MICROCOMPUTERS FOR ACOUSTIC MEASUREMENT AND VIOLIN ASSESSMENT

Colin Gough, Department of Physics,
University of Birmingham,
Birmingham, UK.

Introduction

Microcomputer-based systems are nowadays so powerful and versatile that it is rare not to include a micro at the heart of any experiment, controlling and facilitating the taking and analysis of data, giving a visual display of such data on a monitor, plotter or printer, and providing long-term storage of information on tape or disc. In particular acoustic measurement have been revolutionised by the use of fast A/D (analog to digital) converters interfaced to micro-computers, where the impressive computing power of even the simplest micro can be utilised to perform remarkably sophisticated feats of spectral analysis that would take months to perform by hand. I will attempt to show how, for the cost of the equivalent of a few tens of dollars, almost any 8-bit micro-system can be turned into an extremely powerful tool of acoustic analysis, which a few years ago could only have been found in the best-equipped research laboratories.

As illustrations of the kinds of measurements that can be made, we will include examples of particular relevance to the acoustics of the violin. Such measurements may be of some interest to scientifically-minded violin makers, since results can be obtained almost instantaneously, thus allowing the possible interactive use of measurements as an aid, for example, to choosing the most appropriate tone-woods or for guiding the carving during plate-tuning. This is, of course, no different in principle from the traditional use of one’s ears to tell when to tap-tones as a method of acoustic quality control during violin-making. However, the use of a micro for such testing has the advantage of enabling a quantitative and permanent record to be made, which, if for no other reason, might be helpful in training young makers to distinguish the most prominent modes excited.

The interfacing of microcomputers to experiments is an essential modern technique in electrical measurement practice and is taught as such to all our physics and engineering university students. Indeed, a surprisingly large number of 13 year-olds in the UK are already familiar with the use of such techniques in the school classroom. Micro-computer-based measurements are revolutionising the way that acoustics can be taught, to such an extent that the study of musical acoustics has been completely revitalised, particularly through lecture demonstrations and laboratory experience. An example of the latter is illustrated in Fig. 1, showing two undergraduates developing a simplified version of Ken Marshall’s modal analysis measurements on the violin (1) using techniques very similar to those to be described here.

Figure 1. A student modal analysis project.

Interfacing to microcomputers

Our experiments have all been developed using the BBC-ACORN, a fast and versatile 6502-based, 64K micro with outstanding facilities for graphics and interfacing to experiments. However, the techniques to be described could be adapted relatively easily to any 8-bit micro. For example, one could use an old Apple, Commodore, Tandy, RadioShack, Sinclair/Timex, etc., micro that might well be lying unused and forgotten at the back of someone’s cupboard having been superceded by a later-generation, larger memory micro, such as an IBM PC or its equivalent.

Although each micro will present a slightly different interfacing problem, there are, as implied by my earlier comments, plenty of young people around with the necessary expertise to provide advice and practical help should you need it! There are also several useful books that deal with interfacing 6502 and 280 microprocessor systems in some detail(2). Alternatively, one could buy an Apple, Hewlett-Packard or IBM PC and use plug-in boards specifically designed for the kinds of experiments that I shall be describing, in the past such systems have mostly been designed for the research laboratory, so that the hardware and associated software, although more powerful, tends to be rather expensive. Nevertheless, even this is changing, as an increasing amount of inexpensive software and hardware is becoming available for the home and educational market.

Fig. 2 shows the microcomputer-based configuration used for most of the experiments that we shall be describing. It includes a fast A/D converter to convert acoustic waveforms to a digital signal and a dot-matrix printer to provide a graphical record. Both the A/D converter and the printer are controlled by a single VIA chip (a versatile interface adapter), which provides a programmable interface between the microcomputer and the attached devices. Such a VIA chip is already incorporated in many micros, including the BBC-ACORN and Commodore computers, and is easily accessible to its data and selected address lines (e.g., Apple computers).

(See next page for Figure 2.)
Information is transferred between the micro, VIA and peripheral devices in the form of 8-bit numbers (0-255) on 8 data lines (one for each digit) using voltages of 0V and 5V to represent the digits 0 to 1 respectively. The way that the VIA transfers information between the micro and peripheral devices is determined by the contents of 16 reserved memory locations (referred to as control registers) shared between the micro and the VIA (this is called memory mapping). Entering appropriate 8-bit numbers into these memory locations controls the operation of the VIA by effectively setting 16 banks of switches each containing 8 independent electronic switches set by the digits 0 or 1 stored in them. In our applications we store numbers in the appropriate control registers to configure the VIA so that the 8 data-lines or port A become output lines to the printer and the 8 data lines on port B are inputs from the A/D converter. The VIA also generates accurate timing sequence to control the rate of data taking.

