

A Personal View on Teaching Csound

Gleb Rogozinsky
gleb.rogozinsky@gmail.com

Introduction

This paper describes the author's experience of teaching Csound over several years in Saint-Petersburg State University of Film and Television, Russia. The Csound language is given as a part of the course called Computer Music Technologies. This course lasts for two terms and includes several lectures and practical block. The whole course is new, so the large field for experimentation and research exists.

Saint-Petersburg State University of Film and Television was established in 1918 and during Soviet period was basically focused on technical education for film and television industry. After 1991, during modern Russia period, the University expanded its focus mostly to the field of arts. The institute of Screen Arts provides specialization in Film Directing, Camera Operating and Sound Production.

Author graduated the institute of Audiovisual Technics in 2006 and received a master diploma in Radio Engineering. The author's master thesis was about wavelet-based sound synthesis and processing. In 2010 author finished his candidate work about perceptual audio coding in wavelet domain and holds a candidate of sciences degree (equivalent to Ph.D.) in Audiovisual Processing Methods.

Author's first familiarity with Csound happened in 2005 during the course Digital Technologies of Audiovisual Production. The Csound was briefly observed in connection with MPEG-4 SASL (Structured Audio Score Language) and SAOL (Structured Audio Orchestra Language) which are the direct derivatives of Csound. Having special interest in electronic music, modern composition and synthesizers, author accepted an offer to read lectures for Sound Producers. Luckily granted with some degree of freedom, author chose sound synthesis as a main theme of his course with Csound as a language for formal description of different synthesis techniques.

Due to specialization's specifics and quite conservative attitude of headmasters, the further practical implementation of achieved skills in sound synthesis and Csound still stays uncommon, although there are some positive examples e.g. some students use synthesized soundscapes in their sound etudes and even in diploma film music. The students related to electronic music production unlikely to use Csound in their everyday job, although state that given knowledges in sound synthesis allow them to expand their working field from sample libraries to complex synthesis methods.

I. Initial Study Problems

After several years of teaching Csound, author outlines following issues that could be related to the initial study period: *background*, *interface*, *starting point* and *language*.

To develop such issues author should depict the typical portrait of his course and briefly describe the available equipment. Most students who chose such specialization as Sound Production have already graduated from musical colleges and sometimes even conservatories. This large category of students

would primarily like to achieve more technical skills in sound recording, mixing, production etc. Such students usually have serious problems with mathematics and sound processing but sometimes they are able to become even more successful than students with technical background due to having special interest in sound synthesis and computer music.

Students of other category already work at recording studios, clubs or post-production. Such students are rather nice group but often overrate their skills due to misunderstanding sound synthesis theory with intuitive knob twisting.

The last and the smallest category presented by technical-oriented students who for some reason decided to choose more artistic specialization. Quite often such students show the best results, although lack of artistic view in their projects.

The whole course Computer Musical Technologies takes two terms and includes 14 lectures and 8 practical studies. For practice the whole group (18 – 25 students) is divided into subgroups of 3-5 people. The lectures take place in the music studio, equipped with video projector and 5.1 sound system. Practical studies are held in the computer class with several PCs equipped with multimedia acoustics. Quite often, students prefer to use their own laptops. Besides that, there are MIDI interfaces of several types. Unfortunately, lab is not supplied with hardware synthesizers, so author sometimes uses his own Access Virus C, Waldorf microQ, Clavia Nord Modular G2, MicroKorg and Roland JP-8000 for demonstrating some features.

Background. As it was already mentioned, most students lack of technical education, so all preliminary notions e.g. analog-to-digital conversion, the existence of *a*- and *k*-rate signals, function tables etc., take much time to make all clear. Author uses a lot of graphics to describe anything and avoids formulas like Fourier transform or Nyquist theorem, preferring to discuss their basic ideas. Some useful metaphors could simplify the process. For example, understanding of *a*-rate or audio-rate signals is quite intuitive for everyone, but *k*-rate sometimes seems quite obscure in the presence of *a*-rate signals. The nature of *k*-rate signals becomes quite understandable when compared to automation signals like volume or pan automation. One cannot twist a rotary knob of filter cutoff frequency or mixing console crossfader with sonic rate, so there is no need of such high sampling as for *a*-rate.

