Development and Validation of an ATC Research Simulator

Tomislav Radišić, PhD
Doris Novak, PhD
Biljana Juričić, PhD
Department of Aeronautics
Faculty of Transport and Traffic Sciences, University of Zagreb
Zagreb, Croatia
tradisic@fpz.hr

Abstract—**ATM research and development relies heavily on simulation methods. For studies involving human factors, real-time human-in-the-loop simulations provide the most reliable results. From the perspective of a researcher, these types of studies are often also the most complex to perform. One of the issues researchers face is the lack of suitable research simulators that can be freely modified to perform in the desired manner. Commercial simulators are mostly produced for training and each upgrade, especially development of custom modules, is quite costly, sometimes even prohibitively so. For this reason, in this paper the process of research simulator development will be presented, from the definition of simulator requirements to simulator validation and operation. Some of the key technologies will also be presented along the way. The simulator presented here was built and used to examine the effect of trajectory-based operations on air traffic complexity in en-route sectors. Authors believe that although this design is not generic enough to be used for all purposes, there is still a large number of research topics that can be examined with such simulator. Furthermore, methods and solutions presented in this paper can also be applied to other simulator designs.**

Keywords - ATC; simulation; real-time; human-in-the-loop; air traffic complexity

I. INTRODUCTION

Simulation is a core method for ATM research and training, with different purposes requiring different levels of fidelity and simulation scope. Fidelity refers to the level of similarity between the simulated environment and the actual operations. Simulation scope can be broadly divided into strategic and tactical simulations. Strategic simulation tools (e.g. EUROCONTROL’s NEST) are used to analyze current and forecast future ATM situation on a global level. On the other hand, tactical simulation tools are used to accurately simulate ATC operations on a sector level (e.g. ATCoach by UFA or Micronav’s BEST Radar Simulator). Whatever the purpose, research teams have several ways to acquire the required simulation research tools. Large organizations such as NASA or EUROCONTROL are able to develop and maintain their own ATC research simulator centers. Smaller research groups have to use commercial ATC simulators which are very high fidelity but not easily customizable or develop their own purpose-built simulators with limited features and fidelity. Another option is to use a third-party open-source ATC simulator such as the one developed at the University of Queensland [1] with all its limitations.

This paper presents a set of methods and tools that can be used to develop a custom research ATC simulator. Advantages of a purpose-built simulator are: complete control of the features developed for the specific research task, ability to develop the simulator to the desired standard of fidelity and scope, better understanding of the simulator operations, lower cost, and possibility of future upgrade. Disadvantages, of course, should also be considered and they include: need for expert programming knowledge, need for deep understanding of ATC operations and tools, and spending time that could otherwise be used to do actual research.

The ATC research simulator presented in this paper was developed to study air traffic complexity on a tactical (sector) level. Research objective was to determine the effect trajectory-based operations (TBO) have on air traffic complexity. The scope of the study was limited to nominal en-route operations with no extreme weather conditions. It is presented here as an example of how the simulator requirements were shaped by research objective and how those requirements were implemented in a way which enables future development and growth.

II. SIMULATOR REQUIREMENTS

Simulator requirements can be divided into two groups – general and specific. General simulator requirements are those that are independent of simulator purpose. They are commonly accepted as best practice for most forms of application development. In this project, following general requirements for the simulator development were set: reliability (decreased likelihood of simulation failure, robust exception handling), maintainability (simple code structures, standard naming conventions, modular design, documented code, testability), efficiency (network, disk, and memory management, code optimization), extensibility (loosely coupled modules), and portability (external configuration) [2].

To define specific simulator requirements, the researcher will have to provide answers to questions regarding the purpose and the aim of the research that will be conducted. These are: Is fast-time or real-time simulation needed? How accurate should the aircraft model be? Should generic or actual aircraft types be used? How representative of the real working environment should the simulator be? What hardware is needed? Will there be a pseudo-pilot or will the ATCOs do everything by themselves? What type of communication will there be: voice, datalink, or both? Is it necessary to be able to simulate failures? How detailed should the weather and surveillance models be?

With these questions and general requirements in mind, the following specific requirements for the development of the ATC simulator have been set according to the aims of the study:

Real-time human-in-the-loop simulation. For high-fidelity ATC simulation it is necessary to include the actual ATCOs in the simulation. This means that the simulation will have to be performed in real-time and that the working environment will have to be as similar as possible to the real working environment.

Accurate and versatile aircraft model. It was determined that the aircraft model used in this research had to satisfy following criteria: ability to model more than 95% of aircraft types flying in Europe, have accurate aircraft climb and descent profiles, have realistic turn performance, realistically model aircraft performance and limitations, have reasonably accurate 3D/4D FMS algorithms. This enables usage of the actual traffic data without the need to exclude or substitute aircraft types. Also, accurately modelled turns and vertical profiles enable simulating terminal operations.

Realistic working environment. It had to be similar to the actual working environment to which the ATCOs are used to. This includes the layout of the radar screen, auxiliary screens, keyboard, mouse, and communication switches. User interface had to be similar to the existing ATC simulators and workstations to give the ATCOs a smooth transfer to the simulator (without extensive training). In this project, working environment was adjusted to resemble actual working environment that the participants were used to. For some other purposes, a generic simulator layout could be useful.

Representative ATC tool operation. For this research a limited set of ATC tools had to be developed. It was not necessary to develop a complete set of professional ATC tools because this study consisted of a limited set of simulation scenarios and traffic situations. Those ATC tools that were developed though, needed to function in a manner that is representative of the actual tools. Also, new tools could easily be added due to modular design of the simulator. Required tools were: map tools, display tools, range and bearing lines, level filter, SSR code filter, separation tool, area proximity warning, short-term conflict alert, separation infringement alert, route/trajectory display, flight profile display, strip-less flight progress monitoring tools, datalink interface, velocity vectors, and flight trails.

Ability to record all necessary data. Since the primary purpose of this simulator was research, it was important to implement the function to record as much data as possible. Data that had to be saved were: complete aircraft states (trajectory, heading, TAS, mass, thrust, pitch, bank, fuel flow etc.), human-machine interactions (mouse and keyboard events), voice communications, and radar screen images.

