ABSTRACT

The Keyboard Computer music System developed by the Computers and Music Group at the University of Waterloo is described, and existing capabilities outlined. This low-cost, microcomputer-based music system incorporates a multi-channel digital synthesizer, a microcomputer with floppy disks, a printer with graphics capabilities and a velocity-sensitive polyphonic "piano" keyboard. Some sequencer application areas for this system are presented, along with a list of envisaged uses.

A need for a conventional Music Editor is outlined, and use specifications for such an editor to be implemented on this system are presented.

986
1. INTRODUCTION

The Computer and Music Group In Systems Design Engineering at Waterloo is currently engaged in developing a microcomputer-based system which will support the exploration of a variety of music performance and learning applications. The equipment dedicated to this project consists of two expanded TRS-80 microcomputer systems, one used for software development, and the other controlling the Keyboard Computer Music system (see Fig. 1). To make available the additional characters required by the programming language C, running under the CP/M operating system, a Volker-Craig 404 terminal has been connected to the software development system. A light pen and a joystick are also available as input devices for either system. Output capabilities are provided by a DEC Diablo II printer with extended graphics capabilities. In addition, a Tektronix X-Y plotter is available, although, at present, it has not been incorporated into the actual KCM system.

Two special-purpose devices provide the music input and output capabilities for the KCM system. Input is provided by a polyphonic, velocity-sensitive "piano" keyboard which makes use of variable inductors as key position sensors and a time-multiplexed digital circuit to scan one keyboard for key events. Musical output is provided by a sixteen-channel digital oscillator which uses separate digital to analog converters for the waveform and amplitude functions, and performs the multiplication of these in the analog domain, with separate outputs available for each channel. Both of these devices were designed and built in-house. Design details can be found elsewhere in these proceedings [1,2].

At present, software exists which obtains, and records, input from the piano keyboard, and "plays it" through the digital oscillators in real time. The Keyboard Activity Code (KAC) provided by the piano keyboard interface consists of key number (0-88), direction of travel (up/down), time of event occurrence (in milliseconds), and the time taken for the key to pass through specified portion of its travel distance (used to determine key velocity). This information may be stored on files on disk, and read back into
memory for subsequent replays. By using only the KAC, it is possible to output the corresponding music in various waveform or different touch sensitivities.

An elementary aural training program, involving the presentation of ascending and descending melodic intervals within an augmented ninth, has also been implemented, with the digital oscillator providing the aural stimulus, and the student responses accepted from the typewriter keyboard. This package allows the music teacher to specify up to twenty different levels of difficulty, with a maximum of thirty-one musical intervals per level. The level being presented can be changed either manually, or under program control (based on user-defined criteria), at the user's discretion. Under the latter mode, students are presented, in a given session, with only as many intervals as they can comfortably handle. It is possible for each student using the system to have a different set of levels, as well as a different set of level-control criteria. Statistics are kept in such a way that specific intervals which are causing problems for a student can be given special emphasis.

2. APPLICATION AREAS

In addition to the two applications outlined above, many other uses are projected for the Keyboard Computer Music (KCM) system. As mentioned, it is possible to use the system strictly as a new type of musical instrument. Work is in progress to allow a musician to specify to the computer exactly how the sound is to be controlled. It is envisioned that it will prove possible to simulate the sound of some existing instruments realistically, since the presence of a computer in the control loop will allow the harmonics of a tone to be individually controlled in time, based, if so desired, upon the velocity of key depression (or even release). In addition, the system will, of course, be able to produce a wide variety of "new musically useful" sounds not presently realizable on standard acoustic or electronic instruments.
Due to the nature of the keyboard information obtained, the KCM system can be utilized by composers and arrangers. By varying the control parameters, it will prove possible to hear the same material in a variety of voicings. In this way, it will not be necessary to have available a variety of instrumentalists in order to determine (approximately) how a particular composition (or arrangement) will sound. In addition, work is in progress to produce musical score from a file of keyboard Activity Code, so that it will not be necessary to perform the tedious task of transcribing a given piece of music. The KCM system, at present, is capable of producing acceptable quality hardcopy score for a single melody line (see Fig. 2). It is expected that the current monophonic realization will soon be removed, and that softcopy score will also be available.

The KCM system conceivably could be used by musicologists and music theorists to analyze musical form and style. In this application area, the piano keyboard is used to input the musical examples, thus eliminating the task of converting musical code to some appropriate alphanumeric representation for input to the computer via the typewriter keyboard, as is required by many existing analysis programs.

