UDC 656.7.022

**A.V. IZVALOV, V.N. NEDELKO, S.N. NEDELKO**

***Kirovograd Flight Academy of the National Aviation University, Ukraine***

# VALIDATION OF INPUT DATA FLOW GENERATING ALGORITHM FOR INTELLIGENT AIR TRAFFIC CONTROL SIMULATOR

*Implementation of the automated input data flow generation algorithms for the air traffic control simulator is likely to help to use the advantages of individual-based training better. The article describes the experiment based on individualized learning that was held to validate the algorithm. In the research with the use of developed software the exercises were generated and their parameters were compared to the requirements. As the deviations appeared to be within the acceptable range, it is possible to use the developed software for the purpose of professional training.*

***Key words**: Air traffic control simulator, input data flow, automatic exercise generation, validating experiment, parameters deviations, supervisor interface, individual-based training.*

## Introduction

As the increasing trend of the air traffic intensity exists nowadays, the requirements to the professional abilities and skills of the air traffic control (ATC) specialists are growing. As these skills and abilities result from the professional training, the requirements to its quality grow as well.

Individualized learning is considered to be one of the most effective ways to increase the quality of professional training in ATC field [1]. To reach the required level of professional skills and abilities each trainee should complete his/her own sequence of tasks and exercises. Such sequence can be subject to changes basing on intermediate trainee's results.

Currently such an approach exists in several ATC training facilities, but the individualized learning potential of the intelligent ATC simulator can be dramatically improved if there exists a possibility to produce high quantities of appropriate sequences of tasks within a reasonable amount of time.

At Kirovograd flight academy of the National Aviation University (KFA NAU) the algorithms of automated task generation were developed and implemented in the form of software application for the ATC simulator [2]. To ensure that the automatically generated tasks will have the appropriate learning value, the validation of the algorithm was performed.

## 1. Input data flow of ATC simulator

The intelligent ATC simulator processes several data flows (fig 1).

Input data, which refers to the air traffic parameters, expected conflicts, simulated events etc;

Simulation data, which refers to the flight parameters of the virtual aircraft, it is changing in the real time;

Trainee's commands, which affect the simulation data;

Exercise history, which is the base for the student performance assessment.

The exercise generation is in fact the generation of the input data flow. But the requirements to this process are the parameters of the simulation data. So, the task is to design the algorithm of generation of such input data flow which would create the simulation data with the required parameters, when the exercise is run on the ATC simulator.

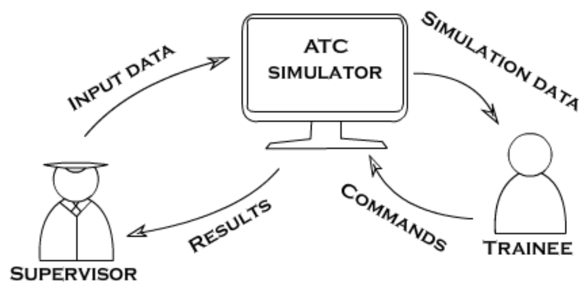


Fig. 1. Data flows in ATC training

Such algorithm should fulfill the following requirements:

The deviation of the simulation traffic parameters from the requirements should be within the appropriate range (which is the subject to guiding documents in ATC training).

The standard training exercises should be generated within a reasonable amount of time, no more than 10 minutes (while the intelligent ATC simulator is prepared for the next session). It must be possible to create various input data which will lead to the same simulation data traffic parameters (as the same trainee may require several tasks with the same learning value but different traffic patterns).

The main element of the input data flow is the set of the flight plans (FPLs) $F = \{f_i\}$. The flight plan has such properties as the function (L(t)) describing the expected change of the aircraft's coordinates with time and aircraft attributes (A), like it's type, call sign, radio language etc.

Over the set $F = \{f_i\}$ the graph K of expected conflicts is defined. It's incidence function returns the value of True if and only if fulfils

$$\exists\tau\left(\tau \in [T_0;T_1] \wedge \mathrm{Conf}\left(L_i(\tau), L_j(\tau)\right)\right),$$

where $[T_0;T_1]$ is the time interval of the simulation, $\mathrm{Conf}\left(L_i(\tau), L_j(\tau)\right)$ – the function which defines if there was the conflict between the aircraft moving along the trajectories of $L_i(t)$ and $L_j(t)$ at the moment of $\tau$.

