INTRODUCTION

This paper describes the realisation of a speech output peripheral, based on the Parametric Artificial Talker (PAT), which is simple to control, and requires a minimum data rate between the computer and the device, being thus suited to incorporation into a multi-access, time-sharing, memory-sharing computer system. The work is part of a larger project on man-machine interaction using speech, and is funded by the National Research Council of Canada.

A BASIS FOR SPEECH OUTPUT

Tape-recorders, audio-response drums, spectrum pattern synthesisers, and even "jukebox" style devices and tape-based electronic organs may all be considered as means of obtaining speech output from computers (HILL 1971). Most such devices, in effect, play back pre-recorded real speech. This renders them inflexible in use, and places practical limits on the size and character of their vocabularies. The electro-mechanical variety are often unreliable, and/or require skilled and frequent maintenance.

It is now practical to convert strings of phonetic sound symbols into speech by using them as a basis for calculating the parameters required to control an electrical equivalent of the human vocal apparatus. It is not necessary to model the physiology of the vocal apparatus, just its resonant behaviour. The Parametric Artificial Talker (PAT) is such a resonance model of the vocal apparatus. A computer system equipped with a PAT, suitable means of driving PAT, and programs to calculate the PAT parameters from strings of phonetic elements, may deliver speech output with the same facility and flexibility that current machines deliver text output. Applications of such spoken output would include teaching children to spell, generating tape-recorded sets of instructions for complicated wiring tasks, answering dialled-in requests over telephone lines, and allowing blind persons to work with computers.

THE PAT BASED SPEECH OUTPUT PERIPHERAL

PAT could be driven from a computer using a standard 8-channel digital-to-analog converter. The controller to be described offers three advantages over this arrangement. First it reduces the bit rate demanded from the computer, secondly it reduces the interrupt service overhead, and thirdly it gives the appearance (to the controlling computer) of a simpler, more unified device. The first two are important in a typical modern multi-access computer system, which must deal with many conflicting demands for service. They are especially important in this work, since the ultimate goal is to handle several synthesizers within one overall system and it is desired to combine the largest number of speech output channels with the minimum degradation of other ser-
vices to users. The third advantage is equally important, in that simple de-
vices may more easily be incorporated into standard operating systems without
specialist aid. The present device appears to be a standard paper-tape punch
to its TSS-8 operating system, and no modification to the standard device han-
dler was required. This will appeal to users with systems maintained under
manufacturer contracts.

The complete device, comprising PAT and the low-data-rate control unit,
is referred to as a PAT Based Speech Output Peripheral (PBSOP), and forms a
basis for speech output, even from small computers, offering economy of data-
rate, and simplicity of control.

HARDWARE DETAILS OF THE PBSOP

PAT has been described elsewhere (Antony & Lawrence 1962) and the current
linear micro-circuit version is described completely in the full technical
report on this project. The production of intelligible speech, using reso-
nance analog synthesisers, has been reported in the literature (most recent-
ly, Allen 1972). We shall confine our attention here to the main features of
the low-data-rate interpolating interface designed to allow PAT to be con-
trolled, as if it were a paper-tape punch, by the TSS-8 computer system.

The control parameter variations required to produce intelligible speech
from a resonance analog synthesiser need not contain frequencies above 100
hz. They may also be approximated by straight-line segments representing
steady states (e.g. vowel sounds) and transitions (important cues for the
perception of consonants). Although a steady valued control parameter may
be adequately represented by the output of a digital-to-analog converter,
even at very low data rates, the transitions, which relate to the informa-
tionally more important consonants, require sample intervals that are short
enough that the step size from one sampled value to the next is acceptably
small. The required accuracy in representing transitions sets a lower lim-
it on the sample rate which must be equalled, or exceeded. By incorporating
linear interpolators in the control parameter driving circuits, the minimum
sample rate required may be reduced, since only the beginning and end values
of a parameter transition need be specified. The effect is analogous to that
of incorporating vector generators into a graphical display controller, and
this analogy suggests that further reduction might be achieved by incorporat-
ing more specialised function generators in the parameter driving circuits.
The minimum sample rate is, at least for the present, still dependent on the
character of the transitions, however, relating to the maximum tolerable du-
ration of the shortest transition required. It turns out to be high enough
that most steady-state sounds, and some transitions, occupy several sample
periods.

