Audio Time Companion for Studio and Performance Synchronization

Marc LoCascio
MIT Media Laboratory
Cambridge, MA
nbl@ems.media.mit.edu

ABSTRACT

Recent research has provided systems that play musical accompaniment on real-time synthesizers while following the performance of human soloists. Another solution to electronic music's problem of syncing live players to pre-recorded tapes is a system using a time code synchronized tape deck whose audio output is pitch corrected for any tape speed deviation. A frequency shifter with SMPTE time code input is required for this task. With a computer-based, variable-rate time code generator, a musical performance can be compared against an internal representation of its score and external devices—be they audio or video tape machines, video disk players, or MIDI synthesizers—synchronized to the live action. Two foreseeable uses of this interactive time code generator will be the freeing of musical sequencing tasks from the host computer in "synthetic performer"—allowing the computer to concentrate on pitch tracking and score following while sequencing and synthesis is accomplished by dedicated MIDI machines—and the synchronization of visual images—such as from video disk—to live musical performance. The complete system of time code generator and reader/pitch shifter will be outlined in this paper, and other uses for the technology developed therein will be suggested.

1. Introduction: Project Motivations

The master versus slave issue has long been a problem in audio and video synchronization. In the video production process, audio tape machines are usually slaved to video machines due to the relative simplicity of their servo mechanisms. Similarly, in MIDI sequencer to tape synchronization, MIDI devices are inevitably slaved to SMPTE time code from an audio tape.1 In live performance, human musicians until recently have been slaved to their MIDI and tape counterparts. With the coming of real-time computer score followers—generating time bases for synthesizer accompaniment to live performers—we finally see the tables turning on the human to MIDI sync problem [Vercue]. The human to audio tape issue is more difficult to solve, however. Since analog audio tapes would have to be speeded up or slowed down to follow a live performance, a pitch shifter must be devised which can track the tape's speed and provide the appropriate pitch compensation. This is the goal of this research. Also, some concept of non-real-time, or "virtual," SMPTE time code must be realized—a representation of a relative time in a score as opposed to a strict clock time. Such a time code generator connected to a computer music workstation running score following software can provide a flexible time stream for the synchronization of audio tapes or even video (such as from video disk) for the accompaniment of music.

1 This paper assumes a basic understanding of the SMPTE time code format. Confused readers should consult the ESCO, Wran, or similar references.

1987 ICMC Proceedings 249
1.1. The Live Music to Video Connection

Another inspiration for the implementation of virtual time code comes from visual artists. Filmmaker Richard Leacock and his students at the MIT Media Lab's Film/Video group expressed an interest in using the technology of pitch tracking and score following to synchronize video imagery to live performance in the fashion of Prof. Leacock's earlier work with the Metropolitan Opera Company. Since it is possible to sync synthesizers to human players, it should be possible to sync video disk imagery as well. A microcomuter based system using SMFTE time code readers and random access video disk players has already been implemented at the Film/Video group (Kanner) - a virtual time code generator interlocked to a score following routine is all that is needed to complete the performance system.

2. The Hardware

2.1. The Computer Music Workstation

The basic system for this project consists of a Hewlett-Packard Series 300 "Bobcat" computer equipped with custom built analog-digital converters, MIDI interface, and SMFTE time code reader/generator. The Bobcat is a microcomputer based on a Motorola 68020 processor with a 68881 math co-processor and 6 Mbytes of RAM, running an HP version of Bel Labs' System V UNIX with 4.2 BSD enhancements. Three 55.5 Mbyte Winchester disks on a high speed disk interface are used for sound and data storage. Four of the HP machines are on an ethernet link with the music studio's DEC VAX 11/750, VAXStation II, and Sun III— all running UNIX and the Csound music processing language. The ADC's and DAC's are 16 bit Sony converters with a 48 kilohertz sampling rate, and MIDI I/O is handled by a Roland MPU-401 (all connected to the Bobcats by in-house adaptor cards). The time code reader and generator construction is part of this project and, as such, will be presented in some detail.

2.1.1. The Time Code Reader

The SMFTE time code reader used in this project is an Otari I-0055 Time Code Reader IC interfaced to the HP-IB (IEEE-488)

Figure 1: The Media Lab computer music workstation

instrumentation bus of the Hewlett-Packard workstation. This SMFTE reader chip outputs BCD (binary coded decimal) hours, minutes, seconds, and frames data when requested by the host processor. Also generated is a clock corresponding to the time code bit clock. This signal can be counted alongside a crystal based clock to determine virtual time to sub-frame accuracy.

2.1.2. The Time Code Generator

The time code generator is microprocessor based and has the capability of producing variable-speed time code for the purpose of synchronizing external devices to the score matched time of a live performance. The SMFTE code has a virtual format of 30 frames per second, non-drop frame. The clock which controls the rate at which the time code is shifted out of the holding registers has an analog bias with a feedback loop to regulate the output speed. The workstation passes down a time compression or expansion (i.e. compression) factor to the SMFTE generator, which alters its clock rate accordingly. The success with which the virtual time code can be fol-
The Roland sync box follows approximately the same SMPTE code speed variation as the DASH machines and provides full chase mode synchronization to sequencers which respond correctly to MIDI song pointer, clock, and start/stop commands.