Easy data editing. Medium and high fidelity ATC simulations are based on actual airspace configurations or sufficiently complex generic instances thereof. Simulator had to enable easy configuration of all airspace-defining data and quick switching between different airspace configurations. Additionally, simulation scenarios had to be created and edited, therefore, the data editor had to allow quick and easy scenario creation and updating. In this project, a data pipeline was established for feeding actual historic flight plan data into the simulator, thus automatically generating traffic for scenarios.

Voice and data link communication. Some ATC simulators allow the air traffic controller to directly change the aircraft state variables such as heading, level or speed. However, ATC simulators aiming at high(er) fidelity have to adopt the approach that more closely mimics the actual ATC operations. This means that ATCO has to issue instructions to the pilot either via voice communication or data link and it is pilot's job to follow those instructions (or, in the case of simulation, it is pseudo-pilot's job to do so). This type of operations is very important in studies examining capacity, workload or complexity, because the communication tasks make a substantial fraction of controller taskload and in some cases they even limit the sector capacity. In this study a commercial VoIP solution was used for voice communication and data link was implemented in the simulator itself.

Local Area Network operations. Since the controller's and pseudo-pilot's stations need to exchange data in real-time, some form of communication was needed between the two stations. In this study the communication was limited to local area network which somewhat simplified the development due to high bandwidth and insignificant lag. For remote operations (over Internet) special care must be taken to reduce the bandwidth requirements and lag.

Simple meteorological model. While weather phenomena are very complex and diverse and have a profound impact on flight operations, for ATC simulation a simplified model is adequate for most purposes. Arguably the most important weather phenomena, in nominal operations, from the perspective of an air traffic controller are wind, thunderstorms, icing, and turbulence [3]. For studies dealing with the weather more specifically, a more advanced model should probably be used.

Support for TBO. The simulator used in this research had to be able to support trajectory-based operations (TBO). TBO support consisted of generating 4D trajectories, simulating aircraft flying 4D trajectories, and displaying those aircraft with all additional information (trajectories and flight profiles).

Simple surveillance system model. ATC simulator can be built to accurately represent various surveillance systems, such as radars, ADS-B, and multilateration, and their properties.

This type of surveillance system models are useful when controller's response to partial radar failures or system degradation are studied. In this study nominal operations were studied, therefore only a simple radar system model was needed. Radar targets were updated every 5 seconds with actual aircraft positions with no options for reduced accuracy or precision. Pseudo-pilot had the option of setting the assigned SSR code and squawking IDENT, while the controllers had the option of filtering the traffic according to SSR codes. It is possible to upgrade this model with more features if the need arises.

These specific simulator requirements were used during the development of the ATC research simulator for this study (air traffic complexity assessment). For other types of studies, different specific requirements would have to be defined and met.

III. SIMULATOR FRAMEWORK OVERVIEW

In accordance with the previously discussed simulator requirements, a prototype was developed and it will be presented in this section. Authors believe that although this design is not generic enough to be used for all purposes, there is still a large number of research topics that can be examined with such simulator. Furthermore, methods and solutions presented in this section can also be applied to other simulator designs.

Proposed simulator framework can be broadly separated into two parts: data and application. Each of these two parts has a series of components. Since the first rule of application portability is not having any hard-coded data, all data and configuration was separated from the application (Figure 1, left side). This includes all user interface (UI) labels, tool tips, names, and database headers, which are all stored in the *Settings* file. Database components also include: geographic data (country borders, coastline, elevation model), Base of Aircraft Data – BADA (EUROCONTROL's database of aircraft performance data), weather (3D grid with wind direction and speed, thunderstorm locations and times, icing and turbulence areas), scenarios (determines which sets of geographic, weather, airspace, and flight plan data will be used), flight plans (database of all flight plans, some or all of which are used in a scenario), and airspace (data defining airspace(s), one airspace is used in any scenario).

Next, short description of simulator modules will be presented (Figure 1, middle). Simulator administrator (researcher) uses *Data Editor* module to input or modify the

