ПРИКЛАДНІ ДОМЕНИ І ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

UDC 004.413:338.5

V.A.Khomenko,
A.Bikadorov,
National Aviation University

APPROACH AND SOFTWARE FOR THE NOISE SIMULATORS DEVELOPING

Approach is based on the controllable sound samples and enables to build the noise simulators for flight simulators using the widely accessible aircrafts cockpits sound records. This approach and software tools can be applied at the low-budget projects under condition of inaccessible aircraft. The case study of the two-engine turboprop airplane noise simulator development is presented.

Підхід оснований на використанні керованих зразках звуку та дає можливість побудови імітаторів шуму для авіаційних тренажерів з використанням загальнодоступних записів звуку у кабінах літальних апаратів. Запропонований підхід та програмні засоби можуть застосовуватися у малобюджетних проектах в умовах недоступності літальних апаратів. Наведений практичний приклад розробки імітатора шуму для дводвигунового турбогвинтового літаку.

Подход основан на использовании управляемых образцах звука и дает возможность построения имитаторов шума для авиационных тренажеров с использованием широкодоступных записей звука в кабинах летательных аппаратов. Предлагаемый подход и программные средства могут применяться в малобюджетных проектах в условиях недоступности летательных аппаратов. Приведен практический пример разработки имитатора шума для двухдвигательного турбовинтового самолета.

Keywords: sotware, noise simulator, flight simulator TL 410.

Introduction

Noise simulator is the integral part of the any modern flight simulator and provides the imitation of the real sound environment [1] during the training. Noise is the important informational source for pilots about states of the board mechanisms and equipment, runway touchdown etc. Modern noise simulators for commercial flight simulators is the hardware-software systems that form and generate the aircraft noises for all flight modes with defined adequacy and quality [2, 3].

The quality noise simulator creation is the complex and expensive process that requires studying and analysis of the object aircraft noises in the different work modes, synthesis of the single noises generators, matching the sound dependencies on the flight parameters and verification of adequacy.

Problem statement

Under the inaccessible object aircraft condition the sound characteristics analysis and adequacy checking are impossible. This condition can be appeared at the small budget projects, implemented independently from manufacturer and operators. But simulator developer now is able to get easy the flight audio (and video) records from the aircraft cockpit, made by any appropriate compact record device. This article proposed low-cost approach to noise simulators creation using accessible aircraft sounds records and describes the case study of one implementation for the two-engine turboprop aircraft simulator.

Traditional approach.

Models. The sound environment study includes the perception and recording of the aircraft sound at the different work modes and analysis for the single noise sources separation and definition of their characteristics. Under access to the aircraft the developer is able to study noises sources separately, using control of the aircraft mechanisms and recording the audio characteristics by the measurement tools, associated to the aircraft work mode and conditions of observations.

The model M of the sound environment is usually presented as the additive set of noises; each of them is created by single source (aircraft mechanism or the environment factor):

$$M = \sum_{i=1}^{n} m_i ,$$

where m_i – model of the single noise.

For each model m_i should be described the noise characteristics and their dependencies on set of the noise source parameters. Noise control can be described by the functional that defines sound parameters dependencies $Y = \{y_1, y_2, ... y_k\}$ on the input parameters of source $X = \{x_1, x_2, ... x_l\}$:

$$m_i(Y) = F(X).$$

It is useful to cast the functional F to the set of the independent functions; each of them describes the one output parameter changing, depending on input set:

$$y_i = f_i(x_1, x_2, ... x_m\}.$$

For example, lets considerate the simple model of the turbojet engine noise that includes only two noise sources – starter m_s and turbine m_t:

$$M = \{m_s, m_t\}.$$

The general starter noise model has the formulae

$$m_s(V) = f_{sv}(S_s),$$

where V – is the noise volume;

f_{sv} – is noise volume function of the starter state dependency;

S_s – is starter state (on/off).

Then model can be reduced to the noise volume function of the starter state dependency:

$$V = f_{sv}(S_s)$$

Here, let's reduce function f_{sv} to binary function:

$$V = \frac{0\_if\ \_S = OFF}{1\_if\ \_S = ON}$$

The model of turbine can be formulated as

$$m_t(V, F) = f_t(R_t),$$

where V – is the noise volume;

F – is the noise frequency bandwidth;

f_t – is volume and frequency functional of the turnover dependencies;

R_t – is the turbine turnover frequency.

