Behavior Research Methods & Instrumentation
1978, Vol. 10 (5), 712-714

Voltage-controlled oscillators: Inexpensive alternative to analog-to-digital converters

BRIAN PARTRIDGE
Department of Experimental Psychology, University of Oxford, Oxford, England

and

RICHARD DAWKINS and CHARLES AMLANER, JR.
Animal Behaviour Research Group, Department of Zoology, University of Oxford, Oxford, England

We describe a simple software mimic of an analog-to-digital (A/D) converter that is suitable for most laboratory applications requiring A/D. The routine samples output from a voltage-controlled oscillator that can be built for around $6. Results of proving trials show the output to be linear and sampling rates of up to 50/sec are possible.

A typical analog-to-digital (A/D) converter for a laboratory computer may cost several hundred dollars. We describe here an alternative that is adequate for many biological and psychological applications, but that only costs around $6 to make. The principle is simple, and the only computer interfacing required is available on most typical laboratory minicomputers, and indeed would be required as part of the interface for a conventional A/D converter. Input voltage to the circuit is used to control the frequency of an oscillator and this frequency is then analyzed by the computer. The reason that the system is so inexpensive is that this analysis is done by software rather than by hardware. The technique has already been used in the "organ" event recording system (Dawkins, 1971). The original paper included a speculative reference to its possible application for analog conversion, but it was not then known whether it was feasible in practice.

The present paper describes an oscillator circuit that is accurate enough to give reliable results and has already given good service in behavioral research (Pitcher, Partridge, & Wardle, 1976; Partridge & Pitcher, Note 1). The system has been tested on a PDP 8 (Digital Equipment Corporation). No doubt modified versions would work on comparable computers. Sample results from proving trials are given, and a version of the program in PDP 8 machine code is included in the Appendix.

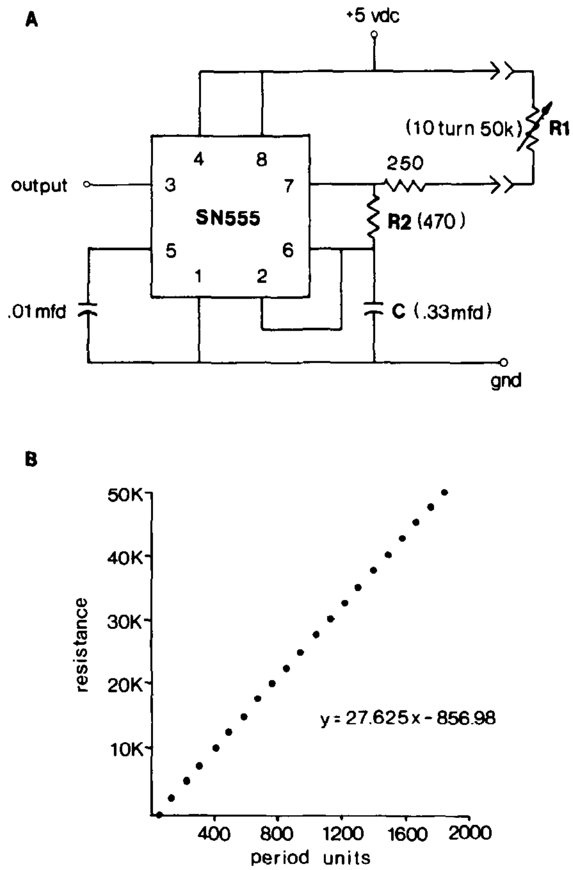

Figure 1. (a) Circuit diagram for voltage-controlled oscillator, using a Signetics 555 timer. R1, R2, and C can be altered (see text) to provide output in the desired range. A single 250-ohm resistor is placed in series with R1 to prevent the resistance across Pins 7 and 8 ever dropping to zero. That would result in output frequency equal to zero, and the computer would hang up, waiting for 10 of the infinitely long pulses. (b) Output from the subroutine ORG as a function of resistance (R1) and our implementation of the circuit. Output will always be linear but individual chips may produce slightly different values for the y-intercept.

CIRCUIT DESCRIPTION

The circuit, shown in Figure 1a, is a free-running square-wave oscillator that has a programmable output frequency dependent on the voltage across Pins 7 and 8. The control voltage may be varied by a potentiometer; in our application, a 10-turn linear 50K potentiometer (Radio Spares Ltd.) allowed us a scale of 2,000 arbitrary units. The circuit produces pulse periods (1/frequency) that are linearly related to resistance over the entire 50-kohm range (Figure 1b). When tested with a decade resistance box, maximum deviation from linearity was 1 unit in 2,000, over the entire range. Due to its low cost, when several channels of A/D are needed, it is far cheaper to build several copies of the circuit than to provide a multiplexor.