When the value of a parameter does not change from one sample time to the
next, but is still updated, redundant data is being transmitted. A second
method of reducing the data rate, therefore, is to cut out the transmission
of this redundant data, while leaving the sample rate unchanged.

If PAT were driven from an 8-channel, general-purpose digital-to-analog
converter, there would be machine instructions to select a channel, and ma-
chine instructions to load the channel buffers. It would also be neces-
sary to provide a clock, and a machine sensible flag, driven from the clock,
to control the sample rate for PAT. Figure 1, which illustrates the final
form of the PBSOP, shows that the basic timing and data flow follow the
The details of the PBSOP differ from the straightforward approach in ways which reduce the cost, implement the data-reducing strategies, and yet give the appearance, to a controlling computer, of a simple device. At most, 6 bits are needed to specify a parameter value, and a 6 bit converter is simpler and less costly than the more usual 10 or 12 bit converter. This also means that, even on a computer with very small word length, the address bits may occupy the same word as the data destined for that address. Since even five bits are adequate, it would be feasible to use character-oriented data transmission to drive the device using 8 bit bytes. Linear interpolation is effected by sampling the difference between the current output of a parameter driver, and the value demanded at the new sample instant, and then integrating this difference so that the demanded value is just achieved at the output by the end of the sample interval. By this time, a new demanded value will have been delivered from the computer, the new difference may be sampled, and the process repeated. The flag that signals the computer for a new set of demanded values is set by the trailing edge of the pulse that samples the difference above, which guarantees that the demanded value will not change during sampling, provided that the computer can service the request during the ensuing sample interval -- say within 20 milliseconds.

Figure 1: Main features of the PAT Based Speech Output Peripheral.
With this arrangement, it is clearly impossible to stop the clock, since without sample pulses, the interpolators would cease to deliver controlled outputs to PAT. At the same time, a computer running a large multi-access system under an interrupt arrangement cannot afford to deal with continuously recurring but unnecessary interrupts. To allow disabling and enabling of the flag, without complicating the control, was a problem. It was solved by providing a "flag enable" flip-flop that is cleared by the signal that loads the CAR (and may clear the flag itself, see below), and is set by the signal that loads a CDR. Since both these operations are normally carried out in sequence, the net effect is normally zero, but the device may be put to sleep by issuing only the former signal. As described so far, the device now closely resembles a standard paper tape punch in its behaviour, except that, unlike the punch, it is usually asking for more than one "character" each time the flag rises.

It would be perfectly easy to take care of this remaining problem by software, and at the same time cut out the transmission of redundant data, using a data list which comprised CDR values, and value counts for each successive sample. The handler for such a device, however, would not then be a standard paper-tape punch handler and, being more complicated, the interrupt service overhead would be increased. The final invention, therefore, was to provide that the device flag is only cleared when the address of channel 7 is loaded. In this way the device handler continues sending "characters" until a channel 7 sample is sent, since the flag does not drop till then. The hardware ensures both that the data in the characters reach the correct channels, regardless of order, and that the flag drops when the "marker" channel (7) is sent. The device may now be handled by a standard paper-tape punch handler, with low overhead, and the redundant data is eliminated simply by not including it in the data list. The end of each block of CDR values comprising a time sample is marked by a channel 7 CDR value.

Some data-rate advantage remains to be gained by using the channel that varies most frequently as marker, rather than channel 7, since the marker channel must always be included in the data list for each sample time. For a typical word synthesised by rule ("zero" which was 980 milliseconds in duration) the number of bits actually transmitted with the present arrangement was 1359, whereas using the most frequently changing channel would have reduced this to 1143 bits. Transmitting all channels all the time would have required 3528 bits with addresses, or 2352 bits without addresses.

REFERENCES

ALLEN, J. (1972) Speech reading machines for the blind: technical problems and the methods adopted for their solution. Proc. LEEE/AFCRL Conf. on Speech Comm. & Processing, Boston


Přeučil M. (1972) Desampling Capability of Polynomial Interpolators. Kybernetika 8, No. 4