Both sets of data-lines have 2 associated "handshaking" lines, which allows the computer to send and receive messages from attached peripheral devices. The particular fast A/D converter that we use (see the circuit diagram in Fig. 3) is controlled by a single handshaking line, CB2. This leaves the other handshaking line, CB1, free for use as an external trigger input to initiate measurements, if required.

The A/D converter

The A/D converter converts voltages in the range —3V to 3V to an 8-bit number from 0 to 255 using a linear scale. The conversion takes place in about 10 microseconds and is initiated by the detection of a rising-edge on the CB2 line. When the micro next forces CB2 low the result of the last conversion is transferred via the data-lines into one of the control registers of the VIA specifically reserved for this purpose. Since this register is simply part of the micro's memory, this data is immediately available for display or subsequent analysis.

To illustrate how easy it is to control and record data using such a system, we list below a simple BASIC program for the BBC micro, which prints out 256 consecutive readings of the voltage at the input to the A/D converter.

```
10 ?&HE62 = 0 : REM sets the VIA register at &HE62 to 0000 0000 so that Port B acts as an input (i.e. a POKE statement)
20 FOR I = 0 TO 255
30 ?&HE6C = &DO : REM sets VIA register at &HE6C to 1101 0001 causing CB2 to go low ready for data transfer
40 P. &HE60 : REM print the number stored in VIA register &HE60 storing the last A/D conversion (i.e. a PEEK) 
50 ?&HE6C = &FO : REM sets VIA register at &HE6C to 1111 0000 switching CB2 high thus initiating a new data conversion
60 NEXT : REM repeat the sequence
```

In this example, the rate at which data is recorded is determined by the time taken to go round each FOR..NEXT loop in the BASIC program. In practice this is far too slow for acoustic measurements and it is necessary to use a machine-code version of the above program to control and record measurements. It is relatively straightforward to write such programs that also include accurate timing sequences generated by the VIA itself to control the rate of data taking. Using such a program, we can convert data at 15 microsecond intervals, which is sufficiently fast for any measurement in the audio-frequency range. A full listing of the BASIC and machine code programs that we use to operate the above A/D converter, together with the software to display such data and to evaluate Fourrier Transforms, is available on request. These programs could be adapted relatively easily for use on other microcomputers.

Examples of recordings

As our first example of the kind of measurements that can be made with such a system, Fig. 4 shows waveforms for the bowed notes of a whole tone scale played on the G-string of a Vuillaume violin, recorded in the open-air at a distance of 50 cm from the violin in the "audience-direction". Such a demonstration immediately illustrates the surprisingly large changes in the sound produced as the bow is moved up and down the string.
in waveform that occur between adjacent notes on a bowed violin, even though such changes appear to have remarkably little effect on the "perceived" tone. We will return to this point later, when we illustrate how the micro can extract spectral information from such waveforms.

Figure 5 is a recording of the "tap-tone" of the same violin, recorded under much the same conditions, the violin being tapped by a rubber-tipped hammer on the G-side of the bridge. In these measurements a piezo-electric crystal (extracted from a discarded gramophone pick-up) was mounted on the hammer, to generate a voltage at the moment of impact. This was subsequently amplified and used as the trigger input to CB1, thus initiating the collection of data. Alternatively, the output from the A/D converter can itself be used to trigger measurements, in just the same way that the self-trigger control on an oscilloscope is used.

The Fast Fourier Transform

Tapping the violin excites a large number of vibrational modes, so that the "tap-tone" is a complicated superposition of many decaying modes. The frequencies of such modes can be extracted from such a signal using the Fast Fourier Transform (FFT) technique, which is relatively easy to implement on a microcomputer. If, in this method, 512 measurements are recorded at regular intervals over a total time T, the FFT determines the amplitude and phase of 256 component frequencies, each being an integral multiple of $1/T$. This is a direct consequence of the FFT algorithm which assumes that the recorded waveform repeats itself indefinitely. By the Fourier Theorem, all component frequencies must therefore be multiples of the repeat frequency $1/T$. The FFT essentially solves 512 simultaneous equations for 512 unknowns — the amplitude and phase of the 256 component frequencies. Thus 512 points recorded over a period of 0.1s produces a spectrum at intervals of 10 Hz (110.1Hz) from 0-256kHz. In practice a complication known as aliasing arises from any higher frequency components present in the original waveform, which have to be removed by analogue or digital filtering before the FFT analysis is performed.