COMPUTER INTERFACE

All that is required is for each pulse from the oscillator to set a flip flop whose state can be examined by a program instruction of the "skip on flag" type. The flip flop should then be reset (i.e., the "flag cleared") by a software instruction. Unlike the original "organ" system on which the present one is based, the present program does not use any program-interrupt facility. Therefore, the flip flop that is set by the oscillator does not need to be connected to the interrupt-request line. Our system has been proved on a PDP 8A with a DKC 8-AA option board (Digital Equipment Corporation). On this option board, the output from the oscillator may be connected directly to the data-ready line, or alternatively, to the data-accepted line. We have two oscillators, one connected to each of these two lines, making two independent channels of A/D conversion.

PROGRAM

The program is written in the PDP 8 machine language, PAL (see the Appendix), as a subroutine called ORG. Each call of the subroutine causes the next 10 pulses from the oscillator to be timed. On exit from the routine, the median interpulse interval is placed in the accumulator. We use the median rather than the mean because it is less subject to error through pulses "dropping out." The program given here will run on any of the family of PDP 8 computers, but we have included the following explanatory notes to assist in translating it for other computers, such as the PDP 11 or NOVA.

On entering ORG, the computer spends most of its time in a software timing loop. On each cycle of the loop, the routine increments a time counter and checks the state of a flag. When the flag has been raised by a pulse from the oscillator, the program exits from the timing loop. The time counter register (COUNT) then contains the interval, in arbitrary units, since the previous pulse. Multiplying this by a constant gives the analog input voltage (see Figure 1b). Since the cycle time of the computer is constant, a hardware clock is not needed. A sequence of 10 intervals is stored in a buffer starting at location DAT. When 10 intervals have been accumulated, the subroutine MEDIAN is called to compute the median interval.

The subroutine ORG may be included in a full machine code program. Alternatively, it may be embedded in the operating system of a high-level language such as ALGOL. On our system it is used as an input channel specification patch for the version of ALGOL supplied for PDP 8 and PDP 11 by RHA Minisystems Ltd. (83 Gidley Way, Horspath, Oxford, England). As far as the ALGOL user is concerned, the oscillator system is treated just like any other device delivering an item of data, such as the Teletype or paper-tape reader. An ALGOL function call simply yields the current value of the interpulse interval.

SAMPLING RATE AND PRECISION

The major disadvantage of the circuit described here is that sampling rate is a function of the period of the voltage-controlled oscillator (VCO) output. This means that at the lowest frequencies, the software is considerably slower than commercial A/D converters. By altering the values of R2 and C (in Figure 1a) according to Equations 1 and 2 below, maximum and minimum output frequencies may be shifted to raise the minimum sampling rate, but at a cost to resolution. This is due to the relationship between pulse period and sampling rate. For example, at lower frequencies the maximum sampling rate will be lower than at higher frequencies.

To calculate operating frequency, it is first necessary to work out the on (T1) and off (T2) pulse times:

$$T1 = .685\,(R1 + R2)\,C, \tag{1}$$

$$T2 = .685\,(R2)\,C, \tag{2}$$

where R1 and R2 are in ohms, and C is in farads. The minimum on time will occur when R1 is minimum, so that the output will be of maximum frequency. Likewise, maximizing R1 results in minimum operating frequency. The total pulse period in seconds is:

$$T = T1 + T2 \tag{3}$$

and the operating frequency is:

$$F = 1/T. \tag{4}$$

For a scale of 2,000 arbitrary points, the maximum sampling rate over the entire range is 10/sec, but this may be increased to 50/sec with a corresponding scale of 400 arbitrary divisions. For most applications, users will be able to find a satisfactory compromise between sampling rate and precision. For instance, the interactive plotter described below rarely samples at rates of over 4/sec. Since the system uses two VCO circuits, we have a scale of 2,000 by 2,000 divisions for a 25 x 35 cm plotting surface and a resultant precision of greater than .1 mm.

