4D trajectory prediction is the central part of many automated decision support tools of the future air traffic management systems. It is used to calculate, in advance, where exactly the aircraft is going to be at each point in time, therefore, allowing potential conflict resolution and route negotiation before the aircraft even takes off. Trajectory prediction relies heavily on aircraft performance calculation which must be somewhat simplified in order to facilitate fast computation of thousands of such routes, but not overly so that it becomes inaccurate. Aside from performance calculation, accuracy of trajectory prediction depends on correct representation of aircraft operation. Therefore, accurately modelling the flight management system is essential to trajectory prediction. This paper considers several possible improvements in flight management system model, mainly in bank angle setting and lateral guidance. Due to many changes in altitude and heading, as well as in thrust and attitude of the aircraft, one of the hardest phases of flight to predict is the arrival phase. Since one of the major goals of trajectory prediction is arrival queue management, this paper focuses on accurately predicting 4D trajectories during the final phases of flight. This paper also deals with conflict resolution and queue management in final phases of flight.

1. INTRODUCTION
The current Air Traffic Management System was designed decades ago and, even with the 6.6% temporary drop in traffic in 2009 (EUROCONTROL, 2010, p.5), it is going to reach its operational and technological limits in near future. In response to the ATM challenge, the European Commission (EC) launched the Single European Sky ATM Research (SESAR) programme, with the objective to achieve a future European ATM System for 2020 and beyond, which can among other goals, enable a 300% increase in capacity relative to performance in 2005 (EUROCONTROL, 2006, p.6). Increase in capacity of such magnitude will significantly decrease delays, both on ground and in the air, thus reducing the cost and increasing the efficiency of airspace user's operations. The achievement of capacity targets in SESAR programme will be, in part, supported by 4D trajectory management, new separation modes, and
wide range of controller support tools (EUROCONTROL, 2007, p.10). This concept requires fast and reliable exchange of data between aircraft and all ground facilities so that the power of shared information can be fully harnessed. With this purpose in mind a System Wide Information Management (SWIM) environment will be implemented. Within the context of SWIM, 4D trajectories will become basis upon which the airspace will be managed. As opposed to 3D trajectories used now, 4D trajectories will accurately describe aircraft state in regards to its physical position (3D coordinates) and time. Airspace users will be the ones who determine their ideal 4D trajectories and collaborative decision tools will be used to negotiate final trajectories in regards to airspace or airport capacity constraints, thus creating the Network Operations Plan (EUROCONTROL, 2007, p.9). Network operations plan will have to be optimized in such a way as to make the most use of the scarcest of the resources which will probably be airport runways. This is what makes 4D trajectory prediction most important during the arrivals and departures.

During the development of future concepts that should enable achievement of SESAR targets, trajectory prediction is used as a tool that can provide data necessary for airspace and airport capacity prediction and assessment. Fast-time simulations are used as a lowest form of concept validation, followed by real-time human-in-the-loop simulations and in-flight validation. For fast-time simulations it is important to find the adequate balance between simulation speed and accuracy. One of the purposes of this paper is to find methods that can increase simulation accuracy without compromising simulation speed. Some of the improvements in trajectory prediction presented here may seem minute when considering en-route flight; however, when considering terminal operations it will be shown that even these minor improvements can have significant impact on accuracy. Furthermore, it will be shown that these small improvements, when coupled with conflict detection and resolution algorithm, can create flyable and more realistic trajectories.

2. TRAJECTORY PREDICTION

There are many different ways to model aircraft movement, some of which are: look-up tables, kinetic models and kinematic models.

The source for the data in look-up tables is extracted from the climb and descent tables as used by the aircraft operators. Vertical profile is divided into several altitude bands and each band is assigned a constant speed. Look-up tables tend to be very inaccurate and need to be tuned depending on the application; however, they are easy to implement and computationally inexpensive (Suchkov, 2003, p.7).
Kinetic models, such as EUROCONTROL’s Base of Aircraft Data (BADA), are based on the fact that commercial passenger aircraft are operated at relatively small flight path angles, an assumption which enables significant simplification of the complex differential equations that govern the aircraft motion. The aircraft model behind BADA is referred to as the Total Energy Model (TEM), which can be considered as being a reduced point-mass model. TEM will be extensively used in this paper.