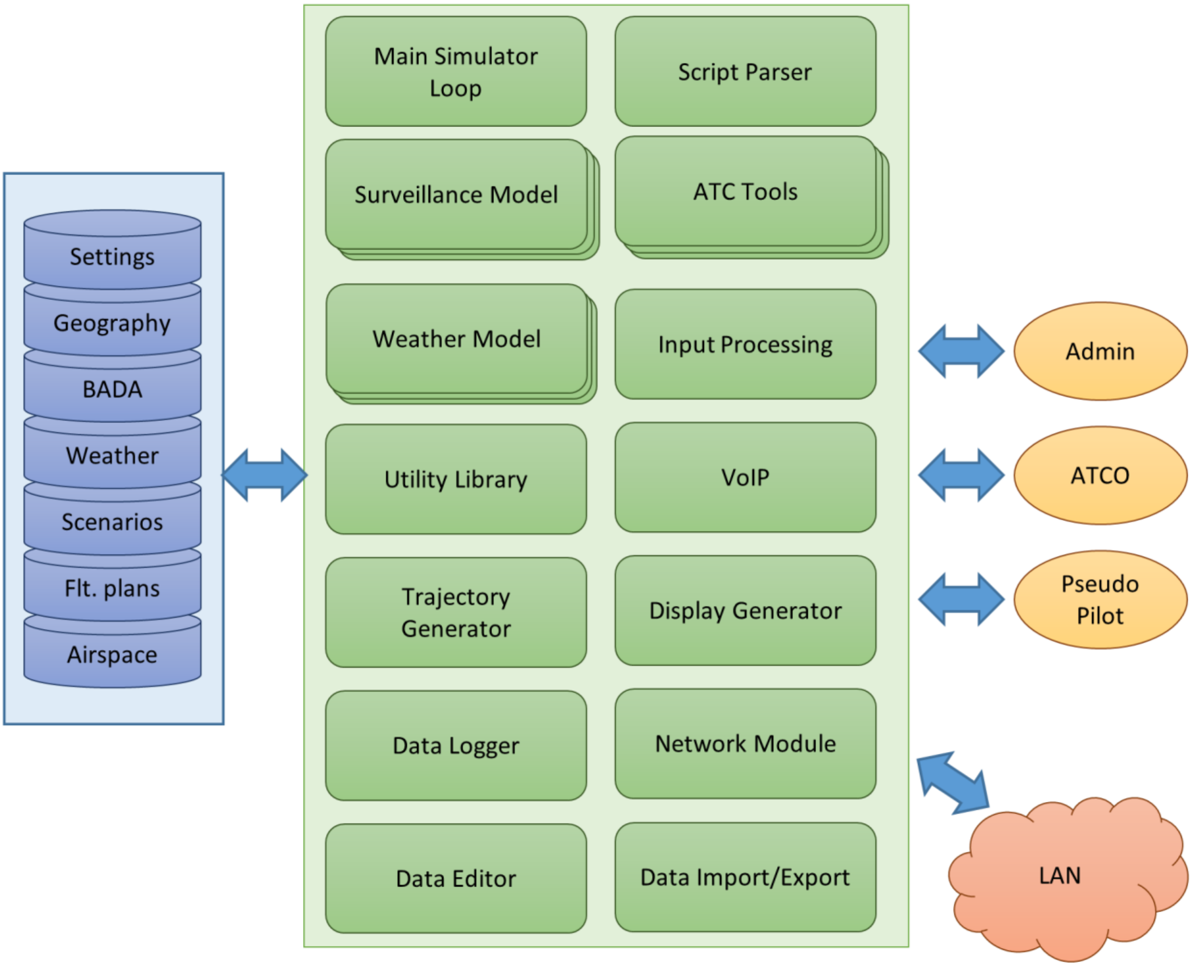


Fig. 1. Simulator Framework

data stored in the databases. For example, in this study Aeronautical Information Publications (AIPs) were used to obtain airspace data, EUROCONTROL’s Demand Data Repository 2 (DDR2) was used for historic air traffic, local meteorological service providers or *meteocentre.com* were used to obtain weather information, and the database of Global Administrative Areas (*www.gadm.org*) was used for country borders and coastlines. Researcher then prepares simulation scenarios which use subsets of this data for the actual simulation runs with addition of scenario script which initiates scenario events (e.g. climb/descent requests, failures etc.).

Main Simulator Loop is responsible for activation of specific modules based on settings and scenario objectives. Multiple versions of some modules (e.g. weather model) can be available for use and depending on the purpose of the simulation, appropriate model will be loaded by the *Main Simulator Loop*. Most modules are started in separate asynchronous threads to prevent one module from pausing others while some longer operation is completed. This module also adjusts the UI and simulator operations for the requested station (ATCO or pseudo-pilot), establishes network connection with the other station (via *Network Module*), and controls the simulator operation (via *Input Processing* module).

Once the simulator is active and the scenario selected, *Data Import/Export* module loads all required data into memory for faster retrieval (in line with efficiency requirement). This data is then used by *Trajectory Generator*, along with data generated by the *Weather Model*, to generate current aircraft positions and their future trajectories. *Display Generator* then renders the radar screen by overlaying radar targets and labels onto the background map created from geographic and airspace data.

IV. CORE SIMULATOR TECHNOLOGY

In this section some of the technology needed for simulator development will be presented. General coding techniques, such as data input/output, parsing, event handling, or multi-threading, will not be covered here in order to save space. The focus will be on three simulator components: aircraft model, workstation hardware layout, and ATC tools.

A. Aircraft Model

Aircraft model is the integral part of the *Trajectory Generator* module and is in fact a hybrid model made of three separate models: aircraft performance model, aircraft dynamics model and flight management system model.

Having considered all requirements mentioned in previous sections, EUROCONTROL's Base of Aircraft Data (BADA) Aircraft Performance Model (APM) was chosen as a starting point for aircraft model. Its main advantages are support for many different aircraft types, easy implementation, and excellent documentation.

BADA, however, provides only for modelling aircraft performance so the models of aircraft dynamics and Flight Management system (FMS) had to be developed from the start. BADA is a database of aircraft data developed and updated by EUROCONTROL Experimental Centre (EEC). As mentioned by [4] the aircraft performance information provided in BADA *'is designed for use in trajectory simulation and prediction in ATM research as well as for modeling and strategic planning in ground ATM operations’*. It provides ASCII files containing operation performance parameters for 405 aircraft types – of these 150 are original and 255 are equivalent aircraft types. Equivalent aircraft types, also known as synonym types, are not covered by one of the BADA files directly, they are linked to one of the 150 original types [5]. For each original aircraft type three files are provided. *Operations performance* file with specific parameters needed to model the performance of that aircraft type. *Airline procedures* file which contains speed schedules for airlines (one default speed schedule is provided for each aircraft type – user can define others). *Performance table* file which provides tabulated TAS, rate of climb/descent, and fuel consumption for each aircraft type at different flight levels. In addition, synonym file (links original and equivalent aircraft types) and global aircraft parameters file are provided.

The kinetic approach to aircraft performance modelling, as used in BADA, seeks to accurately model forces acting on aircraft represented as a single point. Total Energy Model (TEM) is then used to determine the distribution of the work done by these forces towards increase or decrease of aircraft's potential and/or kinetic energy.

As shown in the HYBRIDGE project [6], for ATM simulation purposes, aircraft dynamics can be adequately modelled using a Point Mass Model (PMM). It is the aircraft dynamics system which, based on six state variables (x, y, and z coordinates, TAS, heading, and mass), four inputs (thrust, pitch, bank, and drag), and three disturbances (three wind components), determines the change of aircraft state variables (1). Since three of the six state variables represent aircraft coordinates, output of this system effectively provides aircraft trajectory.

