A RAPID SCENARIO GENERATION TOOL FOR REPEATABLE SIMULATED TRAFFIC CONFLICTS IN FLIGHT SIMULATION

J. Gauci*, D. Zammit-Mangion*
***Cranfield University**

Keywords: *conflict simulation, runway incursions, rapid prototyping*

Abstract

A rapid scenario generation tool has been developed to support the rapid prototyping and simulation of traffic movements and the generation of simulated traffic conflicts on the runway in flight simulation. It consists of three main components: a traffic motion generator, a scenario definition and control unit, and a dynamic traffic control unit. The features provided by these three components provide flexibility and scenario repeatability in a dynamic environment.

1 Introduction

A rapid scenario prototyping tool was required to support the evaluation of a runway conflict alerting system developed by the University of Malta under the European FLYSAFE programme. The programme, running over the period 2005-2009, addresses the need for improved safety in commercial aviation. It aims to mitigate the threats associated with weather, terrain and traffic by developing the Next Generation Integrated Surveillance System (NG-ISS) and a Weather Information Management System (WIMS) [1]. The NG-ISS generates and manages alerts due to different hazards whereas the WIMS provides timely weather information to the flight crew.

In its address of traffic hazards, FLYSAFE is, amongst other activities, also focussing on runway incursions. To this effect, the need for a Runway Collision Avoidance Function (RCAF) to reside within the NG-ISS was identified. Broadly, the RCAF was required to provide timely alerts of an impending conflict on the runway and this function was then developed by the University of Malta [2] and evaluated at Cranfield University as a stand-alone unit prior to integration with other functionalities and further evaluation.

The RCAF essentially performs a surveillance function on the traffic on, and in the vicinity of, the runway and detects whether a conflict or potential conflict exists with the ownship (the aircraft equipped with the RCAF). In the event that a conflict is detected, the system alerts the pilots accordingly.

The surveillance function of the RCAF as specified under the FLYSAFE programme is based on Automatic Dependant Surveillance-Broadcast (ADS-B) [3]. With this technology, each aircraft broadcasts a data packet to surrounding aircraft. This data packet includes airspeed, GNSS-derived position, call sign and intent. With this information, each aircraft can then construct a virtual radar map of surrounding traffic. ADS-B data is transmitted via the MODE-S transponder at 1Hz.

Since the RCAF is intended to be used in safety-critical situations, it was required to undergo rigorous testing to assess several aspects of the design such as the alerting philosophy and reliability of the function. In this way, the overall suitability of the RCAF on the flight deck could be assessed. Accordingly, an experiment was designed to assess the performance of the RCAF. This was then used in the evaluation programme carried out at Cranfield University using Cranfield's Large Aircraft Flight Simulator. This simulator is based on a Boeing 747-200 procedural trainer that has been highly modified to accommodate a glass cockpit, Airbus-style sidesticks (whilst also retaining the Boeing 747-200 control columns), Airbus A320-style Flight control unit and radio panels. The simulator software was designed in-house at Cranfield University and utilises Multigen-Paradigm's Vega software for primary image generation. A SEOS projection system using collimated optics affording 180º horizontal and 40º vertical fields of view provides for critical depth of field cueing and excellent cross-cockpit viewing. Such features are critical for the immersive environment required for such evaluation programmes. The in-house designed software also provides the necessary flexibility to support the evaluation of avionics functions such as the RCAF and to allow easy customisation and upgrades that may be necessary for the evaluations.

In this experiment, 14 carefully selected scenarios involving a runway incursion generated by third party traffic (referred to as the 'target') were designed [4] and used in the evaluations. A number of scenarios involved the generation of an alert in the cockpit as a result of a taxying aircraft crossing the hold-short bars to the runway at a particular instance in time. Due to the nature of take-off and landing manoeuvres, these scenarios were required to generate an alert at specific speed regimes with tightly controlled leeways in terms of time for crew to react to avoid a collision. In this way, critical situations could be simulated in a repeatable manner.

A significant complication is introduced by the fact that crews 'flying' the simulator are given freedom of manoeuvre. For example, in take-off, they are left completely free to carry out a rolling take-off. Consequently, if the leeways are to be tightly controlled and repeatable between runs and different test subjects (pilots performing the scenario during the evaluation), it is necessary to control the movement of the target, nominally taxying towards the hold-short bars, with respect to the ownship. For example, it would be necessary to slow down the target if the acceleration of the ownship during take-off is lower than expected, so that the conflict would still occur at the right moment. Clearly, as the target can only enter the runway from a specific taxiway, the moment the alert is generated on the ownship depends on both the ownship's speed and position on the runway for the targeted leeway of the scenario to be respected.

Consequently, it was necessary to develop a rapid prototyping tool specifically to control the motion of traffic and to support the scenarios designed for the evaluation of the RCAF. The tool developed is organised in three main units, namely:

(1) Traffic Motion Generator: This unit generates the trajectories (flight paths) that will be followed by the simulated traffic movements.
(2) Scenario Definition and Control Unit: This unit defines the runway conflict scenarios. It makes use of the trajectories created with the Traffic Motion Generator.
(3) Dynamic Traffic Control Unit: This unit dynamically controls the speed of the target according to the kinematics of the ownship to maintain the intended leeway in the scenario conflict.