Kinematic models are based on purely parametric approach. In such methods the primary flight path characteristics are modelled and validated directly from the reference flight profile data provided by the aircraft manufacturers. This approach provides accurate model of the aircraft behaviour without attempting to model underlying physics.

The overview of the trajectory prediction model used in this paper can be seen in Figure 1. The main part is the Trajectory Engine which consists of two systems in constant interaction. First is the Aircraft Dynamic system which, based on six state variables, four inputs, and three disturbances, determines the change of aircraft state variables. Since three of the six state variables represent aircraft coordinates, output of this system effectively provides trajectory prediction. Second is the Flight Management System (FMS) model which, based on aircraft current position, flight script, operational procedures and limitations, and a number of other factors, determines the change in aircraft inputs which ensures fulfilment of flight script goals. These two systems will be more discussed in the following subsections.

![Figure 1. Overview of the trajectory prediction model](image-url)
Atmosphere model provides air density and temperature information for the given aircraft altitude, information needed for accurate aircraft performance calculations. Atmosphere model can also provide wind information.

Flight script is the main source of the aircraft intent information. It consists of a list of Trajectory Change Points (TCPs) which are points in space at which the aircraft state (e.g. heading, speed, and altitude) is modified. Each TCP is coupled with the Expected Time of Arrival (ETA), thus providing information needed for speed determination at that particular leg of the flight. In addition, flight script can be expanded to include altitude, speed, thrust (as in noise abatement procedures), and other constraints with the purpose of accurately describing airline preferred routes, avoiding hazard and no-fly zones or regions with severe weather. During the flight, flight script can be updated as new information becomes available or conditions change.

BADA provides data and methods for calculation of flight performance of hundreds of aircraft types (EUROCONTROL, 2009). Besides raw aircraft performance data, more importantly, it provides methods for determination of realistic trajectories based on airline operating procedures and general operating procedures limitations (e.g. longitudinal and normal acceleration limits).

2.1. Aircraft Dynamics

As shown in the HYBRIDGE project (Glover and Lygeros, 2003), for ATM simulation purposes, an aircraft can be adequately modelled using a Point Mass Model (PMM), using six state variables ($x$), three inputs ($u$) and three disturbances ($w$).

First three of the state variables are $x$ and $y$ coordinates and altitude, $h$. Changes in these three variables define the predicted trajectory. The rest of the six state variables are True Air Speed (TAS), $V$, heading angle, $\psi$, and aircraft mass, $m$.

$$x = [x_1, x_2, x_3, x_4, x_5, x_6] = [x, y, h, V, \psi, m]$$  \hspace{1cm} (1)

Three input variables are identical to pilot controls. These are the engine thrust, $T$, the bank angle, $\phi$, and the flight path angle, $\gamma$. In addition to those three variables, a fourth variable, drag coefficient, $C_D$, can be introduced (Poretta et al, 2008) because the pilot has the ability to control several aircraft configuration options (e.g. flaps, landing gear, spoilers).

$$u = [u_1, u_2, u_3, u_4] = [T, \phi, \gamma, C_D]$$  \hspace{1cm} (2)
Three disturbances are three components of the wind speed vector, each of them being parallel to one of the three axes. At the time of writing this paper, only constant nominal wind has been implemented.

\[
\mathbf{w} = [w_1 \ w_2 \ w_3] = [w_x \ w_y \ w_z]
\]  

(3)

After defining vectors relevant to state variables, inputs and disturbances, nonlinear control system can be used to capture the aircraft motion (Glover and Lygeros, 2003).