The set of events $E = \{e_i\} = \{\langle et_i, te_i, oe_i\rangle\}$ is the next element of the input dataflow, where $et_i$ is the i-th event type (for example, engine failure or icing), $et_i$ is the moment of the i-th event and $oe_i$ is the object in the airspace to which the event is applied.

The complexity of the simulation can be defined as the function of time

$$C(t) = \sum_i c_i(t),$$

where $c_i(t)$ is the complexity function of the i-th aircraft from the F set. It depends on the flight mode, conflicts involving this aircraft or emergencies on it. The simulation data parameters which define the requirements to the input data are: graph K, set E and function C(t). As usual, in the official papers regulating the ATC training the requirements to the training exercises are defined using the linguistic or ranged variables instead of the precise ones (e.g. "gradual complexity growth during the first half of the exercise", "1-2 crossing conflicts", "total traffic of 40-45 aircraft", etc) the fuzzy logic can be used to interpret this data. At the same time, the acceptable deviations of the simulation data parameters from the requirements can be defined.

## 2. Algorithm validation

The algorithm of automated task generation was described in [3]. The software developed has three input modes: manual (fig. 2), visual and that from the data file.

For the validating experiment, N=30 sets of requirements were composed basing on the guiding documents. For each set of the requirements $R_i$ M=100 times the automated exercise generator was run. The intelligent ATC simulator in the speed-up mode ran each of the exercises $X_{ij}$ and calculated the simulation data parameters $P_{ij}$. The differences $D_{ij}$ between these parameters $P_{ij}$ and the requirements $R_i$ were calculated. The conditions for the differences to be within the required range were checked.