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_4 \cos(x_5) \cos(u_3) + w_1 \\ x_4 \sin(x_5) \cos(u_3) + w_2 \\ x_4 \sin(u_3) + w_3 \\ -\left[\left(\frac{u_4 S\rho}{2}\right) \cdot \left(\frac{x_4^2}{x_6}\right)\right] - [g \sin(u_3)] + \left(\frac{u_1}{x_6}\right) \\ \left[\left(\frac{C_L S\rho}{2}\right) \cdot \left(\frac{x_4}{x_6}\right)\right] \cdot \sin(u_2) \\ -\eta \cdot u_1 \end{bmatrix} = f(x, u, w) \quad (1)$$

This system uses state variables (x), inputs (u) and disturbances (w), along with additional terms such as aircraft total wing surface area, *S*, air density at altitude, ρ, acceleration due to gravity, *g*, aerodynamic lift, C_L, and fuel consumption factor, η, to calculate the change in state variables.

The purpose of the FMS model is to determine how to change inputs in order for aircraft to follow the desired path from the flight plan. The inputs that FMS uses are similar to the inputs that pilots use to control an aircraft.

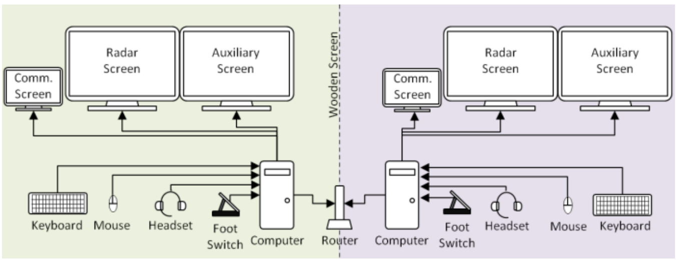
The first thing an FMS must do is to determine the current aircraft position and speed relative to the desired path and speed. Next, it must determine the inputs needed to correct differences between the two. There are however, some differences in control strategies between different phases of flight. For example, the aircraft is controlled differently in climb than in descent, aircraft configuration is different during the approach phase than in cruise flight, different limits on control inputs are enforced during different phases, etc. Due to this, aircraft state is additionally described by a set of discrete variables (e.g. variable *ClimbMode* can have values: *Climb*, *Level*, and *Descent*). For each of the discrete variables (states) a simple finite state machine (FSM), governing conditions under which aircraft changed states, was developed. The values of the variables that are set by finite state machines are used to determine the values of inputs (u) used in (1).

Of the four inputs, pitch angle and thrust are used together to achieve the required TAS and rate of climb/descent. Bank angle is used to move the aircraft towards the desired flight track, and the drag coefficient is set in accordance with the aircraft configuration that is required for the current phase of flight (e.g. gear and flaps are extended during approach phase). A series of limitations is set on the maximum values and maximum rate of change of pitch angle, bank angle, and thrust in order to prevent unusual or overly dynamic maneuvers.

B. Workstations

Humans receive most of the information about their surroundings visually. Radar screen is the main source of the visual information for the air traffic controller. Therefore, the simulator used for this research had to be as representative of the real radar screen as possible.

It must be noted that the term 'radar screen' is slightly misleading in the context of modern air traffic control. Though radars are still the primary source of aircraft position information, modern ATC workstations do not have an actual radar screen. The information provided by the radar is instead heavily filtered and correlated with other information related to that particular aircraft. Because of this, modern 'radar' screens are more akin common computer screens with fairly simple vector graphics than the old analogue radar screens. The main difference between the common commercial electronics computer screen and professional ATC work station screen is the aspect ratio. While computer screens are usually produced in a number of widescreen formats, ATC screens usually have 1:1 aspect ratio (i.e. they are square). Therefore, the case can be made for using commercial off-the-shelf screens to simulate ATC work stations. This approach was used in this research. Hardware layout can be seen in Figure 2.



ATCO Station
Pseudo-pilot Station

Fig. 2. Hardware Layout

Other devices used for human-machine interaction in the context of ATC are keyboard, mouse, and radio communication switch (hand and/or foot operated).

Headphones and microphone are used for radio communication.

For this research, following work station configuration was used:

- One computer screen for radar display.
- One computer screen for additional information (flight plans, meteorological information).
- One touchscreen for central switchboard (telephone and frequency switches).
- Keyboard and mouse for data entry and manipulation
- Headset (headphones + microphone) with hand and foot operated comm. switches.

C. User Interface and ATC Tools

Radar screen interface elements used for this research can be divided into three sections:

- Map with correlated radar targets (aircraft and data labels)
- Tool strip (ATC tools and basic information)
- Control panels (displayed according to controllers actions).

The layout of radar screen display of the simulator developed for this research can be seen in Figure 3. Map is built of individual layers, which represent country borders, Flight Information Region (FIR) borders, coast, restricted airspace zones, navigation points, navigation aids etc. Map can be dragged with mouse and zoomed in/out with mouse scroll button.

Aircraft are displayed as circles with trail of dots representing aircraft's trajectory in the past 30 seconds, and with a line showing its current track vector. The color of the aircraft target changes depending on the state of that aircraft.

Aircraft labels are connected with the appropriate aircraft by solid lines. Labels initially show limited set of flight data; however, they expand on mouse hover to show expanded set of data. Labels are also the main interface between controller and strip-less flight progress monitoring system. By clicking on the aircraft label, the controller can accept the aircraft from the transferring ATC unit, and assign flight level, speed, heading or route according to instructions given to the aircraft (Figure 4). This allows the controller to keep track of the given instructions and to monitor flight's progress. It also makes possible for clearance adherence algorithms to work.