Here, we can present the volume and frequency bandwidth as independent functions of the turnover $V = f_{tv}(R_t)$ and $F = f_{tF}(R_t)$, which can be reduced to the linear dependencies.

Implementation. The traditional simulator structure (fig.1) includes set of the controlled noise generators which output signals are summarized [1, 2]. Simulators are distinguished by the playback channels number (mono, stereo, quadro and more) and functions balances between hardware and software. For each single noise source using characteristics of its model m_i the noise generator and control dependencies are selected. The generators selection and tuning are the expensive operations.

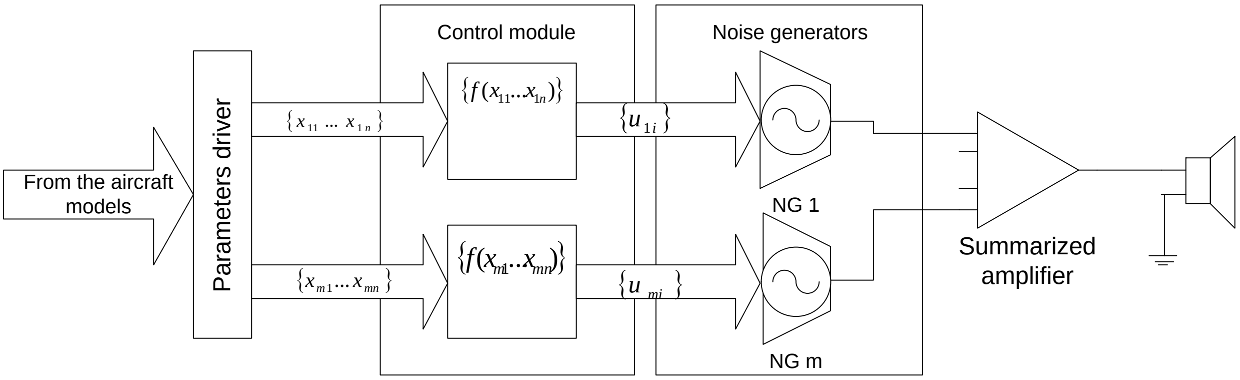


Figure1. Traditional structure of the noise simulator

Simulator's adequcy is verified by the quantity and quality evaluations [4]. The final adequcy have to be verified by the experienced pilot. On the beginning stages of the developing the developers apply own subjective evaluation, using generated noise comparing with the sample records.

The objective approach to the adequacy verification is realized using the generated noise parameters mesurement and comparable spectrum analysis of the generated and sample noises.

Approach, based on the controllable samples

The idea of the aircraft sound records usage for the sound environment imitation is not new, but is not applied for the professional simulator owing to different reasons [1], particularly because perfect digital audio technologies became accessible for simulator developers later than advanced analog noise generation. The controllable sound file playbck is used at the some game flight simulators, e.g. MS Flight Simulator [5], but roughly and somtimes with the low adequcy. Here, is proposed the simulator constructor, based on the playback of the aircraft sound digital record samples. These samples are played once or iteratively with the control parameters, depend on the work mode of the noise sources. The simulator developing includes the following steps: samples aquiring; sound environment model developing; implementation and tuning;.

Samples acquiring. In general, simulator developer, having one or several records of the aircraft sound, made by the thirty person, can only indirectly guess about their conditions, using the general sound panorama or associated video. The first problem of the samples acquiring is the sources separation and identification. The second is the association of the noise characteristics to the sources work parameters. The third is the absolute noise level definition, because the usual recorders do not fix it. Automatic record control systems worsed this problem, made even relative sound levels unauthentic.

The developer has to identify the single noises and define their parameters. The noises can be identified subjectively (by ear) and approved using spectral analysis and comparing the spectral components with the aircraft modes. Then, for ifentified sources the appropriate samples must be extracted by selection and cutting. This task usually is complicated because of other noise sources, outside noises (pilots talking, switches clicking etc). In addition, sample has to be long enough (5 seconds and more) to be played iterativly without spectral structure loosing. These conditions may require the careful exploration of records, using flight documentation.

Sound environment model development. The sound environment model for the controllable samples approach is different from the traditional. The models of the single sources can not be considered as independent, because the records not contain the “pure” noises, but mix of them. Hereby, single model is depended the several other noises:

$$m_i(Y) = F(X, \{m_k\}, k \neq i),$$

where m_i – selected model,

Y – noise parameters set,

X – noise source parameters set,

$\{m_k\}$ – set of the other noises models.