3. The Software

3.1. Time Compassion

The earliest attempts at constant pitch time compassion used audio recorders with rotating tape heads [Lee]. The spinning heads effectively created windows into the soundstream which were either looped or discarded depending on whether the sound was to be expanded or compressed. Essentially the same technique can be used in the digital domain for time manipulation. In the crudest manner, rectangular windows with fixed period can be taken from the audio signal and looped or discarded accordingly. Figure 4 shows the basic time versus sample index output relationship. A simple improvement on this technique uses sinusoidal windows with 50% overlap.

2.2. Audio Sources and Synchronizers

There are two audio recipients of our virtual time code—Sony professional tape decks with integral chase lock synchronizers and the Roland SRX-40 SMPTE to MIDI sync box. The Sony machines are the APR-5003 1/4" half track analog tape deck with center track time code and the PCM-3202 two track DASH (digital audio stationary head) recorder. The synchronizers in these machines provide full chase mode interlock but are susceptible to time code discontinuities (thus precluding virtual time code with added or dropped frames and a fixed clock rate). The DASH machines can track a plus or minus 15% speed variation before muting their output, while the analog machine will follow a much wider time variation. Both machines have rapid but hardly instantaneous transport servo. For this reason, the pitch shifter SMPTE input must read the time code from the audio tape itself instead of the master time code which the tape deck is chasing. Fortunately, both decks allow this.
3.2. Pitch Shifting

Simple audio time variation can be achieved with a variable speed tape deck or by playback of a digital soundfile at different sampling rates, and pitch shifting algorithms can be used to compensate for the subsequent pitch fluctuations. These algorithms are quite similar to those for time compression. Sound segments are repeated or stretched using a variable speed phase accumulator to determine the index into the look-up table of samples—with the resulting resampling providing the pitch change. Again, for the best audio quality, loops between windows should be synchronized at zero crossings of similar slope to reduce amplitude modulation problems. A variety of window sizes and shapes—as well as schemes for overlapping of multiple windows—will be implemented during the course of this research to determine their subjective fidelity. Simple examples of time compression and pitch lowering with 50% overlapping sinusoidal windows are given as figures 6a and 6b.

3.3. Alternate Methods

A variety of other techniques exists for the high fidelity time and pitch manipulation of sound, primarily outside of real-time. Audio analysis and resynthesis methods, such as the UCSD phase vocoder [Dolson], are capable of very "musical" signal processing, but they also require significant computing power and user input as to parameter definition. A new idea for high quality audio manipulation purely in the digital time domain has also been discussed. By marking the most reasonable loop points within a soundfile before playback time (i.e., steady state segments), the audio could later be compressed or expanded without arbitrarily discarding or blurring valuable transient information. This technique has not as yet been implemented.

4. Integration and Applications

4.1. Real-time Control

As UNIX is notoriously bad for real-time development, it is arguably best to keep to a minimum one’s expectations of the workstation itself and build as much power as possible into the hardware peripherals. The problems associated with real-time control have been addressed elsewhere [Puckette], so a cursory discussion of the specific factors in this application should suffice. The process scheduler

252 1987 ICMC Proceedings
Figure 6a: Simple time compression

Figure 6b: Simple pitch lowering
Figure 7: The "way cool" performance system
must respond as quickly as possible to messages signifying changes in tape speed from the SMPTE reader and pass requests for changes of virtual performance time from the score follower to the SMPTE generator. Tape speed variation, when detected, causes a proportional change to the phaser frequency at which samples are read from the look-up tables. Using real-time process priority, as well as memory locking and optimized C or 68020 assembler programming, we hope to achieve adequate timer accuracy and audio processing power for one channel of pitch correction (as a test case). In the long run, the system should be re-implemented with a high-speed DSP board to handle the number crunching for the host computer.

4.2. The Audio/Visual Studio System

The music workstation—equipped with the hardware and software modules previously described—is the center of a network of machines configurable in any number of ways for audio and video production. Virtual remotes for tape deck control can be implemented in mouse-driven computer graphics with an RS422 physical link. Direct data links can be established between digital audio tape machines, samplers, and workstations. SMPTE time code can be used for MIDI, audio, or video synchronization. Various computers of different types can be connected by a combination of ethernet, MIDI, and RS232/RS422 data transmission. Ultimately, this standardization and modularity provides a vast array of new tools for the musician at the price of moderate software development.

4.3. The "Way Cool" Performance System

The final application for this research is an audio/visual performance system in which the human artist acts as master to the computer accompanist. The virtual time code generated in response to live performers can be used to drive any number of audio, video, or MIDI slaves. In turn, SMPTE equipped audio decks can control pitch shifting processes on A-to-D-to-A converter equipped workstations. With such an open ended system, the creative possibilities become limited primarily by the talent of the user.

5. Acknowledgements

Many thanks are due to the people of the Media Lab for their inspiration and help. Particular thanks go to Miller Puckette, Barry Vercoe, and John Amendo for invaluable assistance with the signal processing end of things.

REFERENCES


Puckette, Miller, "A Score Following Algorithm," unpublished manuscript.