The rest of the paper is organised as follows: Section 2 describes the Traffic Motion Generator and gives examples of profiles that can be generated; Section 3 describes the Scenario Definition and Control Unit and gives examples of scenarios that can be simulated; Section 4 discusses the Dynamic Traffic Control Unit and Section 5 presents the results of testing of this functionality; Section 6 presents some conclusions.

2 Traffic Motion Generator

2.1 User Input and Trajectory Definition

The flight path of the target is defined by the user using XML, as this provides a very convenient method of structuring data. Also, XML files can be modified using any text editor without having to recompile any code.

To create a new path, the user first specifies the runway around which the path is defined. Then, the initial position and speed of the target are entered, followed by one or more waypoints. For each waypoint, the user can specify the acceleration, rate of change of acceleration, maximum speed, and turn radius. Using this tool, the user can thus define any path in 3D. Positions are referenced to the axes of a Runway Coordinate Frame (RCF). The origin of this frame is located at the centre of the piano lines of the runway as shown in Fig. 1. The y-axis is along the length of the runway and the x-axis is along the width of the runway. The z-axis is perpendicular to the runway surface.

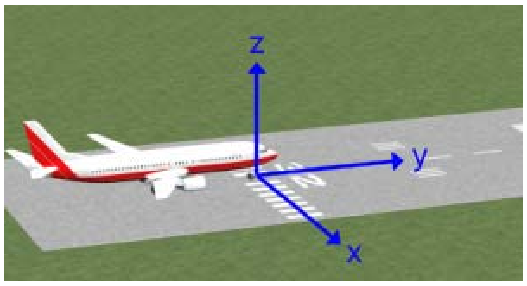


Fig. 1 Runway Coordinate Frame

2.2 Equations of Motion

Once a profile is defined in XML, custom code is used to load the profile and interpolate the position of the target between the specified waypoints. The latitude and longitude of the runway origin are known and this allows the position of the target to be converted from the RCF to geographic coordinates. Apart from position, the program also calculates all the other information that is transmitted in an ADS-B data packet, such as altitude, airspeed, track and vertical speed. This information is calculated at 25Hz and is stored in a spreadsheet file, which is then accessed and altered at run-time by the Dynamic Traffic Control Unit.

In order to find the position of the target between waypoints, equations of motion assuming linear acceleration are used. This was sufficient for the scope of RCAF testing. Assume that the target is moving from waypoint A to waypoint B (Fig. 2). At each time step *t*, the new position vector $\boldsymbol{p}_t$ is found by applying Equations 1-4.

$$\|\boldsymbol{a}_t\| = \|\boldsymbol{a}_{t-1}\| + \dot{a} \times \Delta t \quad (1)$$

$$\boldsymbol{v}_t = \boldsymbol{v}_{t-1} + \boldsymbol{a}_t \times \Delta t \quad (2)$$

$$\boldsymbol{s}_t = \boldsymbol{v}_t \times \Delta t + 0.5 \times \boldsymbol{a}_t \times \Delta t^2 \quad (3)$$

$$\boldsymbol{p}_t = \boldsymbol{p}_{t-1} + \boldsymbol{s}_t \quad (4)$$

where
a is the acceleration vector
v is the velocity vector
s is the displacement vector
Δt is the time interval (40ms)

The direction of these vectors is given by the direction of the vector joining the two waypoints (Fig. 2).

After each position update, the distance between the aircraft and waypoint B is found. The aircraft is assumed to have reached the waypoint when the distance from it falls below a specific threshold. The target speed is also monitored to ensure that it does not exceed the user-defined limit.

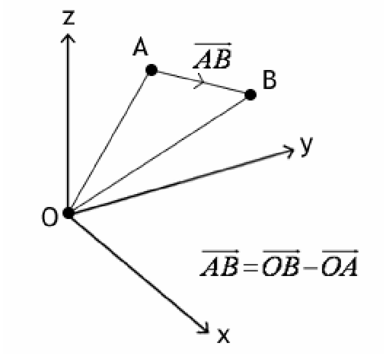


Fig. 2 Vector between waypoints

2.3 Profile Smoothening

A target trajectory is defined using waypoints. This nominally results in a piecewise linear path. In order to smoothen the trajectory profile, the program automatically replaces corners with arcs. Fig. 3 illustrates this process. W_1, W_2 and W_3 are waypoints defined in the trajectory. The process defines new waypoints W_4 and W_5 that are located at a distance d from W_2 (d is the turn radius defined in the XML file). The position vector of W_4 is found using Equation 5.

$$\overrightarrow{OW}_4 = \hat{W}_{21} \times d + \overrightarrow{OW}_2 \tag{5}$$

where
$\hat{W}_{21}$ is the unit vector of the line joining W_2 and W_1
d is the distance between W_2 and W_4
$\overrightarrow{OW}_2$ is the position vector of W_2

The position vector of W_5 is found in a similar way.