Tool strip is located at the top of the radar screen and houses the following tools (seen at the top of the Figure 3):

- Map re-center – Centers the map on the center of the airspace sector. It is used to quickly return to the main

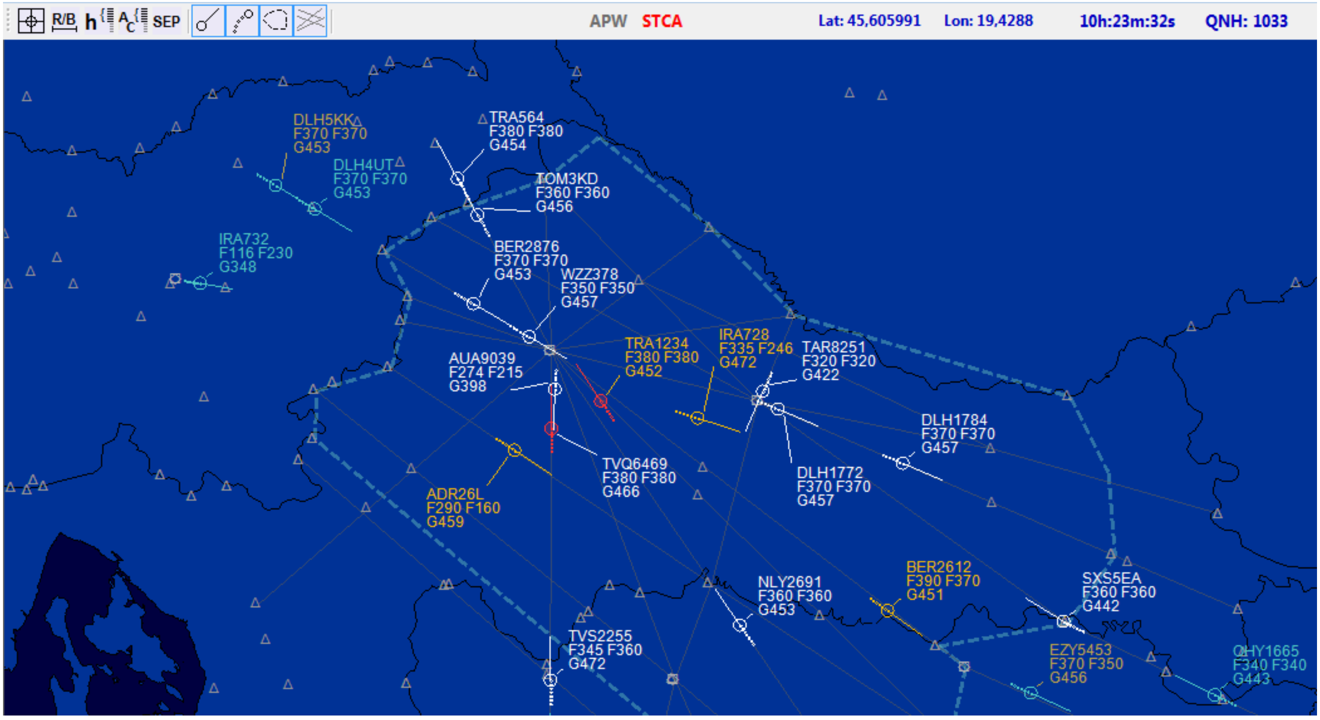


Fig. 3. Main Radar Display

mode of display after zooming in or scrolling to the side.

- Range and bearing line – Measures the distance and range between two points on the map.
- Height filter – Filters the aircraft according to altitude. Filtered out aircraft are displayed as grey aircraft targets without labels.
- SSR code filter – Filters the aircraft according to Secondary Surveillance Radar (SSR) codes.
- Separation tool – Extends the track vectors of two selected aircraft to the point of their closest approach and displays separation distance (in nautical miles) and time until the point of closest approach is reached (in minutes and seconds).
- Display tools – used to adjust four display layers directly from the simulator. These layers are: aircraft track vectors, aircraft trails, sector boundaries, and standard routes. All other display layers are editable through text files.
- Area Proximity Warning (APW) – Activates when an aircraft is about to enter the sector without being accepted or, when an aircraft is about to exit the sector without being transferred to another ATC unit. The ‘APW’ sign starts to flash purple and is accompanied by a single sound alert.
- Short Term Conflict Alert (STCA) – Activates when two aircraft are less than two minutes away from separation minimum infringement. The ‘STCA’ sign starts to flash red and is accompanied by a single sound alert.
- Separation infringement alert – Activates when two aircraft have infringed on the separation minima. The whole tool strip flashes red and aural warning is sounded repeatedly.
- Other data – Latitude/longitude display, simulation time, QNH.

For aircraft flying TBO, an air traffic controller can also see the trajectory profile. This information is displayed in a separate window on the secondary monitor. Flight profile information is used by controllers to separate aircraft flying conventional operations from TBO aircraft.

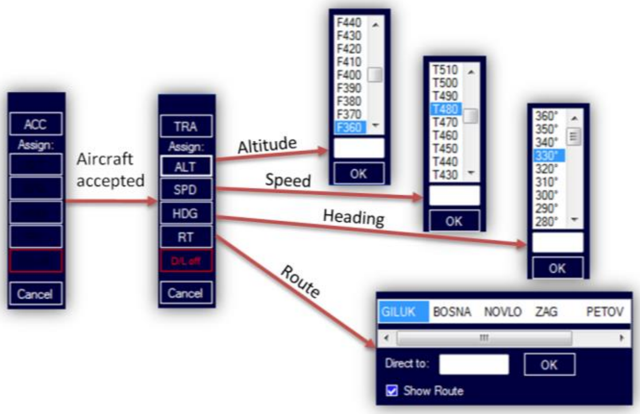


Fig. 4. Stripless Flight Progress Monitoring

Both air traffic controller and pseudo-pilot also have a separate list of flights which contains aircraft call-sign, type, departure aerodrome, route, destination aerodrome, and requested flight level (RFL). The user can sort this list by any column desired. Pseudo-pilot can open the main command panel by double-clicking the desired aircraft's call-sign.

V. SIMULATOR VALIDATION

The purpose of validation is to determine whether the simulator satisfies specified requirements. A full-scale, commercial ATC simulator solution is a complex system with multiple sub-systems using different models and technologies. On the other hand, this ATC simulator is a simplified, single-purpose system with significantly lower complexity and breadth of functions. The simulator was never to be used as a training device in its current form, nor has it had to be certified for safety-of-life functions. These conditions made the validation of the simulator much less demanding.