This problem should be solved by the extraction of maximum disconnected samples and correction of dependencies when the parameters control rules are defined. The technique of exctraction may be following: listener finds the enough long records fragment, where single source noise is most “clear’, then correlates this noise with the source modes. It can be executed using indicator information from the videorecord or using the spectral analysis and knowledge about aircraft mechanisms modes. For example, the air screw noise has the spectral “peaks” of the main screw rotary frequency and its harmonics, multiplied by the screw blades:

$$\omega_i^g = \omega^s \times n \times i,$$

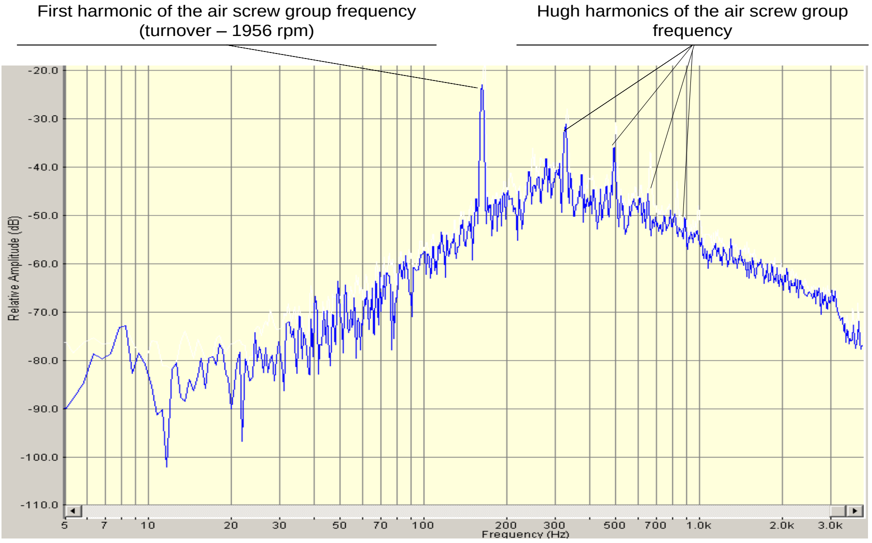
where ω_i^g – frequency of the i harmonic,

ω^s – screw frequency,

n – screw blades number,

i – number of harmonic.

Calculating the harmonics frequencies to the screw frequency and comparing it with the nominal engine modes (fig.2) enable to define the appropriate record fragment.



Take of mode

Figure 2. Spectral analysis of the turboprop airplane

The dependencies between single noises models can be taken into account based on knowledge of the mutual dependencies of the appropriate mechanisms of aircraft. For instance, air screw noise can not be separated from the turbine noise during takeoff and cruise modes. And linear changing of the volume and frequency of this sample is not relevant to the turbine noise control and leads to dip one at the low turnover mode. The turbine noise sample has to be extracted from the other record fragment, where the air screw noise is unsignificant, for example, during the some interval after engine start. Then, the control dependencies of the both samples must take into account the mutual component (fig.3).

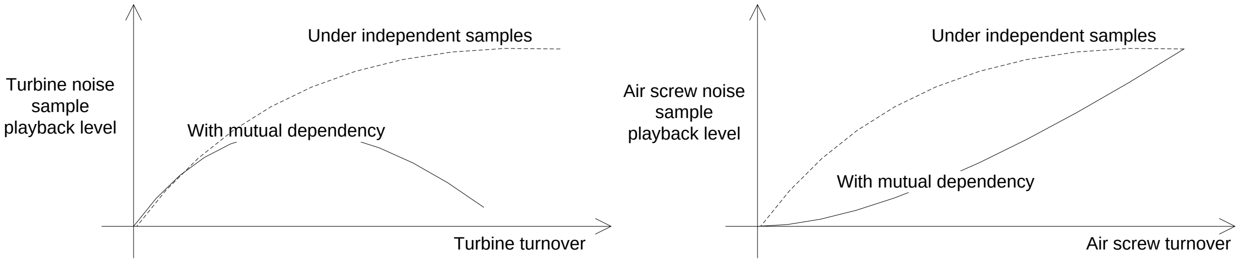


Figure 3. Correction of the volume control for the mutual dependent noise samples

The problem of the absolute noise levels at the cockpit can be solved here, using only expert evaluation.