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3 \\
\dot{x}_4 \\
\dot{x}_5 \\
\dot{x}_6
\end{bmatrix}
= 
\begin{bmatrix}
x_4 \cos(x_5) \cos(u_3) + w_1 \\
x_4 \sin(x_5) \cos(u_3) + w_2 \\
x_4 \sin(u_3) + w_3 \\
\frac{u_4 S \rho}{2} \cdot \left(\frac{x_4^2}{x_6}\right) - \left[g \sin(u_3)\right] + \left(\frac{u_1}{x_6}\right) \\
\frac{C_L S \rho}{2} \cdot \left(\frac{x_4}{x_6}\right) \cdot \sin(u_2) \\
-\eta \cdot u_1
\end{bmatrix}
= f(x, u, w)
\]  

(4)

This system uses state variables, inputs and disturbances, along with additional terms such as aircraft total wing surface area, \(S\), air density at altitude, \(\rho\), acceleration due to gravity, \(g\), aerodynamic lift, \(C_L\), and fuel consumption factor, \(\eta\), to calculate the change in state variables, \(\dot{x}\). Adding \(\dot{x}\) to \(x\) gives new aircraft state variables.

2.2. Flight Management System

The purpose of the Flight Management System (FMS) model is to determine how to change inputs in order for aircraft to follow the desired path from the flight script. The inputs that FMS uses are similar to the inputs that pilots use to control an aircraft. As mentioned in previous section these inputs are: thrust, bank angle, flight path angle, and drag coefficient. It is assumed that the aircraft is in coordinated flight at all times so no input is needed for yaw control.

First thing an FMS must do is determine the aircraft position and speed relative to the desired path and speed. Next, it must determine the inputs needed to correct differences between the two.

2.2.1. Relative position. Aircraft position relative to the desired path in horizontal plane can be expressed using two variables, cross track error (CTE) and heading error (HE). CTE is the distance from the aircraft to the nearest point on the desired path, and HE is the difference between current aircraft heading and the desired heading (Figure 2).
In Figure 2, $P^H(t)$ is a vector indicating aircraft position at time $t$, $O^H(i)$ and $O^H(i+1)$ are TCP coordinates for TCPs $i$ and $(i+1)$ respectively. Heading error, $\theta(t)$, can be calculated simply by subtracting aircraft heading from the desired heading: $\theta(t) = \psi(t) - \psi(i)$. Cross track error, $\delta(t)$, is defined by: $\delta(t) = [\sin(\psi(i)) \cos(\psi(i))] \cdot O^H(i) \cdot P^H(t)$, where $O^H(i) \cdot P^H(t) = \Omega P^H(t) - \Omega O^H(i)$ (Poretta et. al. 2008).

Aircraft altitude is compared to the altitude of the next TCP in order to determine whether the aircraft needs to climb, descent, or remain level. In order to determine whether the aircraft needs to accelerate, decelerate, or maintain constant speed, aircraft speed is compared to the desired speed. Desired speed, $V_{nom}$, is defined as True Air Speed (TAS) which the aircraft needs to maintain so that it reaches next TCP at the required time. It can be easily calculated by dividing remaining distance to next TCP with the time left for reaching it, of course, taking wind into the consideration.

2.2.2. Inputs. When the aircraft position relative to the desired path has been determined, FMS must decide which of the four inputs to change and how much. Input number four, drag coefficient $C_D$, is easiest to set. According to BADA, $C_D$ changes only with change in aircraft configuration (e.g. landing gear, flaps) which depends only on height above ground. BADA prescribes at which height above ground which fixed $C_D$ should be used. Other inputs can be divided into two groups: thrust and flight path angle (used to control climb and speed), and bank angle (used to control horizontal position).

For the level flight, flight path angle is always set to zero. When changing flight path angle, care must be taken not to exceed maximum allowed normal acceleration set in BADA. Normal acceleration limit set in BADA is much lower than the one set in aircraft operations manual because it is chosen to
ensure passenger comfort, not aircraft structural integrity. For the purpose of this paper, when changing flight path angle it is always increased or decreased by the maximum allowed amount. In this way the trajectory of a commercial airliner is more accurately predicted.