Though code testability requirement was met by implementing unit tests at class level, during the integration additional validation of the more complex modules had to be performed. Validation of the aircraft model was very important because accurate aircraft model enabled study participants (ATCOs) to make use of their experience and expert knowledge of aircraft performance to accurately assess the traffic situation (e.g. for conflict detection). It was also quite complex because the aircraft model is a hybrid system made of three distinct models (BADA APM, aircraft dynamics model, and FMS model). The approach taken in this research was to validate the aircraft model holistically by comparing the output of the aircraft model with the actual flight data obtained via Quick Access Recorder (Figure 5). The Quick Access Recorder (QAR) data was obtained for five flights by the Airbus A320 and five by the Bombardier Q400. Though it would have been more representative of the real aircraft distribution to include at least one heavy aircraft into this comparison, such data was unavailable. Nevertheless, the medium range jets and turboprops constitute largest relative fraction of the actual aircraft types in airspace of interest, so the authors believed that the compared aircraft were representative enough. The comparison of actual and modelled flights was however, made difficult by several factors.

First, variation of the weather conditions that occurred during the course of the actual flights introduced many errors. For instance, in one flight the wind varied from 5 knots at ground level to more than 80 knots at FL 240. The simple 3D grid wind model therefore, could not be used. Instead, the aircraft model was temporarily upgraded to include weather information from the look-up table produced from the QAR data. This ensured that the modelled aircraft 'flew' in almost exactly the same weather conditions as the actual aircraft. Each row in the look-up table corresponded to the weather conditions in a 100 meter thick layer of the atmosphere. The upgrade was later dismantled because it had no utility in further simulations.

The second problem was the speed schedule used by the airlines. BADA's default speed schedule was found to be biased towards higher speeds overall, so the speeds had to be decreased in order to match the speed schedule of the actual flight. For example, in BADA the Airbus A320 is scheduled to climb with 250 knots CAS at low altitudes and 300 knots at high, while the actual flight was flown with around 240 and 280 knots, respectively. Also, the BADA airline procedures

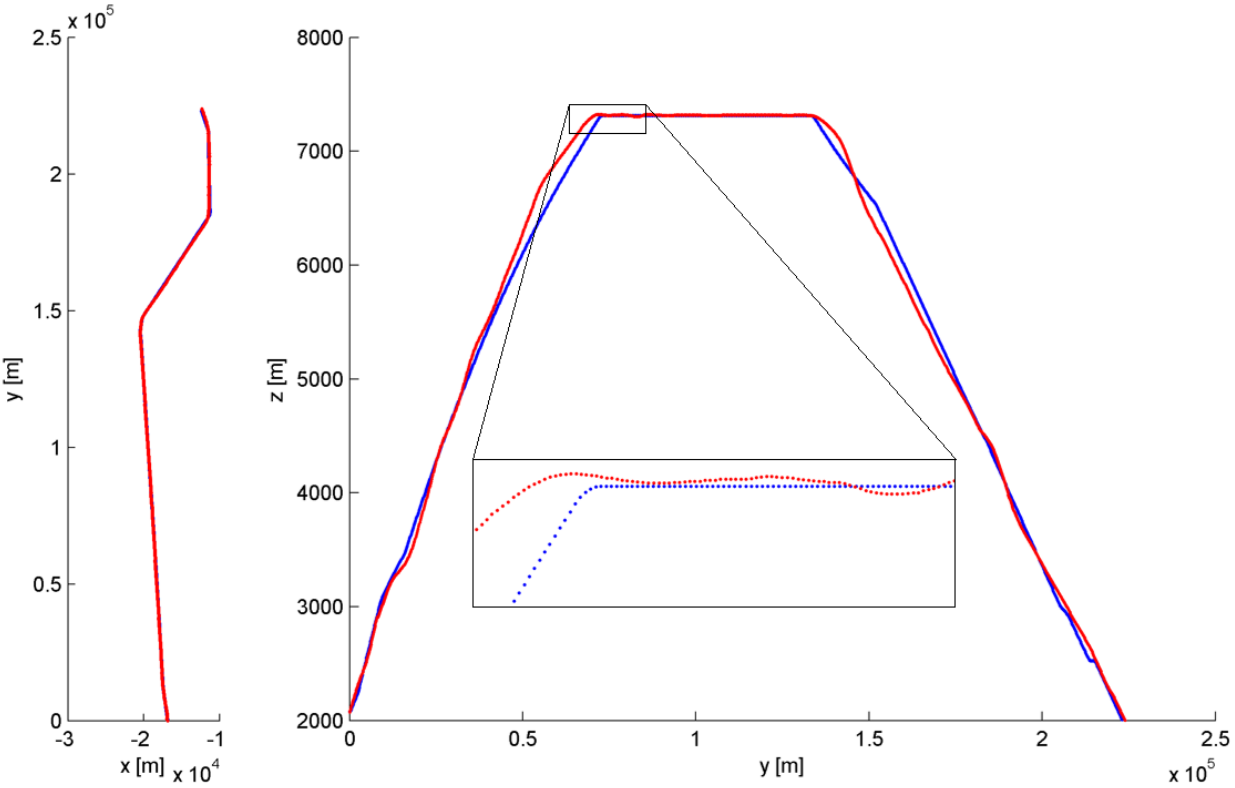


Fig. 5. Comparison of the Actual (Red) and the Simulated (Blue) Aircraft Trajectory (Top And Profile View)

model has only three different values for speed per flight phase (climb, cruise, descent) compared with many speed settings available to the actual FMS.

The third factor affecting the aircraft performance was the initial aircraft mass. Unfortunately (and surprisingly), the QAR data did not have the actual mass information so the masses of the modelled aircraft had to be tuned until the model performed as good as it could.

An example of the comparison between actual and modelled aircraft trajectory can be seen in Figure 5. Pictured is the trajectory of the Airbus A320 on a short local flight. Red lines represent the actual flight and blue lines the simulated flight. On the left of the figure is the top view and on the right is profile view.

One feature that is immediately noticeable is the relative smoothness of the simulated trajectory compared to the actual trajectory. Obviously, the FMS of the actual aircraft has to account for more disturbances than the simulated one (e.g. turbulence), however, the differences in trajectories at such a small level are not noticeable on the radar screen.