A simple FFT program written entirely in BBC basic is included as an appendix, so that anyone with access to a micro can develop simple programs of his or her own to perform spectral analysis of waveforms. In BBC-BASIC it takes about 30 seconds to transform 256 data points. However, the use of a machine code version of the program speeds things up considerably. On our BBC micro, the 8502 machine code program converts 512 points in just over 3 seconds.

In Fig. 5, below the tap-tone waveform, we show the derived FFT spectrum. A strong air resonance is evident just below 300 Hz and several prominent body resonances in the range 500-1500 Hz. Here, the Fourier components have been plotted on a linear amplitude scale, though they could equally well have been plotted as a power spectrum using linear or dB scales. Since every violin gives a unique spectrum, such measurements provide a record that might be useful to the violin maker as a way, for example, of monitoring how the use of different quality woods, different archings and tunings of plates affect the physical characteristics of the completed violin. Unfortunately, any meaningful correlation of physical properties with the quality of tone judged by the performer still appears to be tantalisingly illusive.

Fig. 6 shows FFT spectra for a set of bowed-note waveforms similar to those shown in Fig. 4. The spectral components are again plotted on a linear scale. Note the very weak fundamental component for the open-G string, reflecting the absence of any acoustically significant resonances.

Figure 4. Bowed notes waveforms

Figure 5. Vuillaume Tap Tone
in the neighbourhood of the fundamental. The fundamental component remains relatively weak until the bowed note approaches the frequency of the air-resonance at around C#. It is a salutary experience to observe the large changes in amplitude of overtones from note to note on a good-quality instrument that would be judged aurally to have a uniform tone. Such observations make it difficult to understand why the particular frequency at which a particular resonance (such as the main body resonance) occurs can have anything other than a rather minor influence on the overall tone of an instrument. At best it will only affect one of the overtones on a particular note, which we have already seen can change dramatically without significantly altering the perceived tone of the instrument. However, other physical properties of the vibrational modes, such as their radiativity and effective masses may well be important because they contribute to the “global characteristics” (see Art Benade’s introductory paper given at CAS Conf. 1986/6) well above the resonant frequency, in just the same way that an idealised loudspeaker with a single low-frequency resonance results in a uniform sound output in a wide frequency range above resonance.

In Fig. 7 we show a sequence of FFTs derived from “tap-tones” recorded on a pair of tuned but unvarnished tuned plates intended for Carleen Hutchins’ violin SUS 266, which have been very expertly tuned according to the prescription described in Ref. 3. The plates were freely supported on rubber bands and were tapped in the three positions illustrated to excite various vibrational modes. The sound was recorded by a centrally placed microphone placed about 1/4 of the plate length from the top. The vertical dotted lines on the expanded representation have been drawn as an aid to the identification of the various modes excited (on the top plate modes 3 and 4 cannot be separately identified). Note the very near coincidence of modes #2 and #5 tuned nearly an octave apart in both the front and back plates. As expected, varying the tapping position excited the various modes preferentially so that, with patience, modal lines for particular modes can be mapped out. Although the glitter-pattern vibrational method remains a much more direct way of deriving such information, the use of the above technique to obtain almost instantaneous quantitative records of prominent “tap-tone” frequencies may be of some value to those makers who continue to use frequencies alone in plate tuning. Such measurements would certainly be useful in training apprentice violin makers to identify tap-tones while thicknessing plates.

The above methods might also be valuable in selecting tone woods, which are again often assessed by the “ring” of the sound produced when a block is struck. The FFT technique could be used to determine the frequencies and to estimate the damping of the vibrational modes of standard-shaped blocks of wood, so that a quantitative record of wood-quality and resulting tonal quality of a violin could be kept.

Figure 6. Wave forms and FFT spectra for bowed notes on Vuillaume g-string
Acoustic Radiation Spectroscopy

As a final illustration of the power of the simple micro for both measurement and analysis, Fig. 8 shows spectra for four violins obtained in a student project using a simplified version of Gabi Weinreich's powerful radiation spectroscopy technique (4).