Implementation and tuning. The proposed noise simulator structure is similar to traditional and includes the set of the controllable palyers with summarized outputs (fig.4). The playback can be controlled by speed (for the frequency change effect), by volume and by several additional parameters, as the flange-effect, echo, etc. The playback mode can be one-time (for the one-time sounds, similar gear extension/retraction) and looped (for continuous, similar turbine or air screw noise). The dependencies between the controlled players parameters and modeled aircraft parameters are executed by the special converters. Each converter can realize the function of the one parameter of player from the one or several parameters of the model. Usually, the playback speed is the function of the one parameter. The volume can be product of the single functions:

$$V = V_{\max} \times \prod_{i=1}^{n} f_i(x_i),$$

where V – playback volume;

$V_{\max}$ – maximum noise volume;

$f_i(x_i)$ – normalized function of parameter V from model parameter x_i.

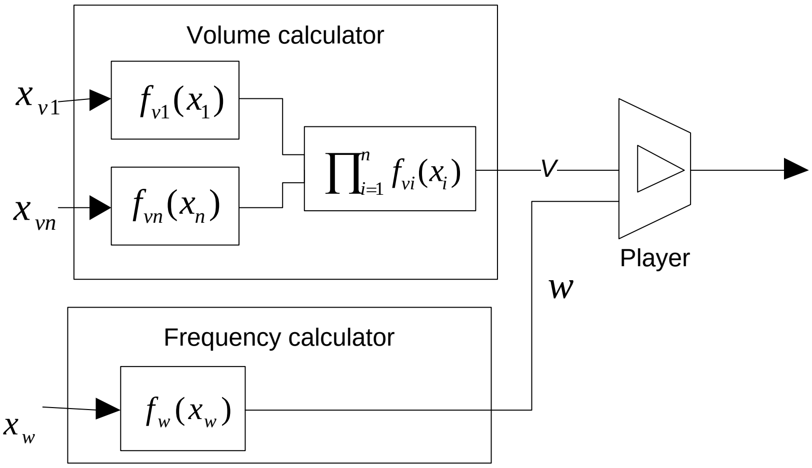


Figure 4. Structure of the controlled player – the noise simulator element

This elaboration is possible due to multiplicative character of the mechanisms volume dependencies. For example, the air screw volume depends multiplicatively on the rotor frequency and pitch (blade angle); wheel touch volume – on horizontal and vertical landing speeds. In these cases the control function can be realized as the product of the normalized approximation functions, defined for the each parameter. This approach enables to create the suitable user tool for the control function specification as the set of the aircraft parameters $\{x_i\}$ and functions $f_i(x_i)$, (fig.4) defined by the piecewise-linear approximation.

Case study

The proposed approach was developed and used at the reengineering project of the full flight simulator TL410M of National Aviataion University [6-8]. The universal short-range airplane L410 «Turbolet» is the classic two-engine turboprop highpalne, equipped with engines Walter M 601A and controllable pitch air screws [9]. Since 1961 more than 1100 airplanes where produced and most of them are exploited at the former USSR countries. Considerable part of the L410 is flying at the Easten Europe, Latin America and Africa.

At the beginning project moment (2006) the original Czech flight simulator was defective long time. Obsolete computer system of the simulator was unrepairable, but main mechanical and electrical systems where operable. The computer system was replaced to new and legacy software was reworked by the teachers and students of university. The original pure hardware noise simulator was out of order. Under the low-budget and aircraft inaccessible conditions the developers (teachers, engineers and students of the computer science department) created the new one, used the audio and video records from the L410 airplane cockpit and special developed software constructor.

Sample aquiring. The original noise simulator provided considerable noise set: engine exhaust fumes, airscrew, wind, starters, taxing [10]. The first task was the acquiring the airplane sound records for the new simulator. The base record was gained by recorder at the cockpit during the flight L410 airplane. This record gave the general presentation about sound at the different flight stages and was used for the part of samples. The fragment of the record oscillogram is shown on the fig.5. The main problem of this record was the absence of noise source identification. The identification was executed using record hearing and comparing with the airplane exploitation documentation. Thereby the main flight stages (рис.5) and noise sources where identified (starters, engines, screw etc). The correlation of these noises to the mechanisms modes where provided using spectral analysis, based on the air screw noise component (fig.2). Screw turnover frequencies, calculated for different flight stages, match accuracy to the pilots' flight guide. Because turbine turnover is hard correlated with air screw rotary its modes where defined too. Then, the most independent noise samples where extracted.