Finally, for the same example flight the 3-D error is shown in Figure 6. The error is calculated as a 3-D distance from the actual aircraft to the simulated aircraft for each second of the flight; therefore, apart from vertical and lateral, it also includes the along-track error. Maximum error is 4.7 km which is negligible for the purpose of this research.

All in all, the aircraft model can be considered valid and representative of the actual aircraft in the context of ATC operations. Several adjustments (wind, speed schedule, mass) are needed to bring the simulation results closer to the actual flight data since the default settings for an aircraft type are different than the settings used in practice.

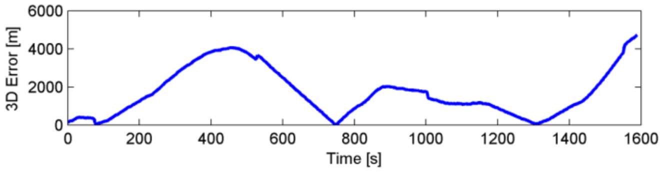


Fig. 6. 3-D Error of the Simulated Trajectory

Next in the validation process was the validation of the user interface and functionality testing. User interface was designed in accordance with the best practices observed from two professional ATC systems. However, as stated previously, not all of the tools have been, or needed to be, developed because not all of them were useful for this research. Validation of the user interface and functionality testing was performed during the trial runs with the assistance of two air traffic controllers who were not involved in this research in any other way. Feedback was received via unstructured interviews during which the controllers explained which user interface elements and simulator functions needed to be modified and why. These trial runs resulted in minor changes to functionality of the separation tool, color schemes, and interface layout. Additionally, some of the simulation scenarios were adjusted during these runs.

VI. SIMULATOR-SUPPORTED STUDY

The ATC research simulator was developed in order to examine the effect of TBO on air traffic complexity. In this section a brief overview of that study, with emphasis on simulator operations, will be presented. Full explanation of methodology and detailed analysis of the results will be presented in another paper.

This research was motivated by a combination of factors. The SESAR documents clearly emphasize the expected reduction in air traffic complexity after the introduction of TBO [7] but on the closer inspection the authors have concluded that there was virtually no scientific evidence of such an effect. Although the positive effect of TBO on complexity could be expected (based on the aggregated body of evidence explaining interactions among complexity, workload, and capacity [8], [9], [10], and [11]), only a dedicated study could prove or disprove its existence. Filling the gap between current evidence and expected results was the main motivation for the authors to begin the research.

Other reasons for this research stem from the previous research by the authors. Previous research, which was mostly focused on 4D navigation and conflict detection and resolution, was conducted using the fast-time simulations which proved (to the authors) the feasibility of 3D and 4D trajectory generation using hybrid aircraft models. A logical step forward was to test the concept using the real-time human-in-the-loop simulations.

Therefore, the main objective of this research was to measure the effect of TBO on air traffic complexity in en-route operations. This was to be achieved by performing an experiment on an ATC HITL simulator with air traffic controllers giving subjective complexity scores for conventional and trajectory-based operations.

Participants in the experiment were all trained and licensed ATCOs who had experience controlling the traffic in the Croatia Upper North airspace sector.

Nine different simulation runs were conducted involving three operations environments (conventional traffic, 30% aircraft flying TBO, or 70% aircraft flying TBO) and three air traffic levels (*low*, *high*, or *future*). Traffic data were sampled during off-peak periods to build scenarios with low traffic levels, and from peak periods to build scenarios with high traffic levels. In scenarios featuring a future traffic level, additional flights were added to routine traffic to give rise to an unrealistically high aircraft count; in addition, the proportion of aircraft climbing or descending was higher than in scenarios with low or high traffic levels. The aim of the future simulations was to expose controllers to complexity beyond what can be expected nowadays and beyond what the controllers had previously encountered in their careers.

Before the simulations, each controller received brief training in order to become accustomed with the simulator interface and operational procedures. The training consisted of an introductory lecture, pre-simulator briefing, trial simulator runs, and a post-simulator briefing. The introductory lecture covered basic topics in air traffic complexity, the subjective complexity rating scale used in our study, TBO, simulator tools and features, airspace, simulator scenarios, and operational procedures. The trial simulator runs lasted at least 90 min and involved two scenarios, one with conventional operations and one with TBO. All participants declined to participate in additional training simulations that were offered, indicating that they felt sufficiently comfortable with the simulator operations.

Controllers were asked to subjectively rate air traffic complexity throughout the simulation, using a modified Air Traffic Workload Input Technique (ATWIT) [12] scale that we term the Air Traffic Complexity Input Technique (ATCIT). The ATCIT scale features seven levels of complexity.

TABLE I. ATCIT SCALE

Complexity Level	Description
1	No complexity – no traffic
2	Very low complexity – very little traffic, no interactions
3	Low complexity – situation and interactions obvious at a glance
4	Somewhat low complexity – firm grasp of the situation, interactions are anticipated and prepared for
5	Somewhat high complexity – aware of the situation, interactions are handled in time
6	High complexity – having trouble staying aware of all interactions, occasionally surprised by unnoticed interactions and conflict alerts
7	Very high complexity – losing situational awareness, unable to track all interactions, responding reactively

The levels of subjective complexity on this scale reflect primarily the controller’s self-assessment of situational awareness, while also taking into account aircraft-aircraft and aircraft-airspace interactions. Before using this scale, controllers were briefed about the objectives of the ATCIT scale and the meaning of ‘complexity’, ‘interaction’, and ‘situational awareness’.

During each simulation run, a Subjective Complexity Measurement (SCM) tool opened every 2 minutes, accompanied by non-intrusive aural notification. The tool presented 7 buttons labeled 1-7, and the controller had to click on the button most closely matching the perceived level of air traffic complexity. Each assessment was time-stamped and stored. This is an example of simulator customization that might be very difficult or impossible to perform on an off-the-shelf simulator.