Interface. Modern students are living in a point'n'click world so the text-based interface of Csound can and does cause a shocking effect on students who have never tried to program anything but changing tag fields when posting to social networks. During his lectures, author demonstrates a lot of examples and always outlines the usage of manual. It should be clear enough that great number of well-documented opcodes and examples as well as simplicity of instrument creation and connection makes Csound an ideal medium for sound synthesis learning. The important approach here is the conception of a network of primitives. The most accented moment is the existence of multiple modules; some with out(s), some with in(s). When discussing it, the author always draws some schemes of Csound instruments which consist of several modules interconnected with lines to show different data flow. By the end of such introduction, students become familiar with connecting virtually any modules with each other just by looking through the syntax given in the manual. Besides that, the author shows to his students the parallels with some other modular systems like Clavia Nord Modular by making the conception of modular systems clear and simple.

Starting point. We can talk here about “passive” learning when zombified student mechanically copies into his/her workbook or laptop the examples of instrument codes from the screen just to forget about it right after the lecture. To make it more effective, author asks his students to repeat given examples at home with some simple alterations e.g. varying opcode parameters or using another GEN routine. By default, author works with QuteCsound front-end to prevent any platform dependence difficulties, simplify interface issue and provide a number of built-in examples and compositions. Unfortunately, QuteCsound sometimes takes time to be set up in a proper way. Sure, there is no problem for a techie guy to set some parameters and adjust hardware buffer size for Real-Time work, but imagine a girl who

plays violin and uses her laptop only for checking e-mails and blog when she is trying to set up all of this. Moreover, some hardware configurations do not provide stable work. In such cases the author recommends using WinCsound that is solid stable and clear although most of the students prefer QuteCsound for opcode coloring, auto filling and built-in manual.

Language. While some students can freely speak English, most of them could hardly understand Csound manual completely without any help. Such sad fact prevents the author to give a lot of opcodes for home self-learning. So some time here should be spent on translating syntax and description. There are several Csound resources in the Internet which are in Russian, but the flexibility of Russian language and differences between programming, musical and acoustical thesauri often leads to uncertainty of notions and decreases overall understanding. Sometimes this is the worst initial issue due to its dead-end feature.

Tasks. There's no use of demonstrating initial exercises here, since they are quite clear and unsophisticated. As it was already mentioned, author asks to repeat instruments given on lectures and to substitute known opcodes with new ones e.g. correctly change *tone* to *butterlp* or *linseg* to *expseg* etc. This activity can be characterized as working out an ability to read the manual. Great variety of help material could be found in the first chapter of The Csound Book which is free to download from csounds.com [1], The FLOSS manual Csound pages [2] and Richard Boulanger's TOOTs [3].

II. Discussing Modulation

When initial issues are left behind and some kind of knowledge base begins to form in the minds of students, there goes the next level of difficulty. From the author's point of view, one of the most important things which should be discussed right after preliminary notions of any modular system is the modulation and some related notions like bias and LFO.

The main issue of this step is common problem with understanding the real nature of modulation. It should be clear that in the field of sound synthesis, every signal could modulate almost every parameter. The ones who begin to learn sound synthesis usually restrain themselves with only amplitude and frequency modulations. The author shows and discusses multiple examples of modulation chains, with increasing difficulty. From the author's point of view, some popular software or hardware synthesizers with a good modulation matrix could be great examples of source – destination chains. The author uses Access Virus C, Waldorf microQ as well as some software synths to demonstrate the flexibility of modulation including more bizarre cases like arpeggiator pattern modulation.

Getting familiar with modulation principles can help students in developing of difficult signals and better understanding the complex presets of virtually any synthesizer through reading modulation matrix. Finally, such knowledge could be really useful with FM synthesis, especially when analyzing patches which consist of several operators.

Tasks. Author gives a number of special tasks for working out the modulation principles and issues. They are basically targeted on finding out the source(s) and destination(s), the signals used for modulation, proper bias and levels through analyzing signals with some peculiar features. Figure 1 illustrates three examples of envelopes of signals to be created in Csound using complex modulation as a part of modulation signal practice. The sound signals inside the envelopes are arbitrary, although their generation could be a part of another task. Signal A envelope corresponds to series of pulses modulated by frequency using saw wave; signal B envelope corresponds to two-stage amplitude modulation with appropriate bias given; signal C envelope corresponds to amplitude modulation using saw wave which is modulated by frequency with pulse wave, the same pulse wave changes panorama of a stereo signal. One possible solution for third example using white noise as enveloped sound signal is given in Figure 2.