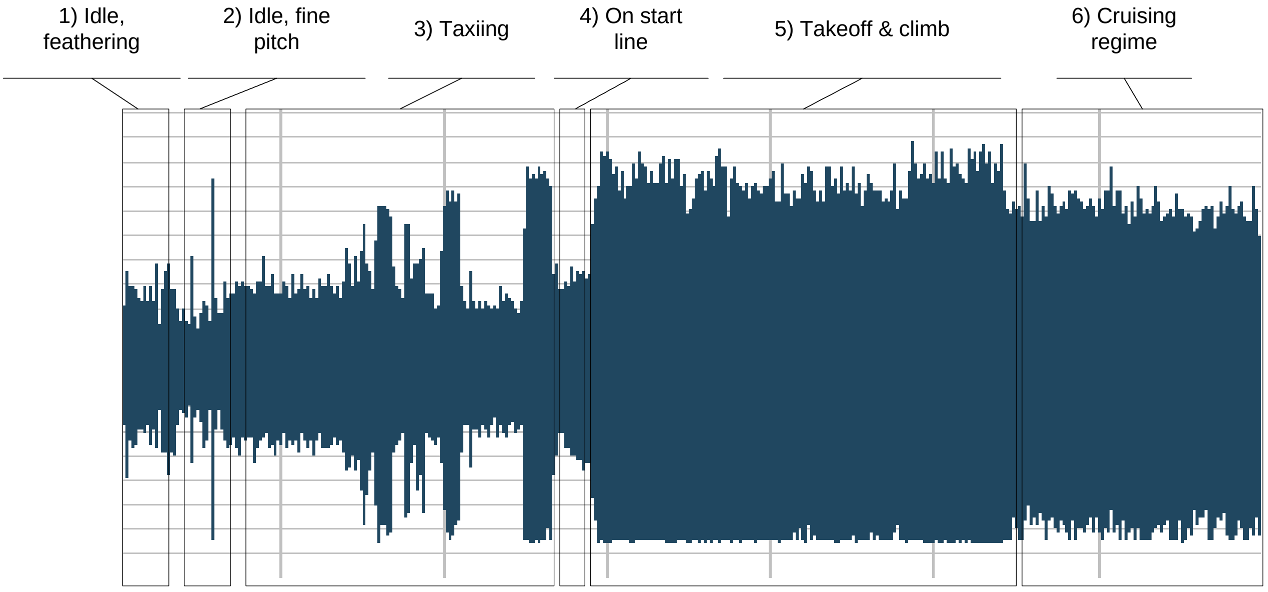


Figure 5. The airplane noise oscillogram.

Lacking noises, not extracted from the base record, where acquired from the open Internet resources, like “youtube” or specialized portal “avsim.su” [11]. The result was the complete set of noises, including electricity converters, fuel pumps, engine fire, blade pitch mechanism, wheels touch and screetch and so on.

Thereby, the proposed approach and special program constructor enabled to recover all noise features of the legacy simulator and extended their with noises of additional components.

Noise simulator constructor. The noise simulator constructor is the software tool, developed for the proposed approach that provides the following:

- creation of the random number of the noise players;
- association the noise samples with players;
- definition of the dependencies of the noise players with the aircraft model parameters by the piecewise-linear approximation;
- tuning of the additional playback effects;
- noise players testing;
- GUI for the noise simulator developers;
- complex noise imitation testing with the parameters receiving;
- simulator compilation to the ready to use console application.

The constructor is implemented on the .NET platform by the C# language (fig.6). Software architecture is based on the «Model–View–Presenter» [12]. The division of model and presentation requires additional development effort but enables the separation the model and GUI, simplifing the testing and permiting to replace the GUI or model components.