Our hypothesis in these experiments was that TBO would lead to lower air traffic complexity than conventional operations in en-route airspace sectors. The hypothesis was tested in three stages: first, means were compared between conventional and TBO scenarios in simulations with *low* traffic level; next, this process was repeated for simulations with *high* and *future* traffic levels. The hypothesis was tested using one-way repeated-measures ANOVA independently for each of the three traffic levels.

For scenarios with *low* traffic levels, after correcting for lack of sphericity, the results showed no significant effect of TBO on subjective complexity. For scenarios with *high* traffic levels, the results showed that TBO was associated with significantly lower subjective air traffic complexity scores. Post-hoc analysis showed that the mean difference was significant only between 0% TBO and 70% TBO, and between 30% TBO and 70% TBO. Since subjective complexity was assessed on an ordinal scale, we confirmed our results using the non-parametric Friedman test which yielded same results.

For scenarios with *future*-traffic analysis showed that TBO significantly reduced subjective air traffic complexity scores. Post-hoc analysis using the less stringent least significant difference to adjust for multiple comparisons showed significant differences between 0% TBO and 70% TBO and between 0% TBO and 30% TBO, but not between 30% TBO and 70% TBO. Results were confirmed with non-parametric Friedman testing.

These results suggest that TBO can significantly reduce subjective air traffic complexity, but only when the traffic level and proportion of TBO aircraft are high.

VII. CONCLUSION

ATC simulators are commonly used tools for ATM research, however, they are usually not available to smaller research teams due to cost. This paper showed the methods and technology needed to build an ATC simulator for real-time human-in-the-loop use. While developing such a simulator is not always cost-effective, adhering to the general simulator requirements mentioned in this paper ensures that the simulator is easy to upgrade and reuse thus increasing its utility.

Specific simulator requirements should be defined based on the simulator purpose. Here, an example of specific requirements for ATC simulator used in HITL en-route simulations was presented. These requirements are specific to this project; however, authors believe that there are many other research problems that could be tackled with it (e.g. capacity, complexity, or workload assessment, ATC tool validation, procedure design and validation etc.).

The example of simulator framework presented in this paper shows one possible approach to achieving the maintainability requirement through modularity. It also shows which modules are required for implementation of which functionality in this type of ATC simulator. A brief overview of key technologies was presented to help guide other researchers wishing to develop a simulator of their own. The simulator was validated in two ways: by comparing the generated trajectories with actual aircraft trajectories and by comparing the user interface and ATC tools functionality with commercial ATC simulation devices. Both comparisons showed that the simulator performed adequately.

Finally, the study for which this simulator has been developed was presented. During this study the simulator performed well. Licensed ATCOs had no trouble adapting to it during the first 90 minutes of training. Besides testing the simulator in actual working environment, study also provided meaningful results in terms of air traffic complexity assessment. It showed that the air traffic complexity in en-route operations will decrease once the trajectory-based operations were implemented. This decrease in complexity will only be noticeable in traffic situations with larger fraction of aircraft flying TBO and in situations with larger traffic volume.

ACKNOWLEDGMENT

The authors wish to thank the air traffic controllers who participated in this study.

REFERENCES

[1] S. Fothergill, S. Loft, and A. Neal, “ATC-lab Advanced: An Air Traffic Control Simulator with Realism and Control,” Behavior Research Methods, vol. 41, Issue 1, pp. 118–127, February 2009.

[2] B. Berenbach, D. Paulish, J. Katzmeier, and A. Rudorfer, “Software & Systems Requirements Engineering: In Practice”. New York: McGraw-Hill Professional. ISBN 0-07-160547-9., 2009.

[3] K. Amonlirdviman, T. C. Farley, R. J. Hansman, J. F. Ladik, D. Z. Sherer, “A Distributed Simulation Facility to Support Human Factors Research in Advanced Air Transportation Technology”, Fall Simulation Interoperability Workshop; 13-18 Sep. 1998; Orlando, FL; United States

[4] A. Nuić, C. Poinsot, M.-G. Iagaru, E. Gallo, F. A. Navarro, and C. Querejeta, “Advanced Aircraft Performance Modeling for ATM: Enhancements to the BADA Model,”, 24th Digital Avionics *System Conference*, Washington D.C., USA, 2005, p. 2B4.

[5] C. Sheehan, "Coverage of 2012 European Air Traffic for the Base of Aircraft Data (BADA) - Revision 3.11," EUROCONTROL Experimental Centre, Brétigny-sur-Orge, France, Technical/Scientific Report 13/06/27-06, 2013.

[6] W. Glover and J. Lygeros, "A Multi-Aircraft Model for Conflict Detection and Resolution Algorithm Evaluation," HYBRIDGE Project, WP 1, Deliverable D1.3 IST-2001-32460, 2003.

[7] SESAR Consortium, "D1 - Air Transport Framework - The Current Situation," 2006.

[8] R.H. Mogford, J.A. Guttman, S.L. Morrow, and P. Kopardekar, "The Complexity Construct in Air Traffic Control: A review and synthesis of the literature," Atlantic City, 1995.

[9] S. Athènes, P. Averty, S. Puechmorel, D. Delahaye, and C. Collet, "Complexity and Controller Workload: Trying to Bridge the Gap," in Proceedings of the 2002 International Conference on Human-Computer Interaction in Aeronautics (HCI-Aero 2002), Cambridge, MA, USA, 2002.

[10] P. Kopardekar, A. Schwartz, S. Magyarits, and J. Rhodes, "Airspace Complexity Measurement: An Air Traffic Control Simulation Analysis," in US/Europe 7th Air Traffic Management Seminar, Barcelona, 2007.

[11] T. Prevot and P.U. Lee, "Trajectory-Based Complexity (TBX): A Modified Aircraft Count To Predict Sector Complexity During Trajectory-Based Operations," in *IEEE/AIAA 30th Digital Avionics Systems Conference (DASC)*, Seattle, USA, 2011, pp. 3A3-1.

[12] E.S. Stein, "Air traffic controller workload: An examination of workload probe," FAA, Atlantic City, New Jersey, USA, Report DOT/FAA/CT-TN84/24, 1985.