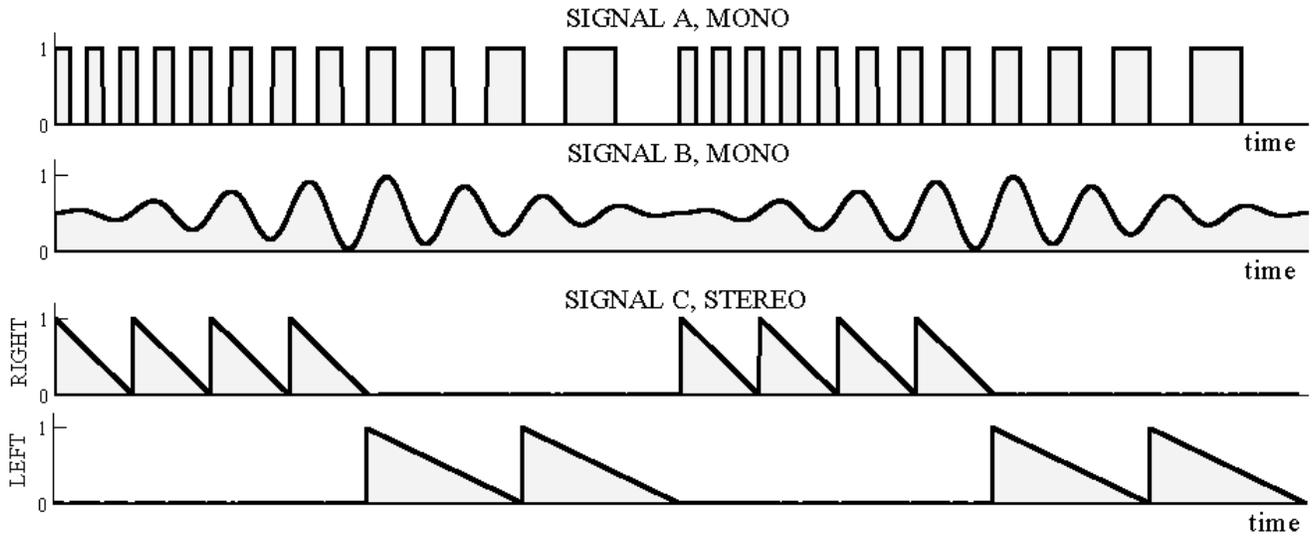


Figure 1. Example of signal envelopes given as a task for modulation practice. Signal A envelope corresponds to series of pulses modulated by frequency using saw wave; signal B envelope corresponds to two-stage amplitude modulation with appropriate bias given; signal C envelope corresponds to amplitude modulation using saw wave which is modulated by frequency with pulse wave, the same pulse wave changes panorama of a stereo signal.

