \quad (1)$$

The speed profile is less straight forward, as the ground speed is highly dependent on wind and aircraft performance. Aircraft performance is not taken into account due to time constraints, but can be implemented in a later refinement. The speed of aircraft is assumed to change instantly at waypoints. When short segments are present in the flight plan, a high fidelity is reached after all. Hence the assumption will have limited effects on the end result. Wind data is used in order to obtain a re-presentable picture. This is done by segmenting the leg into parts of 10km. The knots of these segments are sampled in the wind data model in the 3D plane. The altitude that is used to sample the wind is based on average aircraft descent profiles. For every thousand feet, 3 nautical miles will be flown. This makes sure the wind will be sampled at altitudes that roughly correspond the final trajectory. A graphical representation is shown in Figure 3.

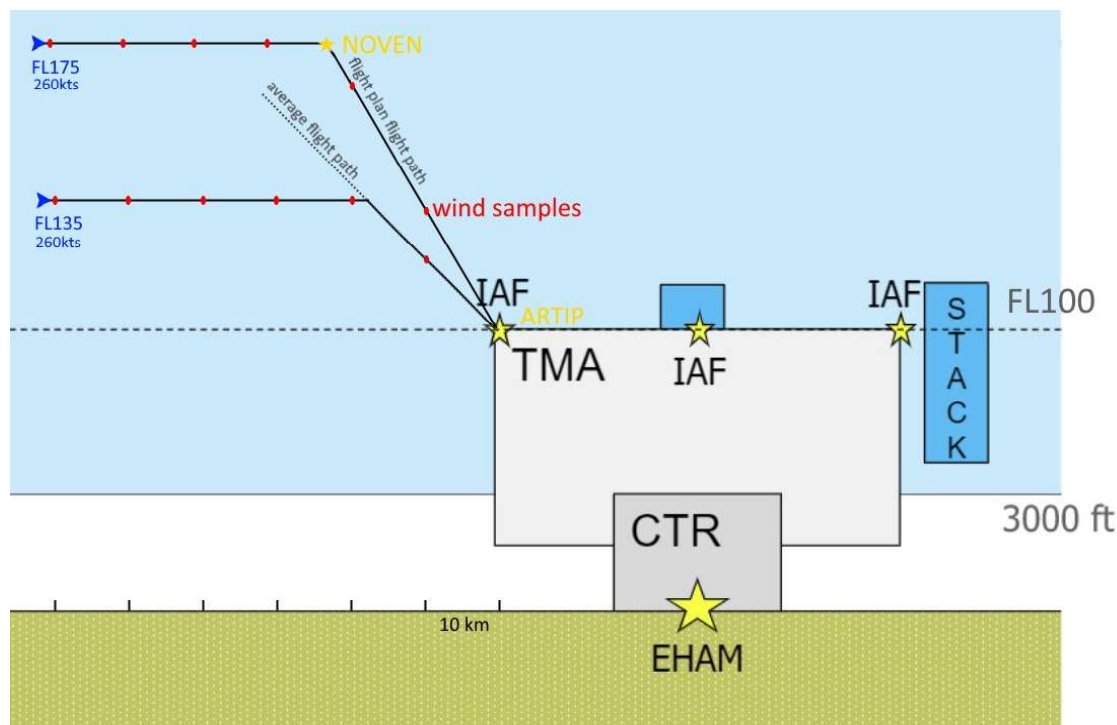


Figure 3: The wind sampling algorithm estimates based on average flight paths when the aircraft will descent and samples this trajectory. Sampling points are marked red. When the aircraft is past this point it draws a direct line from the current altitude to the next waypoint and samples these altitudes.

When all wind samples are taken, the aircraft indicated or selected airspeed can be converted to an expected ground speed. This is used to calculate the flight time by dividing the length of the segment by the wind corrected speed. Doing this for all legs of the flight plan will yield expected travel times for all legs. It is important however to have altitude and speed constraints in the flight plan (VNAV). This will yield the highest fidelity in arrival time predictions. When no selected speeds are inserted, it is assumed that the aircraft flies the trajectory with the current speed. This is of course less accurate, but will produce results that are not very off. Nevertheless, the more detailed a flight plan is, the higher the accuracy of the AMAN.