Maximum cruise and climb thrust can be calculated using BADA. When the aircraft is flying level and it needs to accelerate it must not use more thrust than the thrust that would cause it to accelerate at the maximum allowed longitudinal acceleration set in BADA.

When climbing or descending with constant speed, Energy Share Factor (ESF) is used to determine how much energy will be needed to maintain speed and the rest is used to increase or decrease altitude. ESF is calculated depending on the speed hold mode (constant CAS or constant Mach) and troposphere mode (below or above tropopause) as can be seen in (EUROCONTROL 2009, eq. 3.1-5 to eq. 3.1-8). When the aircraft is accelerating in climb or decelerating in descent, ESF is set to 0.3. For aircraft accelerating during the descent or decelerating during the climb, ESF is set to 1.7.

Bank angle controller used in this paper was modelled so that it mimics pilot’s decision process. Unlike some other controllers in the literature, this one increases or decreases bank angle by no more than 2°/s, thus simulating turn roll-in and roll-out. It has been shown in (Swierstra and Green, 2003) that so called “circular arc” turns introduce an error that is too big for accurate simulation of terminal area operations. Adding this feature to the FMS improves accuracy of trajectory prediction, especially so for the turns with a large change in heading. In Figure 3, comparison of the two types of turns is shown. Green is the circular arc turn and blue is the turn with roll-in and roll-out guidance. In both cases the change in heading is 115° with TAS of 200m/s and maximum bank angle of 35°. The distance of turn end points is almost 2000m and the time needed for completion of circular arc turn is 10 seconds less than the time needed for completion of the roll-in/roll-out turn.

![Figure 3. Circular arc turn vs. Roll-in/Roll-out turn](image)
Bank angle controller is also combined with logic-based interception system that is used to lead the aircraft towards the desired flight path for situations when the aircraft veers off the desired track (e.g. for fly-over waypoints). For large cross track errors it chooses large interception angles, but as the aircraft approaches the desired flight path, interception angle becomes smaller and smaller until it smoothly reaches zero at the same moment the CTE becomes zero.

3. CONFLICT DETECTION AND RESOLUTION

Automatic conflict detection and resolution (CD&R) is essential for fast-time simulations in air traffic management since it is impossible to provide human assistance for such large volumes of traffic. Conflict resolution methods can be categorized into three different cases according to the methods by which a solution is obtained (Hwang and Tomlin, 2002). First, optimized conflict resolution produces a resolution manoeuvre which minimizes a certain cost function such as fuel consumption, time, or distance travelled. Next are rule-based conflict resolution methods in which a conflict is resolved by using pre-described rules. Last class of conflict resolution methods is based on force field theory which assumes that the aircraft is flying in the force field generated by the potential function with attracting force coming from the desired destination and repulsive force coming from the other aircraft.

Force field methods use elegant equations which yield solutions that have one or more discontinuities, thus making them not flyable by real aircraft and not suitable for the intended purpose of this paper. Rule-based methods may require many rules to cover for all the possible situations and can become unsafe when the two different rules can be applied to the same situation due to aircraft state uncertainties. There are many optimized resolution methods, one of which uses genetic algorithms and predefined manoeuvres to find the globally optimal solution (Durand and Alliot, 1995), similar method is used in this paper (Figure 4). Since the purpose of the algorithm used in this paper is to find the best possible outcome, global solution is preferred to pairwise solutions because

![Conflict Detection and Resolution Algorithm](image-url)
local optimums do not necessarily amount to globally optimal solution. Downsides of the genetic algorithms are computational effort needed for finding the solution, and the fact that they can give two different solutions for the same problem. These downsides disqualify the genetic algorithm methods from using them for on-board conflict resolution systems; however, they can still be used for centralized conflict resolution as well as fast-time simulations.