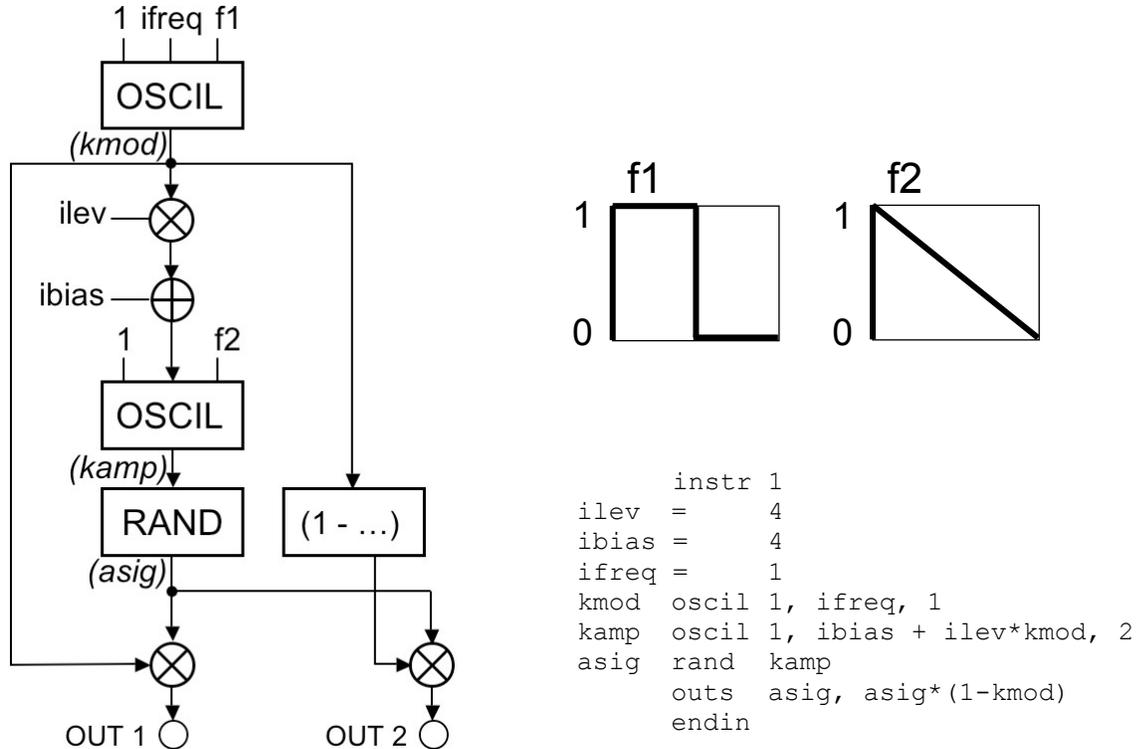


Figure 2. Block diagram and orchestra code of *instr 1* for signal C

III. Visualization of Signals

The examples of alternative music score notation e.g. György Ligeti's Artikulation or Karlheinz Stockhausen's Studie II, prove that electronic sound demands its own language of description. One of the best solutions for representing synthesized sound with dynamic behavior in the most informative way is a spectrogram, also known as a sonogram. A spectrogram is resulting from time-frequency transform i.e. windowed or short-time Fourier transform. It could provide both temporal and spectral information about analyzed signal. Some signals, which features seem to be quite obscure in time domain, could be easily analyzed in time-frequency domain by using spectrogram.

Author treats visualization of signals as an important tool for sound design. The approach is to imagine a signal in time-frequency domain before beginning to synthesize it. When discussing sound design in class, author recommends to use following steps: to fix desirable sound in memory, to draw corresponding amplitude envelope (answering questions about how fast is attack, how long does sound last, does it have a long release, does it have any other peculiar points) and to draw its spectrogram for understanding the dynamics of frequency components. When the shape of sound is fixed in such way, it is time for selecting a proper technique for its realization.

Almost any musical editing program provides means for calculating a signal's spectrogram and could be used by students. While Csound and QuteCsound also include some means for signal visualization, most of author's students prefer to use Audacity for getting waveforms and spectrograms. It is worth mentioning, that some students begin their electronic compositions from graphical scores, which are in most cases related to time-frequency domain. Figure 3 illustrates a spectrogram of one of my students' granular composition based on graphical score.