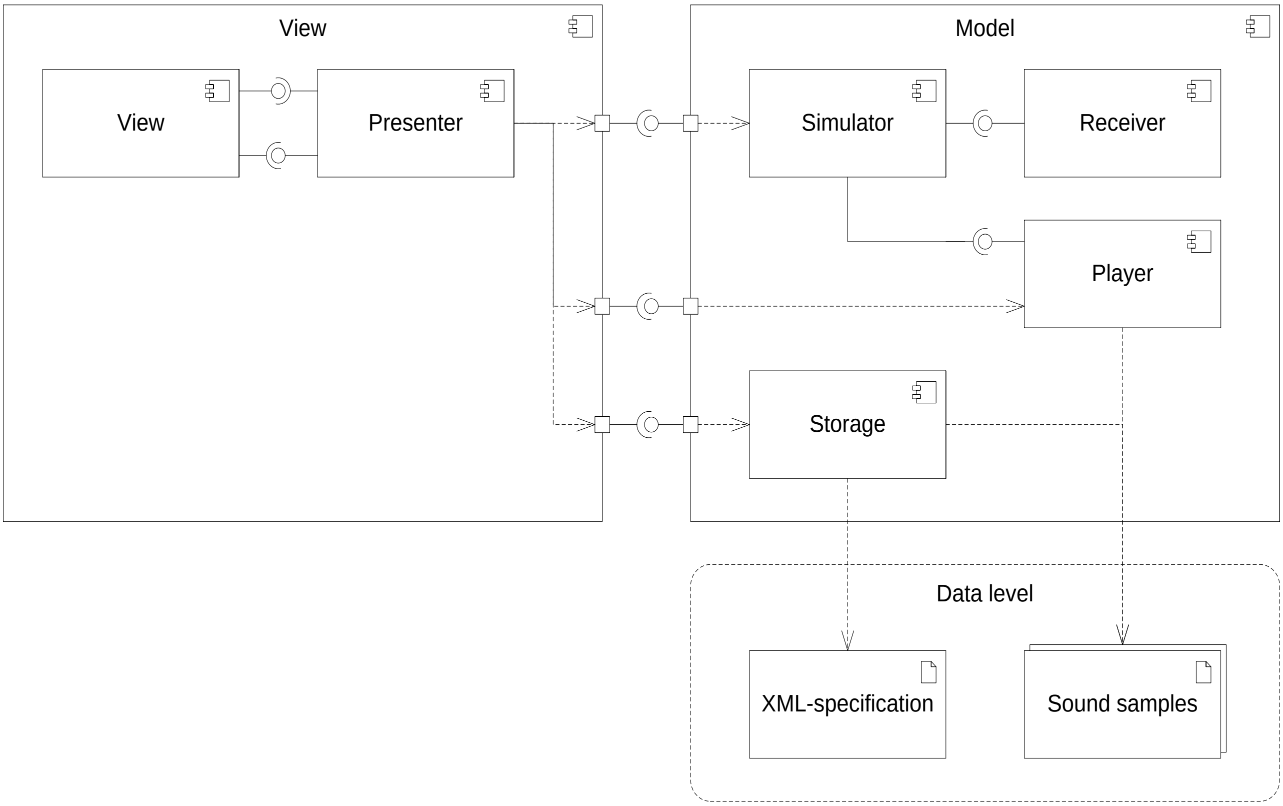


Figure 6. Architecture of the noise simulator constructor software

The simulator specification file example is shown on the fig. 7 and includes the airplane right air screw noise description.