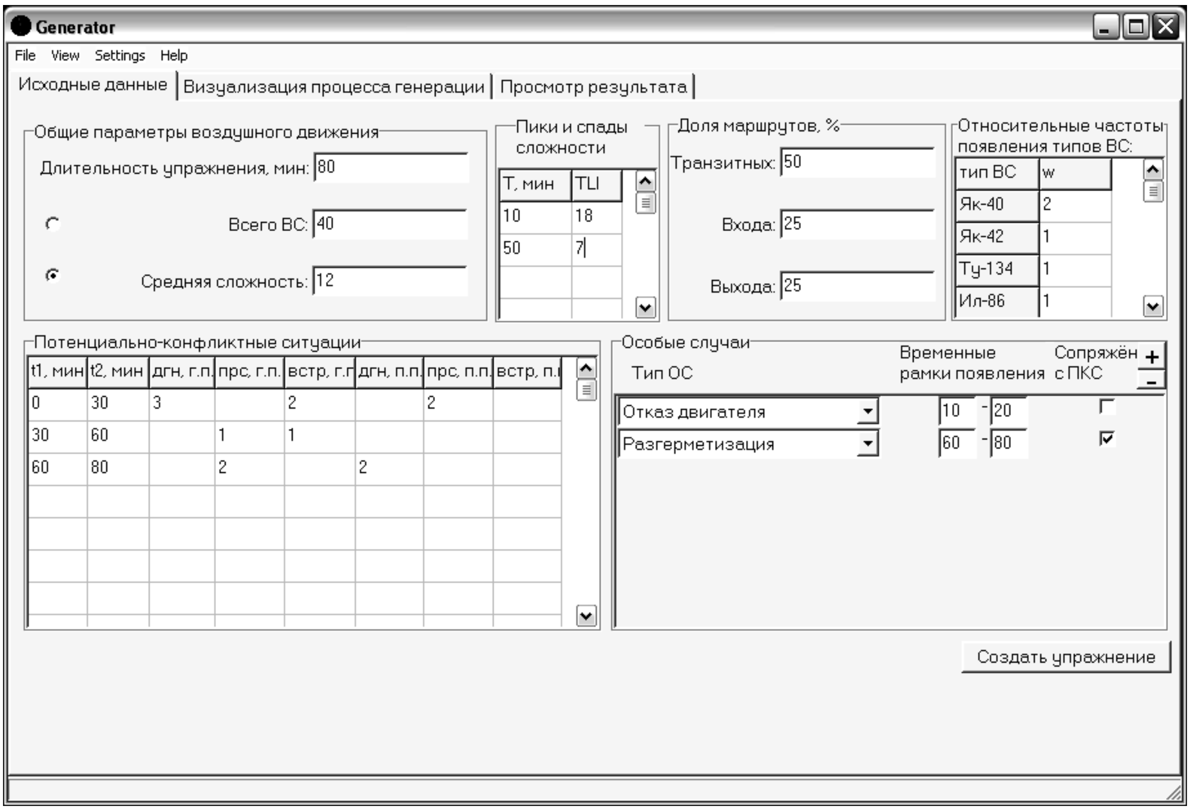


Fig. 2. Manual input screen for the requirements to the ATC exercise

The parameters which were included into the set of the differences and which defined the exercise fitness for training were the following:

$\sigma_{AC}$ – standard deviation of the number of aircraft controlled ($n_{AC}(t)$) in the exercise $X_{ij}$ from the respected function in the requirements $R_i$. $M(n_{AC})$ – maximum absolute deviation of $n_{AC}(t)$ in $X_{ij}$ from that of $R_i$;

$\sigma_C$ and M(C) – the respected parameters of the complexity function C(t) of the created simulation data;

$\Delta K$ - the deviation of the set of conflicts in the simulation data from the graph K of the requirements. It is a matrix, the elements of which $\Delta K_i^t$ are the differences of the number of conflicts of i-th type during the t-th phase of the exercise in the actual simulation data from the requirements;

$\Delta E_i$ - the absolute difference of moments of the events of i-th type (if the event of some time hadn't been generated this difference is considered to be an infinity).

The ranges for these parameters were established:

$$\sigma_{AC} \le 1.5,\ M(n_{AC}) \le 2,\ \sigma_C \le 3.5,\ M(C) \le 5,\ \left|\Delta K_i^t\right| \le 1,$$

$$\sum_t \Delta K_i^t = 0,\ \Delta E_i \le 5 \text{ min}.$$

The structure of the input data flow generating algorithm ensures the exact equality of the sets of conflicts and events in the created exercise and in the requirements. That's why the limiting values of $\Delta K_i^t$ and $\Delta E_i$ are so strict. The structure of the validating experiment is shown in fig.3.