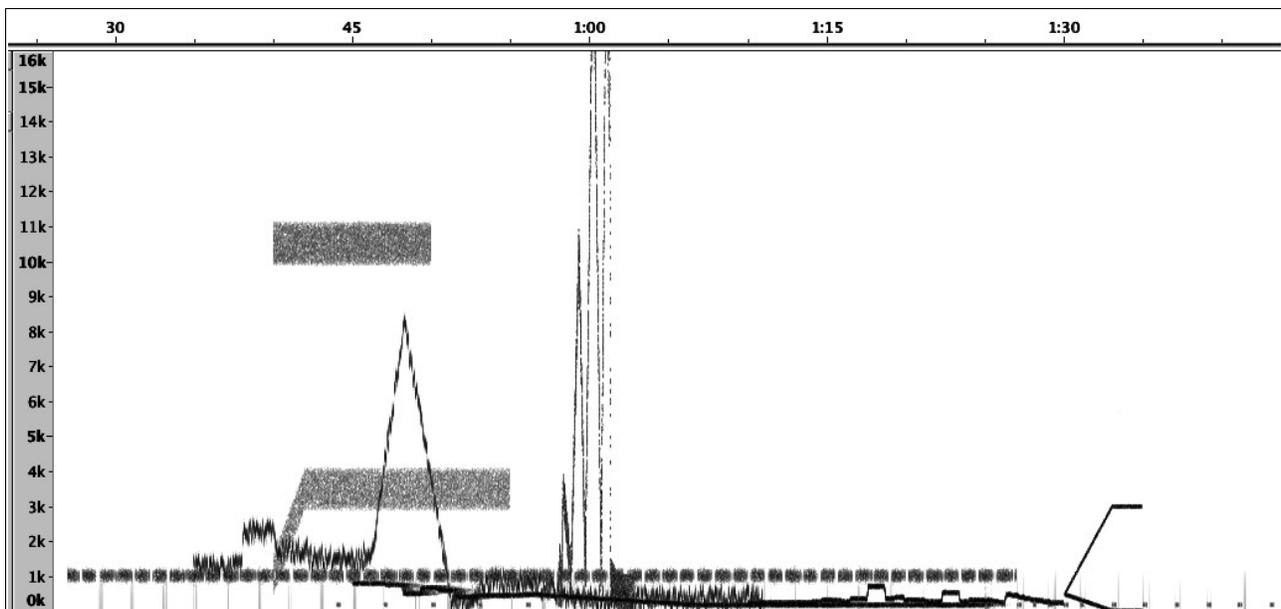


Figure 3. Audacity spectrogram of student's work, based on graphical notation

Tasks. Figure 4 illustrates the example of two tasks combined in one audio file. There are simple signals with frequency domain dynamics. The student should create a similar signals in Csound using given waveforms and spectrograms. These graphs should be analyzed to discover what kind of modulation was used, what is the source of modulation, does any kind of filtering exist and so on.

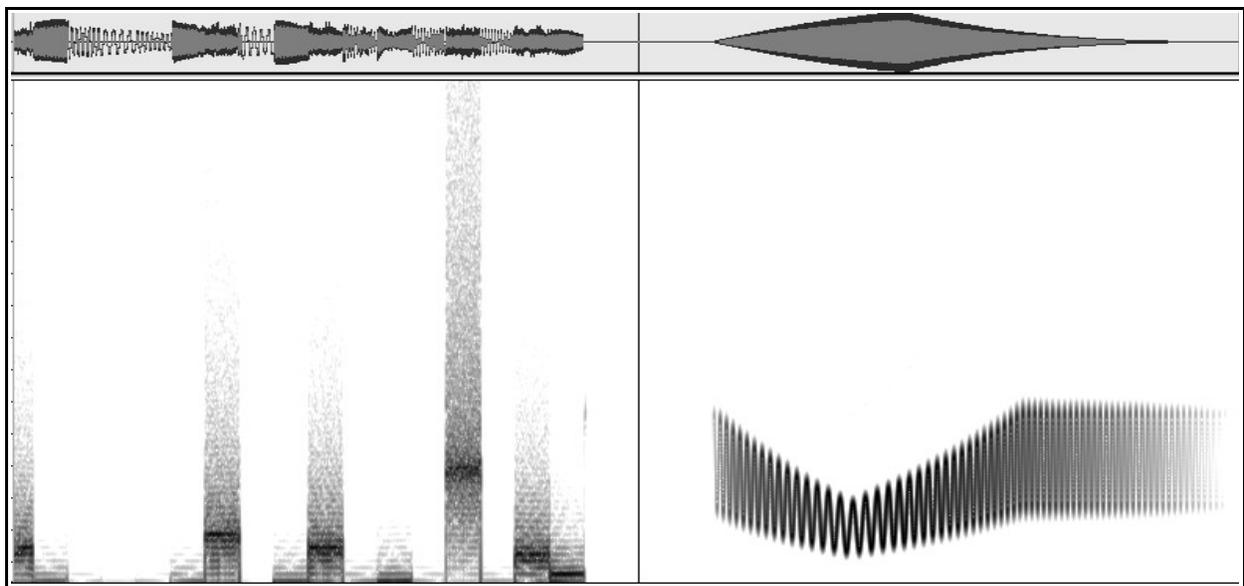


Figure 4. The example of two signals and corresponding spectrograms to be synthesized in Csound. The left one is a white noise filtered by low-pass filter which cutoff frequency is modulated by sample-and-hold LFO. The right one is a tonal signal with exponent AR-envelope, which modulates frequencies of the signal and its vibrato.

IV. Expanding the Boundaries

Author suggests the existence of number of directions to which basic knowledge of sound synthesis could be expanded: improving opcodes thesaurus, electronic composition analysis, discovering world of commercial synthesizers and other modular systems.