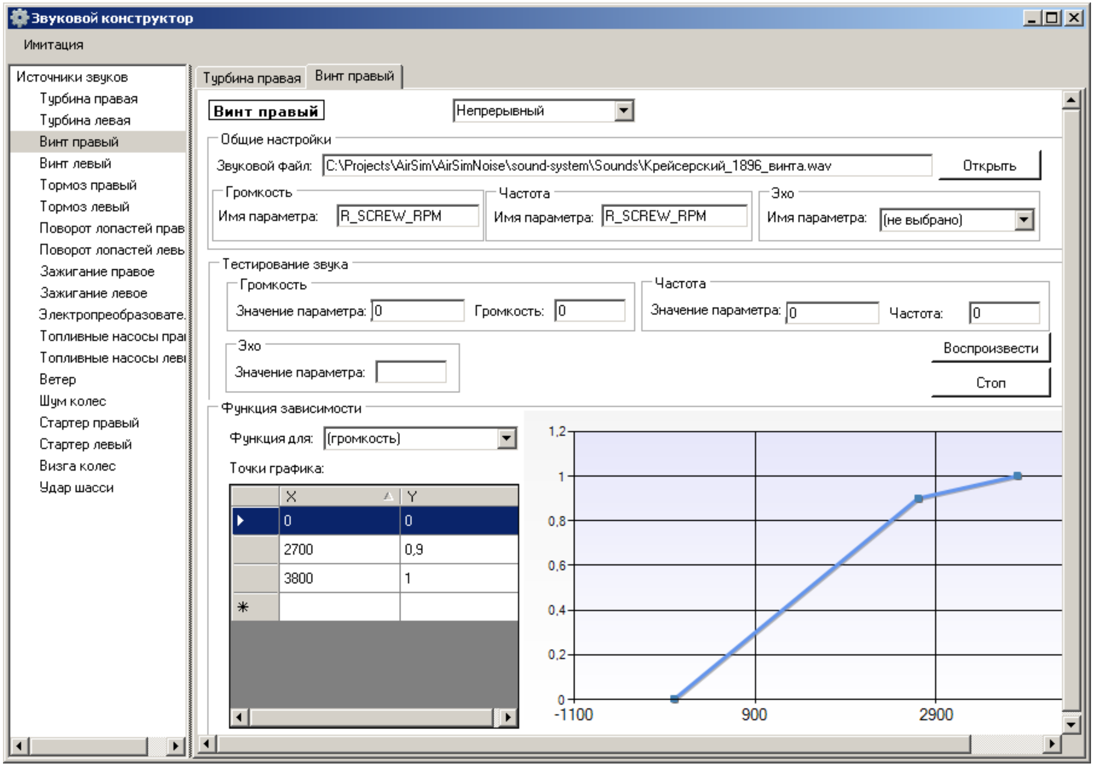
```
            …
<sound name="Right screw" type="Continuous">
    <!—Sample file -->
    <soundFile>C:\sound-system\Sounds\Screw_1896.wav</soundFile>
    <!—Sound channel (Left/Right) -->
    <position>"Right"</position>
    <!--  Set of input volume parameters -->
     <volumeParameters>
      <parameter name ="R_SCREW_RPM">
      <!--  Volume approximation points table for the screw rotation -->
     <tableOfValues>
          <entry x ="0.0" y ="0.0"/>
          <entry x ="2700" y ="0.9"/>
          <entry x ="3800" y ="1.0"/>
        </tableOfValues>
      </parameter>
      <parameter name ="R_SCREW_PITCH">
      <!--  Volume approximation points table for the screw pitch -->
        <tableOfValues>
          <entry x ="0" y ="0.5"/>
          <entry x ="90" y ="1.0"/>
        </tableOfValues>
      </parameter>
    </volumeParameters>
     <!—Playback speed input parameter -->
    <frequencyParameter name ="R_SCREW_RPM">
     <!--  Frequency approximation points table for the screw rotation -->
    <tableOfValues>
        <entry x ="0" y ="0.0"/>
        <entry x ="1896" y ="1.0"/>
        <entry x ="3800" y ="2.0"/>
      </tableOfValues>
    </frequencyParameter>
  </sound>
            …
```

Figure 7. The part of the simulator specification XML-file

Simulator model is realizing the initialization and functioning of simulator and includes the samples players, control module and approximator.

The GUI permits to engineer to add/delete the noise sources, define their attributes and dependencies, and test them (fig. 8).

The parameters’ driver receives the required aircraft model parameters and notifies the simulator model about changes.



Винт правый   Непрерывный

Общие настройки

Звуковой файл: C:\Projects\AirSim\AirSimNoise\sound-system\Sounds\Крейсерский_1896_винта.wav

Громкость — Имя параметра: R_SCREW_RPM

Частота — Имя параметра: R_SCREW_RPM

Эхо — Имя параметра: (не выбрано)

Тестирование звука

Громкость — Значение параметра: 0   Громкость: 0

Частота — Значение параметра: 0   Частота: 0

Эхо — Значение параметра:

Функция зависимости

Функция для: (громкость)

Точки графика:

X	Y
0	0
2700	0,9
3800	1

Figure 8. GUI of the constructor

Parameters' network receiver provides the obtaining of the parameters values from the modeling computer through the network. Receiver realizes the exchange, using special protocol, saves the values and notifies the noise simulator about updates.

The constructor software tools enables to generate ready to use noise imitator as console application (fig.9).