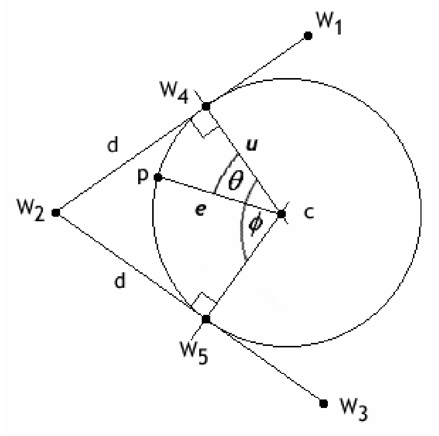


Fig. 3 Replacing a corner with an arc

The centre of the turn is given by the point of intersection of three planes: (1) the plane passing through W_4 and perpendicular to line W_2W_1, (2) the plane passing through W_5 and perpendicular to line W_2W_3 and (3) the plane containing W_1, W_2 and W_3. The first plane is given by Equation (6) and the planar coefficients $A1$, $B1$, $C1$ and $D1$ are given by Equations 7-8.

$$A1x + B1y + C1z = D1 \tag{6}$$

$$(A1, B1, C1) = \overrightarrow{W_2W_1} \tag{7}$$

$$D1 = A1 \times W_4(1) + B1 \times W_4(2) + C1 \times W_4(3) \tag{8}$$

where
$\overrightarrow{W_2W_1}$ is the vector joining W_2 and W_1

The coefficients of the second plane ($A2$, $B2$, $C2$, $D2$) are found in a similar manner. The line of intersection $\boldsymbol{l}$ between the first two planes is given by Equation 9.

$$\boldsymbol{l} = \boldsymbol{a} + t \times \boldsymbol{v} \tag{9}$$

where
$\boldsymbol{a}$ is the position vector of a point on the line
$\boldsymbol{v}$ is the direction vector of the line
t is a scalar which can be varied to obtain any point on the line

$\boldsymbol{v}$ and $\boldsymbol{a}$ are found using Equations 10-13.

$$\boldsymbol{v} = \overrightarrow{W_2W_1} \otimes \overrightarrow{W_2W_3} \tag{10}$$

$$\boldsymbol{a}(1) = 0 \tag{11}$$

$$\boldsymbol{a}(2) = \frac{D1 \times C2 - D2 \times C1}{B1 \times C2 - B2 \times C1} \tag{12}$$

$$\boldsymbol{a}(3) = \frac{D2 - B2 \times \boldsymbol{a}(2)}{C2} \tag{13}$$

where
$\overrightarrow{W_2W_3}$ is the vector joining W_2 and W_3

The point of intersection between $\boldsymbol{l}$ and the third plane is the centre of the turn $\boldsymbol{c}$ and is given by Equation 14. The value of t at the intersection is given by Equation 15.

$$\boldsymbol{c} = \boldsymbol{a} + t \times \boldsymbol{v} \tag{14}$$

$$t = \frac{-A3 \times \boldsymbol{a}(1) - B3 \times \boldsymbol{a}(2) - C3 \times \boldsymbol{a}(3) - D3}{A3 \times \boldsymbol{v}(1) + B3 \times \boldsymbol{v}(2) + C3 \times \boldsymbol{v}(3)} \tag{15}$$

where
$A3$, $B3$, $C3$ and $D3$ are the coefficients of the third plane

The circle that passes through $\boldsymbol{c}$, W_4 and W_5 is given by Equation 16.

$$\boldsymbol{p} = R \times \cos\theta \times \hat{\boldsymbol{u}} + R \times \sin\theta \times \boldsymbol{n} \otimes \boldsymbol{u} + \boldsymbol{c} \tag{16}$$

where
$\boldsymbol{p}$ is the position vector of a point on the circle

R is the radius of the circle
$\boldsymbol{n}$ is a vector perpendicular to the plane containing the circle
$\boldsymbol{u}$ is the vector joining $\boldsymbol{c}$ and W_4
$\boldsymbol{e}$ is the vector between $\boldsymbol{c}$ and a point on the circle
θ is the angle between $\boldsymbol{u}$ and $\boldsymbol{e}$

When θ=0, $\boldsymbol{p}$ is equal to $\overrightarrow{OW_4}$. When $\theta=\phi$ (Fig. 3), $\boldsymbol{p}$ is equal to $\overrightarrow{OW_5}$. ϕ is given by Equation 17.

$$\phi = 180 - \cos^{-1}\left(\frac{\overrightarrow{W_2W_1} \bullet \overrightarrow{W_2W_3}}{\left\|\overrightarrow{W_2W_1}\right\|\left\|\overrightarrow{W_2W_3}\right\|}\right) \qquad (17)$$

Hence, different points along the arc can be found by varying θ. The angular rate $\dot{\theta}$ depends on the aircraft speed. The speed during a turn is kept constant and is equal to the speed at the beginning of the turn. $\dot{\theta}$ is given by Equation 18.

$$\dot{\theta} = \frac{speed}{R} \qquad (18)$$

Fig. 4 shows a typical 3D trajectory generated with the Traffic Motion Generator. This profile shows the aircraft taking off, climbing, doing a circuit and landing at the same runway. It can be observed that the corners have been smoothened out.