The resolution of the system is, of course, a function of both its precision (in our case, a scale of 2,000 units) and its repeatability. As stated above, the maximum deviation from exact linearity was 1 unit in 2,000 over many repeated samplings of the entire range. This error is not one of repeatability but simply a rounding error at the maximum resolution of the system. For instance, a period of 1,586.5 units would be recorded half the time as 1,586 and half the time as 1,587 units. When the analog input does not fall exactly between two timing loops, however, repeatability is extremely good, and repeated conversions of the same input resistance will produce the same output.

APPLICATIONS

The technique of using software conversion of VCO-generated frequencies could find wide usage, since most behavioral and physiological apparatus can be modified to produce resistance rather than voltage as output. Our immediate use for this device arose from the need to transfer an interactive coordinate plotter (Partridge & Cullen, 1977) to a computer that did not have conventional A/D conversion. The plotter operates by sampling the positions of two linear potentiometers that turn as the cursor is moved. The sampling of the two channels appears simultaneous to the user, but it is really, of course, successive. By measuring changes in the VCO-generated frequency caused by position shifts of the cursor, we have completely eliminated the need for an A/D converter. The system is now giving good results in this position-plotting application.

REFERENCE NOTE

1. Partridge, B. L., & Pitcher, T. J. *Sensory physiology of fish schooling,* in preparation.

REFERENCES

DAWKINS, R. A cheap method of recording behavioural events, for direct computer-access. *Behaviour,* 1971, **40,** 162-173.

PARTRIDGE, B. L., & CULLEN, J. M. A low-cost interactive coordinate plotter. *Behavior Research Methods & Instrumentation,* 1977, **9,** 473-479.

PITCHER, T. J., PARTRIDGE, B. L., & WARDLE, C. S. A blind fish can school. *Science,* 1976, **194**(4268), 963-965.

Appendix

```
/PSEUDO A TO D CONVERTER, USING DAWKINS ORGAN PRINCIPLE
/EACH CALL OF SUBROUTINE ORG SAMPLES OSCILLATOR INPUT
/ORG RETURNS WITH MEDIAN INTERPULSE INTERVAL
ORG,    0         /ENTER ORG WITH ZERO IN AC
        TAD DATST;DCA POINT     /SET UP POINTER TO START OF BUFFER
BACK,   DCA COUNT               /ZERO COUNTER
LOOP,   ISZ COUNT               /INCREMENT TIMING LOOP COUNTER
        6571                    /SKIP ON OSCILLATOR FLAG
        JMP LOOP                /NO INTRPT. BACK TO TIMING LOOP
        6573                    /INTRPT. CLEAR FLAG
        TAD COUNT;DCA I POINT   /PACK LATEST TIME IN BUFFER
        ISZ POINT               /INCREMENT BUFFER POINTER FOR NEXT TIME
        TAD POINT;TAD MEND      /GOT BIG ENOUGH SAMPLE YET?
        SPA SNA CLA;JMP BACK    /NO. RETURN FOR ANOTHER
        JMS MEDIAN              /YES. COMPUTE MEDIAN OF BUFFERED TIMES
        JMP I ORG               /EXIT WITH MEDIAN TIME IN AC
MEDIAN, 0                       /SUBROUTINE TO COMPUTE MEDIAN
        CLA;TAD DUMMY;DCA MAX;DCA I MAX
        TAD MAXST;DCA PO1;TAD M5;DCA MCO1
LOOP1,  TAD DATST;DCA PO2;TAD MCOUNT;DCA MCO2
LOOP2,  TAD PO2;DCA TRY;TAD I TRY
        CIA;TAD I MAX;SMA SZA CLA;
        JMP ENDLOOP;TAD TRY;DCA MAX
ENDLOOP,ISZ PO2;ISZ MCO2;JMP LOOP2
        TAD I MAX;DCA I PO1;ISZ PO1
        TAD I MAX;DCA CAND;DCA I MAX
        ISZ MCO1;JMP LOOP1;TAD CAND
        JMP I MEDIAN
MCOUNT, -12;    MCO1,   0;      MCO2,   0
POINT,  0;      COUNT,  0;      CAND,   0
PO1,    0;      PO2,    0;      TRY,    0
MAX,    0;      DUMMY,  MAX;    DATST,  DAT
DAT,    0
END=DAT+11
*END+1
MEND,   -END;   MAXST,  MAXI;   MAXI,   0
*MAXI+11
M5,     -5
$
```

(Received for publication April 13, 1978;
accepted May 11, 1978.)