Due to the exact nature of the control attainable over such parameters as frequency and amplitude, it is possible to utilize the KCM system for conducting psychoacoustic experiments. Not only can the computer be used to generate sound stimuli in a precisely timed and repeatable way, it can also be used to collect the subject’s responses, and perform some preprocessing and analysis of the experimental data. Such capabilities might prove of interest to psychologists, acousticians and designers of audio systems.

There are also many possible applications of the KCM system in the area of Computer-Assisted Musical Instruction (CAMI) [3]. Work is under way to utilize the piano keyboard as an input device for both students and teachers. A new area of CAMI, that of Keyboard Performance Skills, is envisioned wherein the information provided to the computer by the keyboard
interface is used to illustrate such student problem areas as unconscious retention of notes, rhythmic inaccuracies and unevenness of touch. Since much of performance is subject to personal interpretation, the computer will only be used to "show" variations from the score, not to "tell" that corrections are required. It will be up to the student, in conjunction with the teacher, to decide whether corrective action is in order. The KCM system is presently capable of displaying keyboard activity in a piano roll-like analog representation, known as MIDI notation. It is felt that this MIDI notation is appropriate for displaying much of the information to be detected by the keyboard performance skills program [4].

The features of the KCM system can also be used for other teaching and learning applications. Answers to simple aural training questions can be entered into the computer through the piano keyboard, rather than resorting to typing in a coded response. When used in conjunction with computer-generated musical score as a stimulus, rather than an audio stimulus, it becomes possible to use the KCM system for sight reading drills.

The emphasis of any CAMI programs developed will be on user interface. Since the intended users are musicians with little, if any, computer experience, the computer must aid the user by indicating exactly what sort of response is required at any particular time, as well as being "forgiving" in case of incorrect input. It is intended that the computer can be used as a tool to simplify the task of acquiring musical skills; it is not intended that the computer replace the music teacher.

3. A MUSIC EDITOR

Based on the preceding discussion of application areas, the value of a music editor, analogous in many ways to the text editors used in word processing [5], is apparent. Composers and arrangers using the Keyboard Computer Music system need to be able to edit their input in the process of producing finished harmonies score or audio tapes. Keyboard musicians will want to be able to produce records of performances - both their own, and those of other musicians - for purposes of self-analysis and improvement.
music educators will want to be able to correct any errors they may have made when entering music examples at the piano keyboard.

The music editor must be capable of creating and deleting, by name, files of Keyboard Activity Code, as mentioned in the Introduction. Each file should have associated with it a header block, which contains such information as the name of the file, the key signature of the example, the time signature, the tempo and text information such as composition title, composer's name, and date of creation. Obviously, header information will be best input via the typewriter keyboard.

Once the file has been created, or once a previously created file has been retrieved, there must exist some means of adding or modifying data in the file. There are three input modes envisioned for these purposes:
- using the piano keyboard to input keyboard activity code
- using the typewriter keyboard to simulate the input of keyboard activity code
- using the typewriter keyboard to input such information as notes and rest values (as well as the header information mentioned above).

The ability to choose any one of these three modes as desired must exist, as must the ability to have the input either appended to the end of the file, or inserted at any desired point in the file.

In order to properly make file insertions at will, there must exist some means of unambiguously locating each individual entry. This implies the existence of some sort of editing pointer which can be used to indicate a particular place in the current file. It would appear desirable to be able to control the position of this pointer both absolutely and relatively. The former control (i.e., absolute positioning) involves being able to indicate a specific bar, or pseudobar number, or even a specific note, as well as being able to position the pointer at either the beginning or end of the file. Relative positioning moves the editing pointer in terms of a displacement from its current position (e.g., either +n or -n bars, notes, etc.).
In conjunction with the ability to position the editing pointer, it would be desirable to have "find" capabilities, whereby a musical pattern could be specified, the file searched, and the pointer positioned either before or after the desired occurrence of that pattern, if found. A wide variety of operations for the find command can be considered. The pattern could be a melodic motif (i.e., a series of notes played sequentially), with the matching performed in terms of absolute pitches and rhythm (within limits), relative pitch and/or rhythm (with the other parameter matched absolutely), or by matching the pitches either absolutely or relatively, with no regard for rhythm. Alternatively, the pattern specified could be a harmonic motif (i.e., a group of notes played simultaneously), with matching performed either in terms of absolute pitches, or relative pitches. It is even conceivable that matching could be desired simply to a rhythmic motif, either in terms of absolute durations (within some predefined tolerances), or relative durations. Table 1 presents some possible "find" modes for the music editor, while Fig. 3 illustrates some of these modes in order to clarify the discussion above.