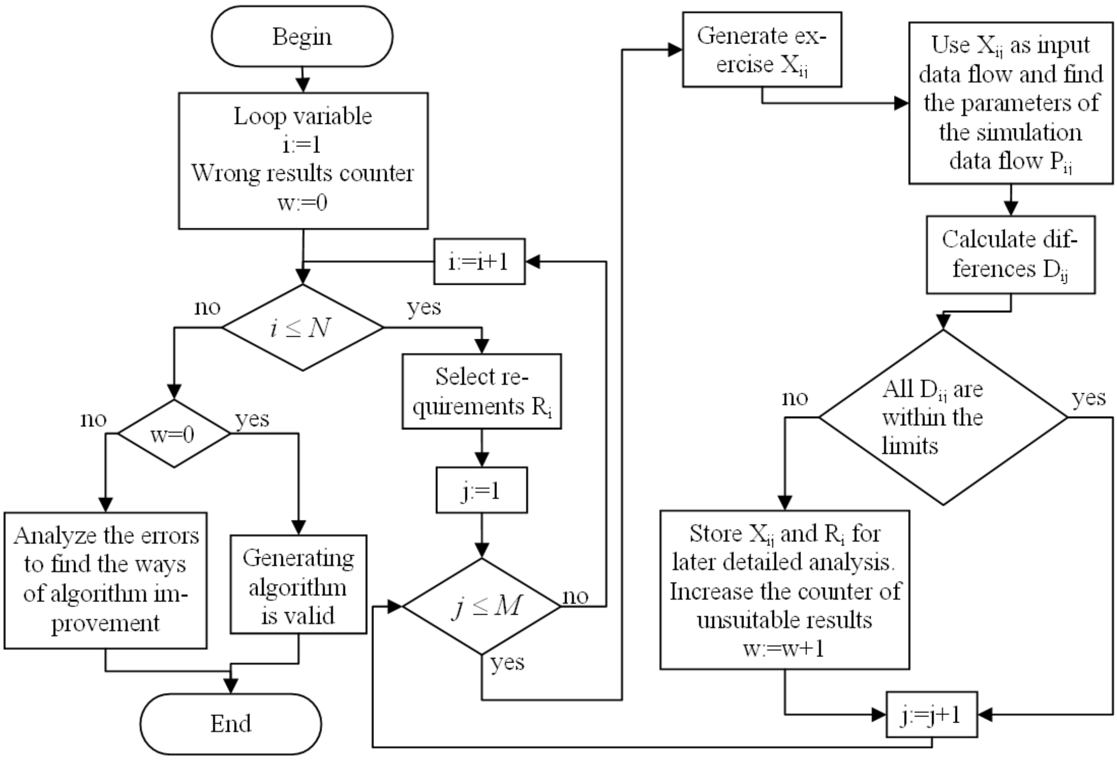


Fig. 3. Algorithm of the validating experiment

The experiment proved that the algorithm is valid and the software is able to generate the exercises with the exact numbers of conflicts and events of given type.

The parameters $\sigma_{AC}$, $M(n_{AC})$, $\sigma_C$ and M(C) remained within the required ranges. Their mean values throughout all the 3000 of the created exercises were:

$$\overline{\sigma_{AC}} = 0.76,\ \overline{M(n_{AC})} = 1.57,$$

$$\overline{\sigma_{AC}} = 2.32,\ \overline{M(C)} = 3.74.$$

The example of the required C(t) and the complexity of the simulation data flow is shown in fig. 4.

## 3. Visual exercise construction interface

The software developed was implemented into the ATC training process in KFA NAU. For the purpose of quick editing already created exercises and designing the exercises in the visual mode, the interface shown on fig.5 was developed. The judgement of the ATC training centre supervisors was taken into account. The interface elements are: (1) – main menu which enables to carry out save/load and automatic exercise generation functions, (2) – ATC sector map where the currently added FPLs and expected conflicts between them can be viewed, as well as the planned events. This element is also used to add/remove/edit the parameters of FPL using drag-and-drop technique. (3) – time trackbar, it makes possible to see the expected state of the simulation data at any certain moment. (4) – list of all FPLs, (5) – list of points of the selected FPL's track. They are used to edit the FPL's properties using the keyboard input. (6) – planned and actual complexity graphs. The latter element lets the designer control the workload caused by the currently developed exercise. (7) – visual representation of dependency between the flight parameters of the selected/added FPL and the presence of the certain-type conflict. As the designer moves the mouse over the red cell, the place of the future conflict is highlighted on the map (2).

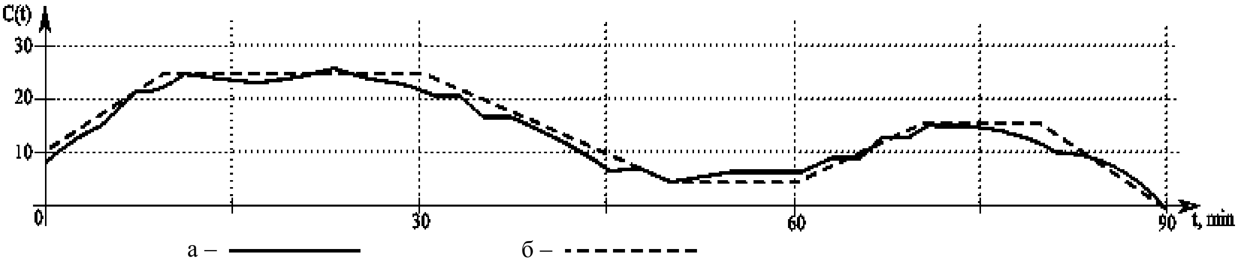


Fig. 4. Examples of target (a) and actual (b) complexity functions of the simulation data flow