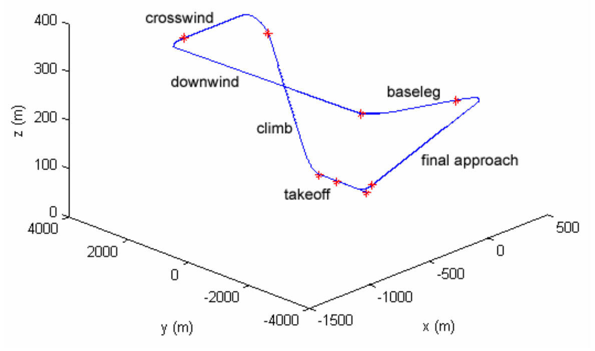


Fig. 4 3D trajectory

3 Scenario Definition & Scenario Control

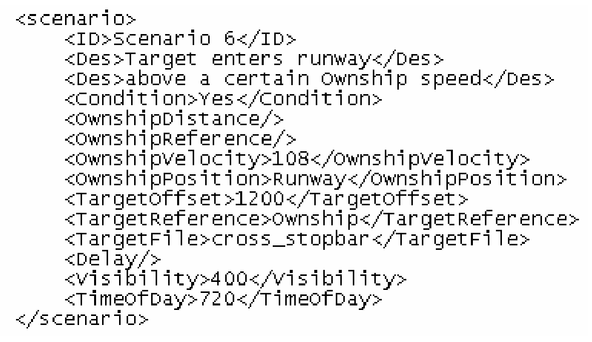
3.1 Scenario Definition

As with the Traffic Motion Generator, the scenarios are defined in XML format. Each scenario is given a unique ID and its definition begins with a brief description of the scenario. Then, the user can specify a number of conditions that have to be satisfied before the target starts moving. The conditions that can be specified are:

(a) Ownship speed – triggered when the ownship exceeds a user-specified threshold.
(b) Ownship position – triggered when the ownship lands or enters the runway.
(c) Ownship distance – triggered when the distance between the ownship and a particular reference point is less than a user-specified threshold. This reference point can be (1) the runway threshold (2) the piano lines or (3) the target traffic.
(d) Delay – This is used in conjunction with the above conditions. When the primary conditions are met, the program waits for a further time delay before the target starts moving.

When the scenario conditions are satisfied, the target starts moving along a certain pre-defined path. The user determines the path to be followed by entering the name of a target trajectory file in the scenario definition. At run-time, the path can be either left in its original position or it can be shifted along the length of the runway with respect to a reference point. This point can be (1) the ownship (2) the piano lines or (3) the touchdown point. Finally, the user can specify two other parameters: visibility (m) and time of day (minutes). These parameters are used to interface with the simulation environment of the Cranfield Large Aircraft Flight Simulator to set the runway visual range and time of day accordingly.

Using this tool, the user can thus define a whole set of scenarios related to runway manoeuvres. Fig. 5 presents an XML script example. In this scenario, the target is programmed to cross the runway when the ownship speed exceeds 108kts. The trajectory file *cross_stopbar* is a recording of the target crossing the stop-bar and entering the runway near the piano lines. However, in this scenario, the target trajectory will be offset 1200m in front of the ownship.



```
<scenario>
    <ID>Scenario 6</ID>
    <Des>Target enters runway</Des>
    <Des>above a certain Ownship speed</Des>
    <Condition>Yes</Condition>
    <OwnshipDistance/>
    <OwnshipReference/>
    <OwnshipVelocity>108</OwnshipVelocity>
    <OwnshipPosition>Runway</OwnshipPosition>
    <TargetOffset>1200</TargetOffset>
    <TargetReference>Ownship</TargetReference>
    <TargetFile>cross_stopbar</TargetFile>
    <Delay/>
    <Visibility>400</Visibility>
    <TimeOfDay>720</TimeOfDay>
</scenario>
```

Fig. 5 Definition of a runway incursion scenario during take-off

There are certain cases where the user may choose not to specify any scenario conditions. In this case, the target starts moving at the beginning of the scenario, irrespective of the state of the ownship.

3.2 Scenario Control

This is the only part of the tool that interfaces directly with the flight simulator and the RCAF. The user selects one of the available scenarios and the information related to that scenario is loaded from the XML file. During the scenario, the ownship is monitored for position and speed. When the specified conditions are met, data is read from the appropriate target trajectory file and is transmitted to the simulator at 25Hz and to the RCAF (in the form of ADS-B data packets) at 1Hz.

4 Dynamic Traffic Control

With the system discussed in Section 3, a scenario is only controlled up to the point where the target trajectory file is loaded. Once the target starts moving, its motion is completely independent of the ownship state. The Traffic will therefore simply move along its pre-recorded path at a predetermined speed.

The main implication of this method is that a scenario may vary significantly from one run to another because no two take-offs are identical. Also, the ownship would be flown by different pilots with different piloting techniques. For example, consider a scenario where the target is to cross the stop-bar at a distance *d* from the piano lines (that is, from a particular taxiway) when the ownship exceeds 70kts. Since the take-off position and acceleration profile differ from one take-off to another, the exact position and speed at which the ownship needs to be abort the run to stop just before the target will vary. Consequently, the precise moment of alerting cannot be hard-coded in the scenario if repeatability in severity of the conflict is to be achieved.