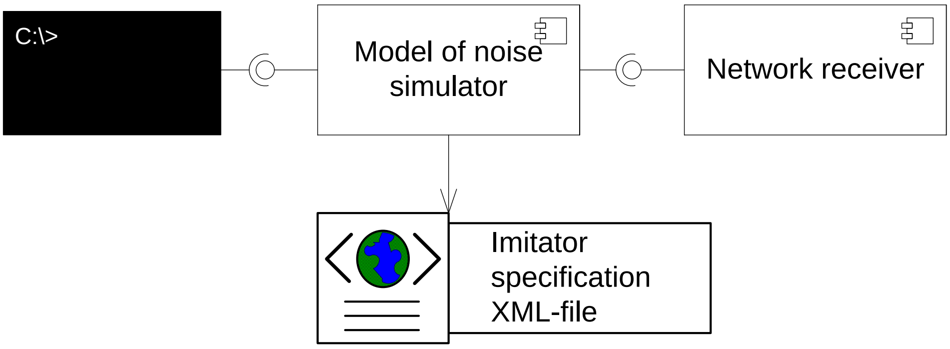


Figure 9. Generated noise simulator conponents

TL410 noise simulator adequacy evaluation. The simulator evaluation was executed by two methods –subjective assessment of the noise and comparing the oscillograms and spectrums of the generated and original airplane noises.

The subjective evaluation permits to detect the defects at the complex sound of several sources noises at the dynamic. The oscillogram and spectrums enables to check the relative volume levels for the various airplane modes and compare the adequacy of the spectral pictures.

Summaries

Formalization of the noise simulator enabled to create the software tools for the rapid construction of the flight simulator noise simulators, supporting their creation, testing and system integration.

The National Aviation University TL410 flight simulator reengineering case study demonstrates the approach and tool operability.

Approach requires the next research for the methods of the preliminary sample processing (influence filtering, spectral correction etc) and automation of the detection and correction of the samples interference.

References

1. Alfred T. Lee. Flight simulation: virtual environments in aviation. – Ashgate Publishing Limited. – 2005. – 147 p.
2. David Allerton. Principles of flight simulation. – John Wiley and Sons Ltd. – 2009. – 457p.
3. Design of a flight simulator software architecture. Göran Ancker, Jan Wallenberg. – School of Mathematics and Systems Engineering, Växjö University. – 2002. – 91 p.
4. Сидоров Н.А., Хоменко В.А., Недоводеев В.Т., Сердюк И.П. Реинженерия программного обеспечения информационно-моделирующих тренажерных комплексов. Управляющие системы и машины. – 2008. – № 4. – С.68-74.
5. Microsoft support site: How to locate the Flight Simulator 9 SDK – http://support.microsoft.com/kb/555857.
6. Сидоров М.О., Иванова Л.М., Хоменко В.А. Методологічні принципи реінженерії програмного забезпечення успадкованих авіаційних тренажерів // Мат. VIII Міжнар. наук.-техн. конф. „Авіа-2007”. – К.: 25-27 квітня 2007. – т.1, С. 13.119–13.122.
7. Сидоров Н.А. Хоменко В.А., Мендзебровский И.Б. Шаблон программного обеспечения устройств связи с объектом авиационных тренажеров. Проблемы программирования. –2008. – №2.3. – С. 239-248.
8. Web-site of the TL410 simulator of National aviation university – https://sites.google.com/site/tl410nau/home.
9. Руководство по летной эксплуатации самолета Л410 УВП-Э. Книга 1. – Министерство гражданской авиации СССР. – 1986. – 305 с.
10. Тренажер TL410. Техническое описание и инструкция по эксплуатации. Книга 5. – Praha, Rudy Letov. – 1979. – 75 с.
11. Audio and video records of the L410 airplane on resource AVSIM.SU – http://www.avsim.su/files.phtml?uploader=2860.
12. Fowler, Martin. Patterns of Enterprise Application Architecture . – Addison-Wesley Professional, 2002 – 560p.

Хоменко В.А. – к.т.н., доцент кафедри інженерії програмного забезпечення. Наякові інтереси: технології розробки та супроводження програмного забезпечення, прикладні домени і прикладне програмне забезпечення.

E-mail - vlkhomenko@ukr.net

Бикадоров А. – студент факультету комп’ютерних наук. Наукові інтереси: якість програмного забезпечення, прикладні домени і прикладне програмне забезпечення.

E-mail - AndrijL.Bykadorov@livenau.net

Стаття надійшла до редакції 26.10.2011 р.