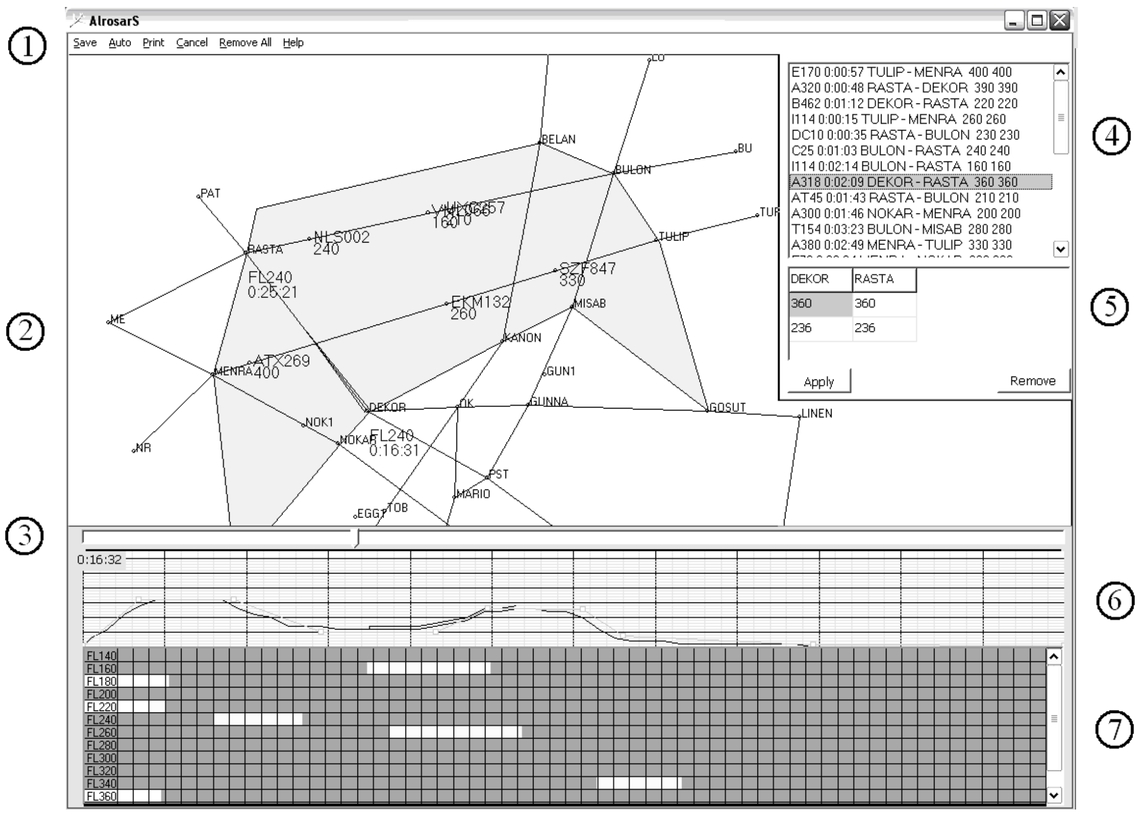


Fig. 5. Interface for visual exercise design and editing

## Conclusion

The algorithm developed to generate automatically the input data flow for the ATC simulator was implemented as the software application. The validating experiment showed that the deviations of the simulation data flow parameters from the requirements were within the ranges defined by the guiding documents in the field of ATC training.

The parameters taken into account were the standard and maximal absolute deviations of the number of aircraft and traffic complexity. Also the conflicts and events like emergencies were taken into consideration. Thus, all the groups of professional abilities and skills were covered.

Besides the automatic exercise generation the visual exercise design is enabled. The exercises developed both in the automatic and the manual modes are extensively used in training, self-training and professional competitions of ATC controllers held in the academy.