3.1. Conflict detection and clustering

Once the trajectory prediction has been made, it is time to detect conflicts. A conflict is defined as a loss of separation between aircraft. This happens when an aircraft enters the protected zone of another aircraft. Protected zone used in this paper was a cylinder with 5 nm radius and 2000 ft height (Figure 5). It should be noted that in the future, with the advent of the performance based navigation, protected zones will be smaller.

![Figure 5. Protected Zone](image)

Sometimes, more than two aircraft are involved in the conflict. Situations like that, with all the aircraft involved, are called conflict clusters. Conflict cluster lasts from the time the first two aircraft come into a conflict until the last two aircraft (possibly entirely different ones) come out of the conflict.

![Figure 6. Simple conflict vs. Conflict cluster](image)
For example, in Figure 6.b a conflict cluster can be seen. Black lines are planned tracks for each of the aircraft with arrows indicating direction of flight and green areas indicating times at which the aircraft is in conflict with one or more other aircraft. Aircraft A and B enter the conflict at approximately same time as aircraft D and E. If not for the aircraft C, these two conflicts would remain separate. So, although the aircraft B and E are far from being in conflict themselves, they are in the same conflict cluster. Conflict clusters like these can span tens of miles (especially if the aircraft are on near parallel tracks) and involve tens of aircraft.

3.2. Conflict resolution
Conflict resolution must start some time before the first conflict in the cluster occurs. The amount of time depends on aircraft speed because the protected zone is of the same size for all aircraft, so when the fast aircraft starts its avoidance manoeuvre one minute before it enters another aircraft’s protected zone, it needs to make smaller course adjustment to avoid the other aircraft’s protected zone than the slow aircraft that starts to manoeuvre at the same time. This problem was solved by including the time of the beginning of the conflict resolution manoeuvre with other inputs for genetic algorithm. In that way, not only were the different speeds taken into account, but the time of the start of the manoeuvre was also optimized to incur the least possible cost.

For each aircraft, the conflict resolution manoeuvre is basically the same (Figure 7). It is made of three legs: original heading leg, avoidance leg, and return leg.

![Figure 7. Pre-set conflict resolution manoeuvre](image)

For each aircraft in a conflict cluster, genetic algorithm selects the time at which it will commence the avoidance manoeuvre, $t_1$, heading change, $\psi_a$, and time to fly on a new heading, $t_2$. After time $t_2$, the aircraft returns towards the final conflict point. Genetic algorithm searches through these inputs until it finds the conflict-free solution that costs each aircraft least time. There are some bounds however; maximum allowed heading change is $\pm 45^\circ$ so that aircraft always keep moving forward, and maximum allowed time for $t_1$ and $t_2$ is one third of the conflict duration so that the sharp turns are avoided at the point when the aircraft returns to its original heading. Sometimes it is not possible to find the conflict-free solution the first time. In that case the algorithm finds the solution with the least number of conflict points and then reiterates. The algorithm was tested on a horizontal superconflict with 8
aircraft flying towards the same point at the same speed. It was able to deconflict them in 4 iterations (Figure 8).

In Figure 8, areas of conflict are green, the scale is in meters. It can be seen that through iterations, the conflict cluster is broken down into smaller ones until it finally disappears. In the last section of Figure 8, the zoomed in centre of the place where the conflict used to be, it is visible that aircraft trajectories are smooth and flyable.

4. SIMULATION OF 4D ARRIVALS
The purpose of simulating 4D trajectory prediction during the arrivals (in this context arrivals are considered as a broader term, starting at the top of descent and ending with final approach) is to find new ways of routing traffic to aerodromes for landing. In reality this process would begin with collecting all 4D business trajectories from various airspace users, then making adjustments to some or all of them to satisfy safety, local constraints, and separation, thus creating the network plan. It is not currently exactly known
which strategies and tools will be employed to make this leap from lots of individual trajectories to coherent, safe, effective, and efficient network plan, nor is it known how that network plan will be updated.