Improving opcodes thesaurus. Each lesson should give students another set of appropriate functions depending on the topic to be discussed, e.g. *rand* opcode, *GEN7* with few words about Gibbs phenomenon, different Csound filters opcodes when discussing subtractive synthesis; *GEN1*, *soundin*, *poscil*, soundfont format opcodes when discussing sampling etc. As a part of control task, each student have to discover some modules not discussed in lectures and make a couple examples of their usage.

Analyzing electronic music. To learn more about world of electronic music, author recommends his students listening to modern electronic compositions and to match the known classes of synthesized timbres, to find out typical sounds and common functions of synthesized instruments. It is important for students to analyze Csound instruments made by other people. There are several electronic compositions included in QuteCsound, i.e. Terry Riley's *In C* and John Chowning's *Stria*, performance-ready.

Discovering commercial synthesizers. Discussing classic synthesis techniques becomes more realistic and colorful since there are a lot of instruments described, modeled, and analyzed (such as the Hammond organ, Moog, Yamaha DX7, Mellotron). For example, an indisputable cornerstone of acid house-related styles, the Roland TB-303 with only a few parameters but having not-so-easy dependencies between them could be quite challenging for emulation. In some cases students could firstly analyze the rate of cutoff variation, the result of accented note, etc., before creating a block diagram of model and then starting to emulate its sound with the means of Csound. Another good example to be emulated is Roland Juno Super-saw digital oscillator, which gave birth to the famous Hoover sound and related family of sounds still popular today. One of the big Csound's advantage here

is the existence of The Csound Catalog[4] which includes a lot of models of commercial synthesizers. Some instruments from this catalog can be found on Data CD that goes with The Csound Book[5]. Another appropriate sources of information are Csound Journal[6] and Csound Ezine[7].

Tasks. Working out the modeling of classical synthesizers could be done with a number of multi-grade tasks depending on level of classic synth emulation. I.e., for emulation of the basic principles of additive synthesis used in Hammond organ, it is enough to connect 9 harmonic oscillators tuned to appropriate frequencies. The next step is to emulate real Hammond volume envelopes. The next difficulty level task could be gained on Leslie speaker and interleakage emulation. Since overall level of students' capabilities could fluctuate, the difficulty of emulation depends on it.

Usually this kind of practice task takes part in computer class with a subgroup of students. For this task students get a lot of information about device, its front panel, samples of its typical sounds, and sometimes good VST emulation. Author do not asks to finish the task at any cost. The main accent is on the understanding of every line of the code and every modulation chain, which should be meaningful and reasonable. This is important for developing algorithmic vision of a process.

V. Beyond the Horizon

In the final part of the Csound course, when students are able to understand syntax and create their own sounds, author makes an accent on widening the horizon of student's knowledge. More difficult synthesis techniques could be discussed in brief form, giving an ability for those who are interested to learn it by themselves and possibly show off their sound synthesis experiments they have learned from their practice. Granular synthesis techniques (and related ones like FoF, trainlet, pulsar, wavelet etc.) are always intriguing for students. Granular-like works of Iannis Xenakis, Barry Truax and Curtis Roads are nice examples on illustrating grain techniques and their expressive features. The fundamental work of Curtis Roads[8] could give great information on granular synthesis beginning from historical aspects to practical realization.

Below is an example of week-by-week schedule. All lectures last for 4 academic hours. The name of the lecture corresponds to its main topic which is discussed during the first half. In the rest part of the lecture author usually shows some new Csound opcodes and corresponding examples with discussing them and answering questions from auditory, so the theory is illustrated with examples at once.

Term	Part	Week	Lectures and practice
First term (Spring)	Initial Study	1	Lecture 1A. Introduction to sound synthesis and Csound
		2	Lecture 2A. Simple work in Csound
		3	Lecture 3A. Modulation
		4	Lecture 4A. Additive synthesis Practice 1A. First steps in Csound. Making simple instruments.
		5	Lecture 5A. Subtractive synthesis Practice 2A. Exploring modulation and filtering.
	Intermediate Study	6	Lecture 6A. FM-synthesis Practice 3A. Discovering FM-synthesis
		7	Lecture 7A. MIDI, OSC and Real-Time work in Csound Practice 4A. Real-Time work in Csound. Using MIDI and OSC in Csound.
		8	Lecture 8A. Sampling
		9-10	End-of-Term testing