In these measurements, vibrations were recorded using a rigidly supported gramophone pick-up with the stylus resting lightly on the bridge of a free-suspended violin irradiated with sound from an electrostatic loudspeaker. The frequency of the sound was linearly ramped across the frequency range of interest and the amplitudes of induced vibrations in phase and in phase-quadrature with the signal to the loudspeaker were recorded using a phase-sensitive detector. These signals were monitored by a slow 12-bit A/D converter incorporated as a standard facility in the BBC micro. The rates of change of these two components were then calculated by the micro, and after squaring, adding and square-rooting were displayed as the outputs shown in Fig. 8.

As pointed out by Weinreich, this method of analysis enables individual modes of vibration to be distinguished far more readily than if the amplitudes or intensities alone had been recorded, thus throwing away phase information (6). A similar analysis can be applied to FFT data to help resolve resonances. Although the resolution and sensitivity of the above measurements falls short of Weinreich's beautiful data, I hope that such results at least demonstrate how, with a micro, a few dollars and a little ingenuity, it is possible to perform remarkably sophisticated experiments in the undergraduate laboratory, home or workshop, which a few years ago could only have been envisaged in the best-equipped research laboratories.

Final remarks

In this paper we have only scratched the surface of the use of the microcomputer in acoustic measurement. Nevertheless, we hope that some of the examples described will persuade anyone interested in the acoustics of the violin of the potential value of such measurements. Although insufficient details have been given to fully implement a microbased recording system for any particular micro-computer, we hope that the guidelines given have demonstrated that the problems to be tackled are fairly straightforward. There are almost certainly other possible applications for microcomputer-based measurements in the violin-maker's workshop but only practical experience by skilled luthiers can demonstrate their ultimate value as an aid to scientific violin-making. I hope that this paper will at least encourage some makes to experiment with such techniques.

Acknowledgements

I am grateful to several undergraduate students who have helped in the development of microcomputer-based acoustics measurements and to Paul Withey, Michael Hume, Steve Wotton and Bernard Phipps in particular. Dr. John Woodward was also most helpful in providing me with a machine-code program for 512 points. I would also like to thank Carleen Hutchins for her invitation to a small workshop session prior to this conference, which enabled me to make a large number of measurements on plates and completed instruments including those shown in Fig. 7.

**Figure 7.** FFTs of "tap tones" - free plates #266 SUS

**Figure 8.** Acoustic resonance spectra
References:
2. For example, the 6502 and Z80 Applications Books by Rodnyak Zaks (Sybex).

APPENDIX

The following BASIC program evaluates the FFT of the input data array entered at line 580. The program uses procedures that could be replaced by GOTO statements in less powerful versions of BASIC. The data is first shuffled using the bit-reversal routine, in which each member of the array is changed with its bit-reversed companion (e.g., 1000100015 = 0100000115 bit-reversed). Even in BASIC the program can be speeded up very considerably by using integer variables wherever possible, and by storing the sine, cosine and bit reversal tables in memory, to avoid having to evaluate the data each time.

```
10 REM FFT 256 points
20 MODE 4
30 PROC Arrays
40 PROC Input
50 PROC Graph
60 PROC Intable
70 PROC Bitreversal
80 PROC FFT
90 PROC Output
100 END
110 DEF PROC Graph
120 MOVE 0, 512
130 FOR T = 0 TO 255
140 D = INT(256 * T)
150 FOR Z = 0 TO T - 1
160 L = INT(D * Z)
170 FOR I = 0 TO D - 1
180 A = 2 * I / T + Z / B = A + T : F1 = Re(A) : F2 = Im(A)
190 P1 = CO(L) : RE(B) : P2 = SIN(L) : IM(B)
200 P3 = SIN(L) : RE(B) : P4 = CO(L) : IM(B)
210 RE(A) = F1 + P1 - P2
220 IM(A) = F2 + P3 + P4
230 RE(B) = F1 - P1 + P2
240 IM(B) = F2 - P3 - P4
250 NEXT: NEXT: NEXT
260 ENDPROC
270 DEF PROC Intable
280 FOR T = 0 TO 255
290 CO(T) = COS(T * K) : SIN(T) = SIN(T * K)
300 NEXT: ENDPROC
310 DEF PROC Arrays
320 FOR T = 0 TO 255
330 NEXT: ENDPROC
340 ENDPROC
350 ENDPROC
360 ENDPROC
370 ENDPROC
```

JCAS #48, November 1987