This problem can be mitigated by having the target interact with the ownship throughout the scenario. The target speed is controlled dynamically depending on the current ownship states (position and velocity). In this way it is possible to control the moment when an alert is generated whilst maintaining repeatability in the leeway allowed for the pilot to react and bring the ownship to a standstill in order to avert a collision. The following sub-sections explain how dynamic target control is achieved in a take-off scenario.

4.1 Aborted Take-off Scenario

Fig. 6 shows a runway incursion scenario about to occur at a critical point during a take-off manoeuvre when the target (red aircraft) crosses the hold-short bars. The ownship (yellow aircraft) is proceeding down the runway. At some point during the take-off run, the target crosses the stop-bar and an incursion alert is generated by the RCAF. Upon hearing the alert, the pilot is expected to immediately abort the run.

The aim is to have the Dynamic Traffic Control Unit capable of creating a scenario that can reliably cause the RCAF to generate an alert at a particular instant in time such that the ownship can be brought to a complete stop at point *p*, the intersection between the runway and the taxiway (Fig. 6). Point *p* represents the projected collision point. Consequently, stopping the ownship at point *p* constitutes the limiting case scenario for a collision to be avoided, because the pilot can afford no further delay in aborting the run if the collision is to be averted. If the Dynamic Traffic Control Unit is capable of achieving this aim, then it would also be capable of providing any amount of leeway in terms of allowed pilot reaction time from the moment the RCAF generates the alert to still avoid a collision.

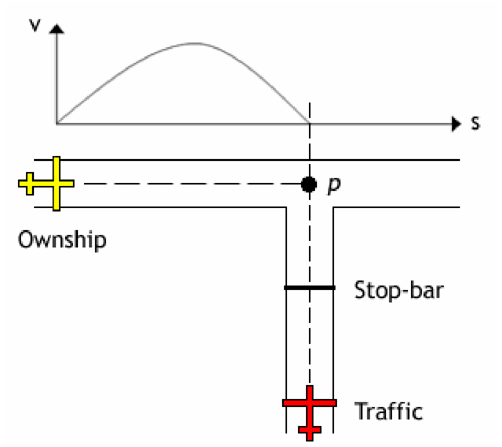


Fig. 6 Runway incursion scenario

To control the reaction time available to the pilot, the speed of the target is modulated according to the speed profile of the ownship. In order to ensure that the ownship stops exactly at *p* in the worst case scenario (when no leeway reaction time is afforded) the following assumptions have to be made:

1. The pilot reacts immediately upon hearing the alert – If the pilot delays to react, the aircraft (ownship) will accelerate further and will require a greater distance to stop. This would result in the ownship overshooting the intersection and a collision would follow.
2. The braking characteristic of the ownship is known *a priori* – In an aborted take-off manoeuvre, the aim of the pilot is to stop the aircraft as soon as possible, thereby applying full braking power. Therefore, whereas there may be variations in the take-off profile (e.g. in the amount of thrust applied), the braking profile is well characterised in the simulated environment during deceleration. Consequently, different profiles for different operating conditions (flap settings, etc) are stored in a database for retrieval at run-time by the Dynamic Traffic Control Unit.

4.2 Traffic Speed Control

Fig. 7 shows the braking characteristic of the ownship (for a flap setting of 20˚) as well as a 6th order polynomial curve that was used to model this braking profile. This fit was obtained using the curve-fitting toolbox in Matlab®. The x-axis is the distance of the ownship from the start of the run and the point where the aircraft is brought to rest represents the intersection point *p* in the critical conflict scenario.

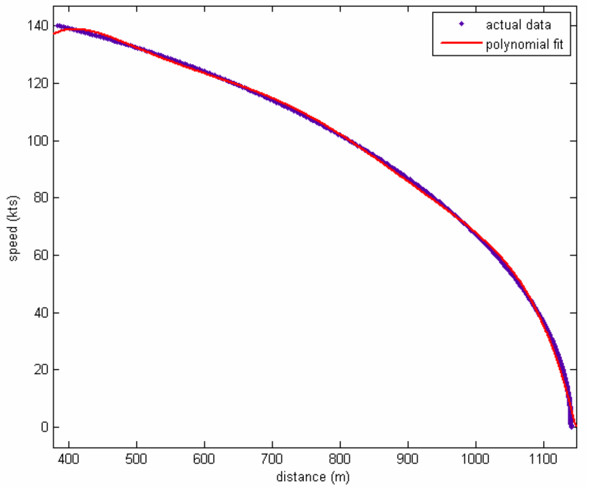


Fig. 7 Ownship braking characteristic

At run-time, the speed of the ownship is continuously monitored by the Dynamic Traffic Control Unit. Fig. 8 shows the ownship speed profile up to a distance s_t from the start of the run. By extrapolating this profile, it is possible to determine the point of intersection *i* between the acceleration phase and braking phase profiles. This point defines the moment at which the run is to be aborted (following the generation of the RCAF alert) so that the ownship can be brought to a standstill at *p*.

