Real-Time CSOUND:
Software Synthesis with Sensing and Control

Barry Vercoe
Media Laboratory, Room E15-490
M.I.T.
Cambridge, MA 02139
U.S.A.
bv@media-lab.media.mit.edu

Dan Ellis
Media Laboratory, Room E15-491
M.I.T.
Cambridge, MA 02139
U.S.A.
dpwe@media-lab.media.mit.edu

ABSTRACT: A new version of Csound is released which brings it into line with computing for the 90's. The familiar audio-processing and music synthesis package has recently been extended and reshaped to put more of the emerging power of computers into the hands of composers and researchers. Audio synthesis has been expanded with phase vocoder and FOF synthesis, and instruments can perform analysis of incoming audio and gestural signals. The software has been re-tailored for real-time operation on sufficiently powerful machines, and the package has been made portable to a wide variety of hosts.

1. Introduction and goals

Csound is a comprehensive digital audio processing package in portable C, designed to provide composers and researchers with a ready-made, flexible environment which supports creativity and research. Initially written in the Fall of '85, and widely used in that original form, Csound has now been restructured for a new decade of musical computing. The goals and current status are:

Expansion of the audio processing vocabulary. The package now includes a phase vocoder and FOF formant synthesizer, complementing the former array of filtering, linear-prediction and additive synthesis techniques. The user can optionally invoke the X-11 window system for detailed viewing of the content of instrument signal-paths.

Run-time sensing and control. The traditional concept of an instrument is expanded to include in-line analysis of incoming audio and gestural signals. Instruments can perform spectral analysis (linear or log-frequency), pitch detection and event detection, then use this to influence output event generation and control.

Real-time operation. The advent of RISC-based desktop computing brings software analysis/synthesis into the arena of realtime computer music performance. Combining the flexibility of software synthesis with the directness of realtime interactive performance has induced an expanded set of input paths and internal protocols.

Portability. The software is maximally portable, with host-system and host-software dependencies isolated in a few modules. Compile-time *make* options and command-line flags provide an easy interface to the most common vendor file-systems and audio I/O devices. Csound currently runs on DEC-Vax, DEC-RISC, Sun and NeXT workstations under Unix, and on Macintosh under Think_C.

2. Hardware lessons for real-time software synthesis

Launching desktop software synthesis into the realtime domain is not a step to be taken lightly. An important influence has come from our recent development of a realtime DSP system [Boynton & Cumming, 1988; Peterson, 1990], which puts 100 mips of 48KHz 8-channel audio processing into a MacII. Although primarily developed for in-house research, this MC56000-based system also occasioned the development of a near-complete version of Csound running in realtime [Vercoe, 1989]. It is important to note that this is an "idealized world" with respect to realtime synthesis: the hard realtime audio deadlines are clearly defined and understandable, and the distinction between the world of asynchronous, interrupt-noisy host processing and that of near lock-step synchronous multi-processing on the DSP boards is constantly reinforced by the reality of Nubus transfers. (The two worlds communicate via buffered message-passing.)

The growth of a hardware-specific Csound for this environment was speedy and informing, and most of the familiar data-paths and unit-generators were quickly implemented. Some units were simplified for realtime reasons: FOF synthesis did not include the octaviation concepts of Clarke (see below). Also, operations not contributing to sample synthesis (e.g. calculation of RESON coefficients during CF motion) were placed in a slower control loop running on the MacII host (this j-rate control is asynchronous with the k-rate paths within instruments). However, the realtime experience was important in hatching a new class of building blocks focussed on *realtime sensing and control*. Most notable were a stable pitch-tracker (live voice or instruments in score-following and synchronized accompaniment, see also [Vercoe & Puckette, 1985; Vercoe, 1990]), and envelope concepts that permit unscheduled onset and release if played from a MIDI keyboard.

The hardware multi-processor is an attractive platform for future research, and we have made good progress on how to distribute the load of unscheduled events [Peterson, 1990]. However, until multiprocessor systems are widely available, and vendor software mature, this approach will be frustrating to musicians because of almost annual obsolescence and continual re-implementation. As one way of beating this, the methods of realtime sensing and control that grew from our DSP experience have been ported almost intact into Csound. The DSP hardware version has also helped suggest how a realtime software Csound might possibly survive in a Unix-like environment.

3. Extension of the software signal processing vocabulary

The new software version includes several new synthesis modules, two of which are detailed here. A C version of IRCAM's FOF (formant wave function) algorithm was contributed by Michael Clark of Huddersfield Polytechnic. This algorithm allows sound construction from a controlled *formant* spectrum independent of pitch. Several FOF generators in parallel produces a sound with a precise, complex formant structure. FOF is thus well suited to human voice synthesis, along with other possibilities. The Csound unit is based on the elemental generators of IRCAM's *chant* program, but includes Clarke's extensions of formant modulation and octaviation [Clarke, 1988].

A phase vocoder, similar to the [Dolson, 1986] program yet distinct in implementation, is now incorporated into Csound. It uses a one-pass short-time fourier transform *pre-analysis* of a sound segment. At run-time, *pvoc* reads the entire analysis into memory, then interpolates and warps the fourier spectra to resynthesize a modified sound. Modification is controlled by a *time pointer* into the analysis file and a *pitch transposition index* (to one octave up or down), permitting the pitch and time base of the original sample to be independently and arbitrarily varied. The time pointer (like that in Csound's LPC) is an arbitrary control-signal, enabling such exotic effects as smooth loops and discontinuous time mappings. Although the frequency resolution is fixed at analysis time (32 to 2048 frequency bins), the resynthesis frame rate is determined within the orchestra and can thus be varied between runs without repeated analysis. This is made possible by oversampling the original spectra in time, and interpolating these frames during resynthesis.