Once a list of desired Keyboard Activity Codes has been input and modified to its desired form, the music editor must be able to provide useful output. Numerous types of output are desirable, including audio output via the digital oscillator, hardcopy musical score via either the Dotwriter printer or the Tektronix plotter, softcopy musical score via the video screen, modified files written out to the disk drives (possibly with simultaneous creation of a backup file consisting of the file as it existed prior to editing), plus any listings and diagnostics considered necessary. The editor should allow either the complete file to be output, or any specified portions thereof.

In conjunction with the output capabilities, some means of performing certain common operations on the Keyboard Activity Code should be allowed. Without actually changing the KAC, it would be useful to transpose the music up or down a specified number of semitones and change the tempo while selecting different timbres. It may also be desirable to be able to change the key and time signatures.
In addition to the capabilities outlined above, a music editor must be capable of performing global changes in order to easily enter certain types of material. Rather than playing entire chords or arpeggios, a single note could be played each time, and then the desired pattern substituted for each occurrence. A global transposition capability would allow the user to play in a preferred key, yet still produce records for the actual key desired.

A necessary music editing feature would be to combine two or more files into one file. This file combination could take two forms - concatenation, or merging. In concatenation, files are added to the end of the preceding file. In this way, repetitious passages need only be played once, and then sufficient copies of the passage concatenated. Moreover, various compositions could be created by combining suitable "sketch" files. Under merging, each file would contain, for example, a single voice. The file resulting from the merge operation would be a multi-voice musical composition. Thus, arrangements could have one file for each instrument for the purpose of printing scores, and could merge all these files to hear the complete arrangement.

The preceding discussion of editor capabilities is not intended to be exhaustive, but should help to indicate the type of functions required, and give some idea of the complexity involved in writing such an editor. Unlike text, which is one-dimensional, music is multi-dimensional. It is expected that the development of an effective music editor will be challenging not only in terms of time consumption but also in a creative sense. It would appear, however, that the effective completion of this work would yield immense benefits to a number of interesting application areas.

4. CONCLUSIONS

The Keyboard Computer Music system described in this paper has already proven to have considerable promise. In order to realize even more versatility, certain additional capabilities should be added. The ability to produce common music notation scores appears to be important, as does the ability
to control the timbre of the sound in rich, musically interesting ways. These capabilities alone, however, are not sufficient. In order to be truly useful, an effective music editor must be implemented.

When designing a music editor, it is important to remember that the majority of the users will be musicians with virtually no computer experience. Consequently, input should be made on devices which are familiar to the user, and to which the user can relate musically. The ECA system has, as major advantages, the fact that it is low-cost, and portable. In order to be truly serviceable, however, it is essential that the operating and applications software be comfortable for the intended users.

REFERENCES


**FIGURE 2. SAMPLE SCORE OUTPUT FROM KEYBOARD COMPUTER MUSIC SYSTEM.**

<table>
<thead>
<tr>
<th>PITCH MATCHING MODE</th>
<th>RHYTHM MATCHING MODE</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSOLUTE MELODIC</td>
<td>ABSOLUTE RHYTHM</td>
<td>FIG. 1c, 1f</td>
</tr>
<tr>
<td>ABSOLUTE MELODIC</td>
<td>RELATIVE RHYTHM</td>
<td>FIG. 1c, 1e, 1f</td>
</tr>
<tr>
<td>RELATIVE MELODIC</td>
<td>ABSOLUTE RHYTHM</td>
<td>FIG. 1d, 1g</td>
</tr>
<tr>
<td>ABSOLUTE MELODIC</td>
<td>ANY RHYTHM ACCEPTABLE</td>
<td>FIG. 1h, 1c, 1a, 1f</td>
</tr>
<tr>
<td>RELATIVE MELODIC</td>
<td>ANY RHYTHM ACCEPTABLE</td>
<td>FIG. 1d, 1g, 1h</td>
</tr>
</tbody>
</table>

**NOTE:** This table illustrates only matching of monophonic patterns, for reasons of simplicity. The match pattern being used is illustrated in Figure 1a.

**TABLE 1: SOME POSSIBLE FIND MODES FOR THE MUSIC EDITOR.**