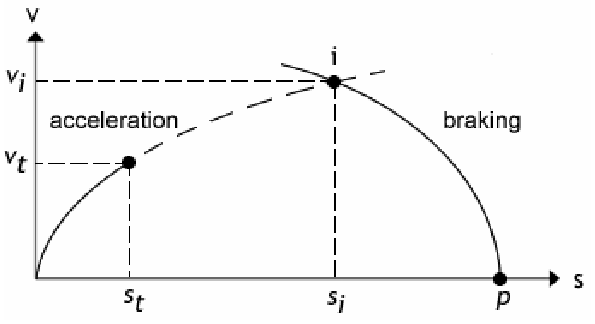


Fig. 8 Intersection between acceleration and braking phase profiles

At each time step, the past speed-distance values of the ownship are normalised and modeled by a 4th order polynomial given by Equation 19.

$$f(x) = p_1x^4 + p_2x^3 + p_3x^2 + p_4x + p_5 \qquad (19)$$

where
x is the distance from the start of the run (nominally at the piano lines)
$f(x)$ is the speed

The polynomial coefficients are found using Linear Least Squares techniques as shown in Equation 20.

$$\begin{pmatrix} n & \Sigma x & \Sigma x^2 & \Sigma x^3 & \Sigma x^4 \\ \Sigma x_i & \Sigma x^2 & \Sigma x^3 & \Sigma x^4 & \Sigma x^5 \\ \Sigma x^2 & \Sigma x^3 & \Sigma x^4 & \Sigma x^5 & \Sigma x^6 \\ \Sigma x^3 & \Sigma x^4 & \Sigma x^5 & \Sigma x^6 & \Sigma x^7 \\ \Sigma x^4 & \Sigma x^5 & \Sigma x^6 & \Sigma x^7 & \Sigma x^8 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{pmatrix} = \begin{pmatrix} \Sigma f(x) \\ \Sigma x f(x) \\ \Sigma x^2 f(x) \\ \Sigma x^3 f(x) \\ \Sigma x^4 f(x) \end{pmatrix} \qquad (20)$$

The extrapolation of the 4th order polynomial cannot be used to provide a reliable estimate of the intersection point i. As a result, a linear extrapolation is used instead, as this ensures a monotonic increasing profile. Whilst such a solution does not provide a very accurate estimate at early stages of the run, the error in the estimate falls to zero as the run progresses and the amount of extrapolation required reduces accordingly. The gradient of the slope is found by differentiating the 4th order polynomial at the most recent speed-distance point (v_t, s_t). Point i is given by the intersection between this line and the braking curve and is determined by solving Equation 21.

$$p_1x^6 + p_2x^5 + p_3x^4 + p_4x^3 + p_5x^2 + p_6 + p_7 = mx + c \qquad (21)$$

Once i is known, it is possible to estimate the time t_i at which the ownship will reach this point. To do this, a polynomial is first fitted on the speed-time history characteristic of the ownship in a similar way to the speed-distance profile. Then, the gradient of the curve at the last speed-time value is found and this gradient is extrapolated to obtain an estimate of the time t_i at which the ownship will achieve velocity v_i.

The time at which the ownship must reach point i is also the time that the target must cross the stop-bar to trigger the alert (ignoring crew response time). Thus, the target speed is controlled using Equation 22.

$$s_t = \frac{d_s}{t - t_r} \qquad (22)$$

where
s_t is the target speed
d_s is the remaining distance between the target and the stop-bar
t is the remaining time to cross the stop-bar
t_r is the reaction time available

If it is required that the ownship stops exactly at the intersection between the runway and taxiway, t_r is set to 0; otherwise, it can be increased to provide any amount of reaction time. In practice, the minimum value of t_r is greater than 0 for two reasons:

1. To allow for the standard pilot reaction time. Even if the pilot reacts immediately to the alert, a delay is still incurred before the ownship starts to decelerate because of the time it takes the crew to reduce thrust and activate the braking devices. The average time from alert to deceleration has been calculated over a number of aborted take-off manoeuvres and is about 0.7s for the Cranfield Large Aircraft Flight Simulator.
2. The update rate of the RCAF is 1Hz. Therefore, in a worst-case scenario, the

RCAF will detect a runway incursion 1s after the target crosses the stop-bar.

Thus, the minimum value of t_r is 1.7s.

Speed control of the target aircraft is achieved by retrieving and altering the sequence of the data stored in the spreadsheet file generated by the Traffic Motion Generator. The Traffic Motion Generator synthesises the motion of the target according to the defined path with the target moving at 0.5m/s. Positional data is stored every 40ms (25Hz). Consequently, speed variations in multiples of 0.5m/s can be synthesised by the Dynamic Traffic Control Unit by skipping a specific number of successive entries in the database. Whilst this would still result in an update rate of 25Hz, it would accommodate speed control of a target in discrete steps of 0.5m/s. A smaller step size can be achieved by recording the data at slower speeds on the spreadsheet, but a sensitivity of 0.5m/s was found to be adequate for the application.