The final inbuilt functionality is the proximity alert. Based on arrival times, it can be checked how close aircraft are arriving at the IAF. The difference between arrival times should be large enough to ensure safe separation. This is set to be 2 minutes for now, but can later be changed to incorporate time based separation RECAT-EU values based on the pair combination. The first aircraft to arrive is prioritized and stays green. The second in row will receive a red color marking and should thus be slowed down to arrive a bit later.

conclusion

In order to show the functionality of the plugin, an example scenario has been created. This scenario is called "AMAN-scenario" and has a high density inbound peak from the east arriving at Schiphol. These flights have identifiers specified by KL2xx. There is also some traffic from the west, which is there to show that when the user changes the active IAF in the plugin (SUGOL in this case) that this also works for other waypoints. These aircraft are identified by KL1xx. Lastly, there is one aircraft that has EHAM as destination but no flight plan yet and can be identified by KL301. This aircraft will need controller attention in order to sequence. The example scenario, computes the arrival times every 10 seconds and prints the expected arrival order. The scenario contains a wind model that has a southern wind that blows 113 km/h at FL400. This reduces to a wind from heading 270 at a speed of 56 km/h at FL100. The wind is interpolated in between for the wind at respective altitudes. The AMAN tool can help a controller to decide which aircraft is arriving first and thus make the most efficient sequence. The list is continuously updated and might not always be 100% correct with respect to the final sequence, primarily due to the assumptions made. The proximity warnings also represent that these aircraft are close and can thus be flipped position in the sequence easily. It is a measure of accuracy in that sense as well. The AMAN tool developed delivers a very workable sequence that the area controller can use to make decisions while merging. It has minor deficiencies that are primarily caused by the assumptions. When improving accuracy, the biggest steps can be made to implement aircraft specific performance. The code was written to easily hook up aircraft performance parameters. This can also be used to implement RECAT-EU time based separation for maintaining separation at the IAF. When looking at the AMAN functionality as defined by EUROCONTROL in Figure 4, the inputs in green circles have been touched upon in this plugin. The red circles can still be implemented.

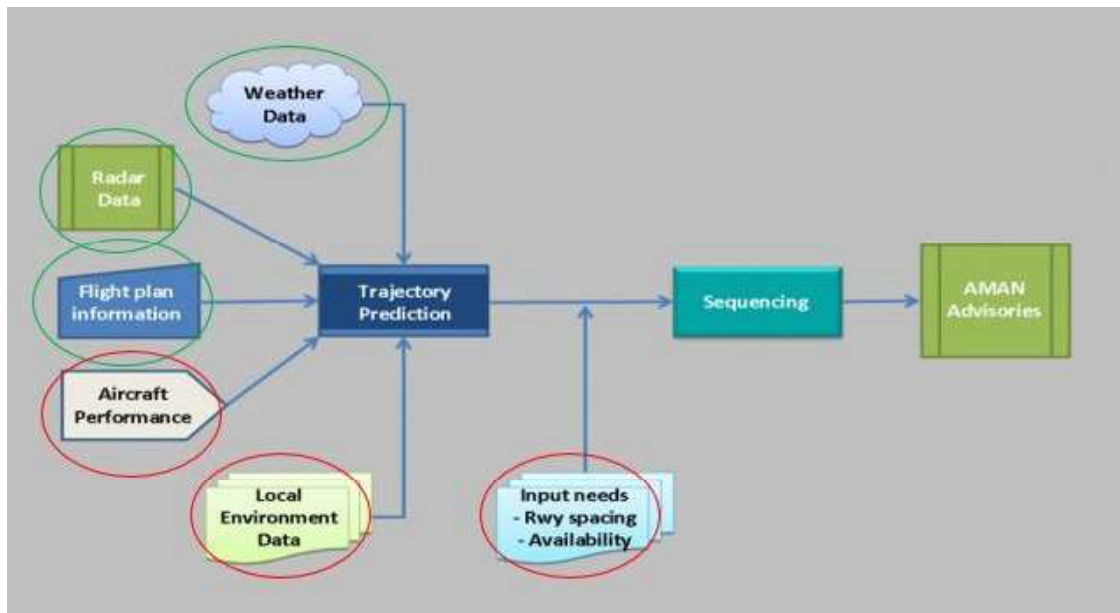


Figure 4: EUROCONTROL's schematic layout of AMAN functionality