## References

*1. Dong, Mei Zhang. Exercise design approach for intelligent simulation based training system [Text] / Dong Mei Zhang, Alem, L. // Intelligent Information Systems, 1995. ANZIIS-95. Proceedings of the Third Australian and New Zealand Conference on. – 1995.– P. 13 - 18.*

*2. Ізвалов, О.В. Алгоритм обробки даних повітряної зони, що забезпечує можливість автоматичної генерації навчальних вправ [Текст] / О.В. Ізвалов, В.М. Неділько, С.М. Неділько // Радіоелектронні і комп'ютерні системи. – 2009. – № 5 (46). – С. 301 – 304.*

*3. Извалов, А.В. Алгоритм автоматизированной генерации упражнений для тренажёра управления воздушным движением [Текст] / А.В. Извалов, В.Н. Неділько, С.Н. Неделько // Сучасні інформаційні та інноваційні технології на транспорті: матеріали Міжнародної науково-практичної конференції. Т. 3. – Херсон: Видавництво Херсонського державного морського інституту, 2009. – С. 27 – 29.*


*Поступила в редакцию 1.03.2013, рассмотрена на редколлегии 27.03.2013*

**Рецензент:** д-р техн. наук, проф. зав. каф. В.А. Краснобаев, Полтавский национальный технический университет им. Ю. Кондратюка, Полтава, Украина.

## ВАЛІДАЦІЯ АЛГОРИТМУ ГЕНЕРАЦІЇ ВХІДНИХ ДАНИХ ДЛЯ ІНТЕЛЕКТУАЛЬНОГО ТРЕНАЖЕРУ АВІАДИСПЕТЧЕРА

***О.В. Ізвалов, В.М. Неділько, С.М. Неділько***

Впровадження алгоритмів автоматичної генерації вхідного інформаційного потоку для тренажера авіадиспетчера дозволить краще використовувати переваги індивідуального підходу до навчання. Стаття описує експеримент, який базується на індивідуальному навчанні, які було проведено з метою валідації алгоритмів. У дослідженні за допомогою розробленого програмного засобу в автоматичному режимі виконувалася генерація вправ і порівняння їх параметрів з цільовими. Відхилення опинилися в допустимих межах, що дозволяє використовувати розроблений програмний засіб у навчальному процесі.

**Ключові слова**: тренажер авіадиспетчера, вхідний інформаційний потік, автоматична генерація вправ, валідаційний експеримент, відхилення параметрів, інтерфейс інструктора, індивідуальне навчання

## ВАЛИДАЦИЯ АЛГОРИТМА ГЕНЕРАЦИИ ВХОДНЫХ ДАННЫХ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОГО ТРЕНАЖЁРА АВИАДИСПЕТЧЕРА

***А.В. Извалов, В.Н. Неделько, С.Н. Неделько***

Внедрение алгоритмов автоматической генерации входного информационного потока для тренажёра авиадиспетчера позволит лучше использовать преимущества индивидуального подхода к обучению. Статья описывает эксперимент, базирующийся на индивидуальном обучении, которые был проведен с целью валидации алгоритмов. В исследовании при помощи разработанного программного средства в автоматическом режиме выполнялась генерация упражнений и сравнение их параметров с целевыми. Отклонения оказались в допустимых границах, что позволяет использовать разработанное программное средство в учебном процессе.

**Ключевые слова:** тренажёр авиадиспетчера, входной информационный поток, автоматическая генерация упражнений, валидационный эксперимент, отклонения параметров, интерфейс інструктора, индивидуальное обучение.

**Извалов Алексей Владимирович** – старший преподаватель кафедры информационных технологий, Кировоградская лётная академия Национального авиационного университета, Кировоград, Украина, e-mail: aviglau@gmail.com.

**Неделько Виталий Николаевич** – канд. техн. наук, доцент, Кировоградская лётная академия Национального авиационного университета, Кировоград, Украина, e-mail: nvn@glau.kr.ua.

**Неделько Сергей Николаевич** – д-р. техн. наук, профессор, Кировоградская лётная академия Национального авиационного университета, Кировоград, Украина, e-mail: asup@glau.kr.ua.