5 Results

Testing of the rapid scenario generation tool took place on Cranfield University's Large Aircraft Flight Simulator running a simulated environment of Bristol airport (Fig. 9). During testing, all the simulator parameters were recorded in a log file.

As dynamic traffic control should work irrespective of the ownship's take-off profile (take-off thrust setting, take-off position, etc.), the rapid scenario generation tool described was tested in each of the scenarios given in Table 1. In each scenario, the conflict was created at the intersection with Taxiway Charlie. Each scenario was repeated four times using pilots to introduce variations between runs. This gave a dataset of 32 runs. In the first four scenarios, the ownship enters the runway from Taxiway Alpha (Fig. 9), whereas in the other scenarios it enters the runway from Taxiway Bravo. The distance between starting positions A and B is 225m. In each of these scenarios, t_r was set to its minimum value (1.7s); therefore, the ownship should theoretically have been brought to rest at the intersection between the runway and Taxiway Charlie.

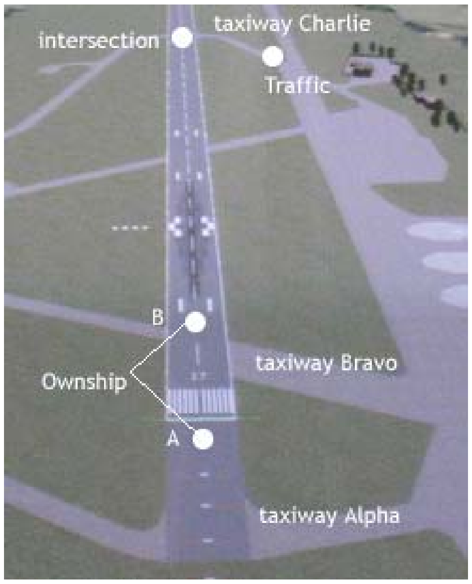


Fig. 9 Bristol runway

Table 1 Dynamic traffic control scenarios

Scenario	Starting Position	Take-off
1	A	Rolling
2	A	Full thrust
3	A	1.4 EPR
4	A	1.35 EPR
5	B	Rolling
6	B	Full thrust
7	B	1.4 EPR
8	B	1.35 EPR

Table 2 summarises the results obtained. It can be observed that the actual ownship speed at point *i* (maximum speed) was always slightly lower than the predicted speed and the difference was within 2kts in all cases. Similarly, the actual ownship position at *i* was consistently closer to the piano lines than expected and the difference was less than 60m in all cases. The position of the ownship at standstill was always within 45m of the intersection between the runway and Taxiway Charlie.

Table 2 Results of dynamic traffic control

Sce	Intersection speed (kts)			Intersection position (m from piano lines)			Final position (m)			Delays (s)				
	predicted	actual	diff.	predicted	actual	diff.	ideal	actual	diff.	RCAF	reaction	total	t_r	diff.
1	108.5	108.0	-0.6	678	655	-23	1140	1150	10	1.13	0.58	1.71	1.7	0.01
2	109.3	107.6	-1.6	674	615	-59	1140	1107	-33	0.84	0.64	1.48	1.7	-0.22
3	103.6	102.7	-0.9	713	668	-45	1140	1097	-43	0.98	0.6	1.58	1.7	-0.12
4	99.3	99.1	-0.2	740	713	-27	1140	1100	-40	0.97	0.85	1.82	1.7	0.12
5	97.9	97.1	-0.7	747	737	-10	1140	1140	0	1	0.62	1.62	1.7	-0.08
6	99.1	98.3	-0.8	740	715	-26	1140	1128	-12	0.98	0.62	1.6	1.7	-0.1
7	93.8	93.3	-0.5	770	751	-19	1140	1107	-33	0.97	0.67	1.64	1.7	-0.06
8	90.0	89.8	-0.2	788	777	-11	1140	1097	-43	0.98	0.71	1.69	1.7	-0.01

There are several reasons for the ownship not stopping exactly at the intersection. These include:

- The actual delay due to pilot reaction time and RCAF update will vary from the allowed 1.7s
- The actual braking characteristic may differ slightly from the modeled braking profile, particularly due to reaction time in applying the brakes
- The polynomials used to model the speed-distance and speed-time profiles may not be sufficiently accurate

Figure 10 shows the speed profile of the ownship for one particular scenario. As expected, there is a slight delay before the ownship decelerates when an incursion alert is sounded. Figure 11 shows how the prediction of ownship speed and position at *i* varies throughout the scenario until the target crosses the stop-bar. The predictions become more accurate as the take-off progresses because, as the run progresses (Fig. 12), the tangent of the curve has a progressively flatter slope, thus resulting in a progressively lower estimate of intersection speed that is closer to the actual value eventually achieved.