4. Run-time sensing and control

Signal processing in real time offers an alternative source for instrument control signals. A recent addition is spectral analysis of live audio, whose pitch or spectral content can control the evolution of synthesized voices. These units typically employ successive octave downsampling and discrete fourier analysis (constant Q, exponentially spaced in each octave) whose rate of change, say, can be used as a beat detector to control the time-warping of an uninterpreted scorefile [Vercoe, 1990].

Such control concepts have caused a broad expansion of available data paths. Ascii score events appearing on *standard input* (from the console or some other process) are injected directly into the performance stream. Instruments can also be played from a MIDI keyboard, and instruments can in turn generate MIDI output. Audio output can be piped from one Csound process to another, or redirected to */dev/audio* for realtime audition. We have explored these aspects extensively on a DEC5000/200 workstation (25 mips RISC machine) and have found the flexibility quite enabling.

In a different kind of application (currently running at the Media Lab on a SUN Sparcstation), a graphic *drum-set* is played using a 3-D hand-position sensor, and the appropriate partials (based on the elastic properties of the objects experiencing collision and restoration) are conveyed via *stdin* to a Csound realtime additive synthesis instrument.

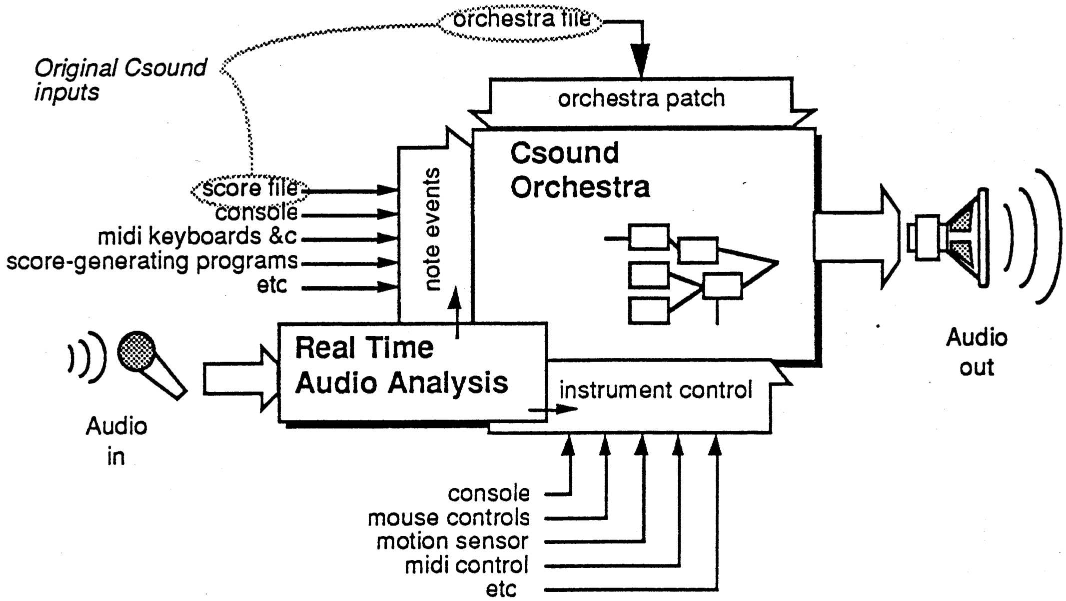

Fig 1 The new Csound

5. Status and availability

The new Csound is currently quite robust, but is not expected to become stable or definitive while realtime control remains the subject of much experiment. In some ways this version is simpler: while the earlier Csound used system overlays in limited memory, the revision is a single module with fewer system calls and is thus easier to port. It makes no assumptions about local soundfile formats, and connects to the host environment via compiler options and command-line flags (e.g. to a SUN 4.1 multimedia file, or directly to its /dev/audio). A Mac installation creates Sound Designer II files, suitable for either Digidesign's Accelerator board or an AudioMedia board; if given a -c flag it will produce character samples for your cheap on-board speaker.

Scores and orchestras developed under version 1 will run without change. As fast or realtime software synthesis becomes increasingly possible, the free exchange of instrument and score files becomes worthwhile. The efforts of Richard Boulanger in this regard merit acknowledgement.

Csound is distributed by ftp or floppy disk. The usual educational and non-profit restrictions apply.

6. References

Boynton, L. & D. Cumming. "A Real-Time Acoustic Processing Card for the Mac II," ICMC *Proceedings* 1988, pp. 349-356.

Clarke, M., P. Manning, R. Berry & A. Purvis. "VOCEL: New implementations of the FOF synthesis method, " ICMC *Proceedings* 1988, pp. 357-371.

Dolson, M. "The Phase Vocoder: A Tutorial," *Computer Music Journal,* 10(4), 1986, pp. 14-27.

Peterson, K. *Psuedo-Static Scheduling of Music Processing Algorithms for a Small-Scale Multiprocessor,* M.S. Thesis, M.I.T. Media Lab & Dept of Electrical Engineering, August, 1990.

Vercoe, B. & M. Puckette, "Synthetic Rehearsal: Training the Synthetic Performer, ICMC *Proceedings* 1985, pp. 275-278.

Vercoe, B. *The MacII Real-Time Audio Processor: Reference Manual,* MIT Media Lab, 1989.

Vercoe, B. "A Realtime Auditory Model of Rhythm Perception and Cognition," Internat. Conf. on Music and the Cognitive Sciences (Cambridge, England), *Proceedings,* 1990.