For this paper, a somewhat simplified airspace was designed. It consists of just 16 aerodromes, all contained within a 1000km by 1000km box. No restricted zones or any other spatial constraints were implemented at this stage of research.

Fifteen business trajectories were created with aircraft taking off at 15 different aerodromes and all of them landing at the same, 16th aerodrome within 15-minute window (Figure 9). End of the trajectory prediction was at the final approach point.

![Figure 9. Trajectory prediction during arrivals](image)

This creates a typical bottleneck situation, common at many regional hubs.

Landing times were not allocated in advance for each aircraft; airspace users were allowed to decide when they wanted to land. In other words, landing times were stochastic. Future ATM framework will continue using take-off and
landing slots which are basically time constraints which may or may not be of
the same duration as the airspace user’s preferred business trajectory. Future
slot allocation will probably be coordinated between all aerodromes but it will
be extremely hard, if not impossible, to find a solution in which everyone will
be able to take-off and land at times that perfectly match the duration of their
preferred trajectory. Airspace users will need to find a solution that costs them
the least. For example, if the aircraft is time-constrained by slots to a 120-
minute flight for a route that ideally lasts only 115 minutes, the airspace user
will have to decide whether to slow down the aircraft, lengthen the route or
choose different altitude. Slots will be used in authors’ future research.

As expected, most conflicts occur at the point where all routes converge, in
this case, at the final approach point (Figure 9). The algorithm managed to
deconflict these aircraft while changing their estimated times of arrival by no
more than 30 seconds, however, first hints of possible problems began to
show. For example, conflict resolution algorithm added new legs to the
business trajectories for separation purposes, thus making them longer. This
casted the top of descent point to move; therefore aircraft reached the final
approach point at different altitudes than planned. Obviously, top of descent
point should be automatically readjusted so that aircraft reaches final
approach point at the desired altitude, but this example reveals one of the
issues with this kind of collaborative decision making: when changing
business trajectories to make them fit into the network plan, Air Navigation
Service Provider (ANSP) will have to know not only aircraft performance, but
also airspace user’s business intent and aircraft operational procedures.
Formulating business intent is much harder than just formulating preferred
trajectory because intent goes beyond just one flight. It evolves from a
complex decision-making process made deep within the airspace user’s
organisation. For example, airspace user designs the preferred trajectory for a
certain flight with take-off and landing times that perfectly fit both their
trajectory (duration-wise) and their business objectives. ANSP gathers all
trajectories, adjusts them and makes them fit into a network plan. In this
example ANSP moves the take-off and landing times five minutes later but
the trajectory itself remains unchanged. From the ANSP’s point of view the
efficiency is high, after all, the aircraft will fly the most efficient route from one
aerodrome to another; however, airspace user might find the new trajectory
completely unacceptable because it disrupts their crew or maintenance
schedule. Maybe at this point the airspace user would prefer to fly a less
efficient route but land on time.

The above example shows that optimization of the network plan is not all
about predicting the most efficient combination of trajectories. It can be
concluded that new ways of stating business intent should be developed.
5. CONCLUSION
This paper presents a combination of tools that can be used to accurately predict aircraft trajectories, identify possible conflicts and adjust the trajectories to resolve the conflicts. Also, it presents few minor adjustments in flight management system model, namely bank angle controller and path interception method, which should be used for accurate prediction of aircraft trajectories during the arrival phase of the flight. Furthermore, a slight modification of the existing global conflict resolution technique is shown.

Simulation of arrivals has shown that even globally optimal conflict resolution does not produce a viable network plan. Many problems have emerged that cannot be solved by such algorithms in their current state.

Authors’ future work will be focused on increasing the accuracy of trajectory prediction, inclusion of vertical and longitudinal conflict resolution techniques, defining trajectory negotiation process and formulating airspace user’s business intent.

6. REFERENCES


EUROCONTROL (2007). The ATM Target Concept. DLM-0612-001-02-00.