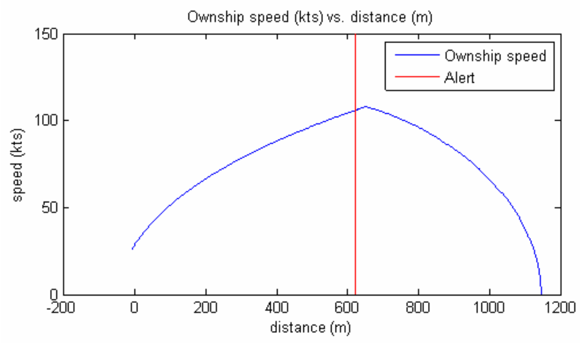


Fig. 10 Speed profile of ownship

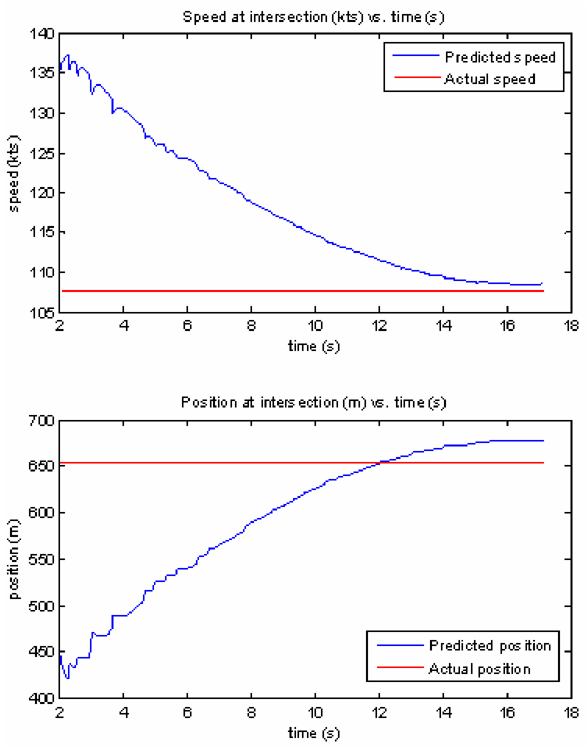


Fig. 11 Prediction of intersection point

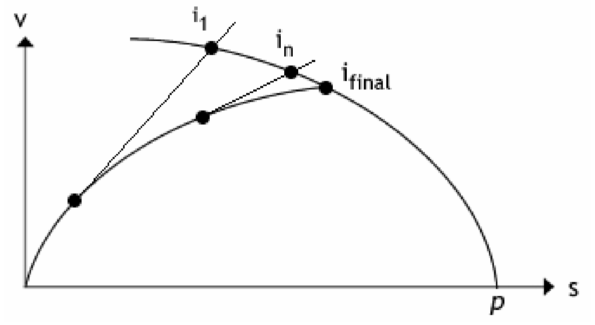


Fig. 12 Accuracy of intersection point prediction

Figure 13 shows how the target speed varies with time. It is observed that the speed increases as the run progresses during the scenario. This is because the predicted ownship speed at *i* decreases as the run progresses and, when fitted into the velocity-time profile to obtain the time to intersection, this results in a progressively smaller time-to-go prediction. Hence the target needs to be accelerated to ensure that it is not late in reaching the hold-short bar to cause the runway incursion and generate the alert.

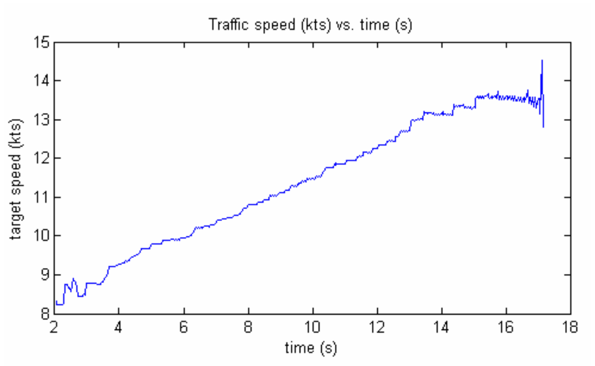


Fig. 13 Variation of target speed

6 Conclusion

A rapid scenario generation tool has been presented. This tool enables the generation of a wide range of traffic trajectories in 3D as well as the simulation of traffic conflicts in flight, such as runway incursions.

The tool provides flexibility and repeatability in scenario execution. Flexibility is provided by the intuitive definition of trajectories and scenarios in XML format, whilst repeatability is obtained through the capabilities of the Dynamic Traffic Control Unit.

Although the performance of this rapid scenario generation tool has proven to be fit for purpose, it can be further developed to enable it to better control the target speed through improved conflict time prediction and to support the simulation of more complex traffic conflicts, such as those involving more than two aircraft.

7 References

[1] *EU-Flysafe* (WWW document). http://www.eu-flysafe.org (accessed May 2008).

[2] Sammut A, Zammit B and Zammit-Mangion D. A traffic surveillance function and conflict detection method for runway manoeuvres. *7th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, Belfast, Northern Ireland, 2007.

[3] *ADS-B* (WWW document). http://www.ads-b.com (accessed May 2008).

[4] Szasz S, Gauci G, Zammit-Mangion D, Zammit B, Sammut A and Harris D. Design of experiment for the pilot evaluation of an airborne runway incursion alerting system. *The 26th Congress of the International Council of the Aeronautical Sciences*, Anchorage, Alaska, 2008.

Copyright Statement