Term	Part	Week	Lectures and practice
Second term (Fall)	Advanced Study	1	Lecture 1B. Commercial synthesizers Practice 1B. Making commercial synth model I
		2	Lecture 2B. Physical modeling Practice 2B. Making commercial synth model II
		3	Lecture 3B. Granular synthesis I Practice 3B. Working with granular synthesis
		4	Lecture 4B. Granular synthesis II Practice 4B. Discussing projects
		5	Lecture 5B. Algorithmic composition
		6	Lecture 6B. World of modular systems
		7-10	End-of-Term testing. Projects

There is no separate lecture devoted to such important thing as usage of Csound effects like reverberation or vocoding, but they are discussed during practice work and some examples are given on lectures. It is worth mentioning that there are separate subjects devoted mostly to signal processing.

To successfully pass the course students have to do several projects besides their casual homework tasks. During the first term every student must create a soundscape etude with duration about 1 min completely in Csound. This etude should include several instruments with modulation, filtering and known synthesis methods. The main course project is done by the end of second term. To avoid the lost of interest caused by undesirable project task, author always offers to make a choice. Students are free to re-group for choosing one or another task though the size of each group is limited. The typical project tasks are:

- *to remake some electronic composition with rather distinctive electronic timbres.* Usually this task assumes taking some instrumental electronic composition or soundtrack and deconstructing it to separate Csound instruments. The instruments created during this work should be described and discussed in class.

- *to translate a classical excerpt into electronic domain.* The student can take some classical composition according to his/her taste e.g. Elsa's Procession to Cathedral from Wagner's Lohengrin or Shostakovich's Violin Concerto and create an electronic orchestra of Csound instruments to play it. The accent here falls on the dramatic role of each instrument. The instruments by themselves are not required to sound like real ones, but to reflect the dramatic relation of original composition.

- *to create an inter-computer performance with OSC-based message transporting.* Students who are focused on technical aspects or performances used to take this task. At least one computer of the network should use Csound. This task could be done in real-time with some controllers. It is up to students to decide what to control and how to control. The sound design aspect here is not primary goal.

- *to create an algorithmic model or total serial model with description and short score example.* This is a task mostly for those who do not happen to make a friendship with Csound for some reason. Usually it is preferred by conservatory graduates if any.

During their concert or presentation, which is open for everyone, students have to demonstrate block diagrams of their Csound instruments, graphical scores or other material they used when working on projects, accompanying it with brief analysis.

Most students interested in specialization have strong musical and artistic backgrounds, so the final period is always the most interesting part of the course. Comparing to western educational system, it is not common for Russian universities to allow selection of subjects according to student's interest so after finishing Csound classes students do not have any possibility to choose something close to it. On the other side, all of them have to start course Synthesized Sound in Film Production. This course has more artistic context and focused mainly on discussing electro-acoustical sound application for the purposes of film production with analysis of number of well-known Russian and Western movies with noticeable use of electronic sound.

Conclusion

The main goals of discussed course are to make an introduction to sound synthesis techniques and Csound programming. During this course students have to do several tasks of which some were described in paper, including final tasks for group's concert.

The author should mention that some percent of students do not pass the subject. On the other hand, most of such students systematically fail the other subjects too so their poor progress seems to be behind the Csound-specific difficulties. Author suggests that existing variance of project tasks leads to progress increasing since students are able to select project in accordance to their level.

In conclusion, the author hopes that discussed approaches could be useful for everyone who teaches and learns Csound. Having both technical and musical education, the author understands some possible difficulties which may prevent students from going deeper into Csound universe (and the whole world of sound synthesis). However, if several important factors like those discussed above are cleared the barriers will fall.

References

- [1] <http://www.csounds.com/chapter1/> Richard Boulanger's Introduction to Sound Design in Csound.
- [2] <http://en.flossmanuals.net/csound/index/> The FLOSS Manual Csound Pages.
- [3] <http://www.csounds.com/toots/index.html> Richard Boulanger's Csound TOOTs.
- [4] <http://www.csounds.com/shop/csound-catalog> The Csound Catalog.
- [5] The Csound Book, Cambridge: MIT Press, 2000.
- [6] <http://www.csounds.com/journal/> The Csound journal.
- [7] <http://csounds.com/ezine/> The Csound Magazine.
- [8] Roads, Curtis. *Microsound*. Cambridge: MIT Press, 2001.