

COMPUTER TECHNOLOGY

A simple method to record arm positioning movements in man with an Apple computer

ANNIE PAUZIE and BERNARD THON
Université Paul Sabatier, Toulouse, France

This paper describes a simple method to record arm positioning movements in humans, via the game paddle entry of an Apple computer. A machine language subroutine reads the successive positions of a rotating potentiometer, with an angular accuracy of .5 deg and a time interval between two measures of 2.5 msec, and stores them sequentially in RAM memory. These measures can be retrieved for further computations of movement parameters.

We describe a simple method for measuring arm positioning movements, using an inexpensive device connected to an Apple II with pre-existing game paddle entry. Successive positions of the arm during a subject's movements are recorded by an Apple II's machine language subroutine. A multidimensional analysis (reaction time, movement time, accuracy, speed, acceleration, etc.) is simplified by the computer's recording movement trajectory, storing data, and conducting further computations.

GENERAL PRINCIPLE

The general method involves recording of the successive positions of a potentiometer rotated by the subject's movements (see Figure 1). An Apple II is normally provided with two game paddles, which are operated via two 150 k Ω rotating linear potentiometers. The resistance can be read from an Applesoft BASIC program with the instruction $X = PDL(n)$, where n is the number of the game paddle ($n = 0$ for the left paddle and $n = 1$ for the right paddle). This instruction returns an integer number between 0 (null resistance) and 255 when the resistance value is greater or equal to 150 k Ω . With this device, one unit corresponds to about 1 deg of rotation.

For research purposes, this method has three major drawbacks: (1) measurement of the angular position of the potentiometer cannot be very accurate; (2) because the BASIC instruction (PDL) is very slow, only about 100 measurements per sec are allowed to record positions and to store them in an indexed variable; and (3) the time between two measures depends on the resistance value of the potentiometer (see below). The timing analysis of the movement is difficult.

The authors wish to thank Daniel Richard for his helpful advice and Michael Cambon for technical assistance. The authors' mailing address is: Laboratoire de Psychophysiology, U.A.—C.N.R.S. N° 664, Université Paul Sabatier, 118 Route de Narbonne, 31062 Toulouse, Cédex, France.

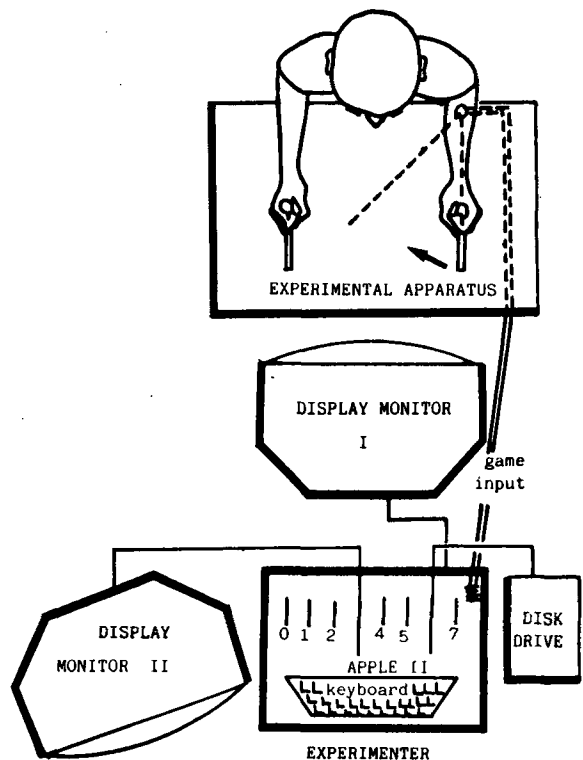


Figure 1. Schematic representation of the experimental set-up. The subject handles two levers, linked with two rotating potentiometers. The resistance of a potentiometer, which changes during arm movement, is measured by the APPLE computer via the game paddle entry. The results can be recorded on magnetic disk for further computations. The instructions to the subject are displayed on the video monitor.

To make the measurement more accurate, the firmware 150 k Ω potentiometers are replaced with 250 k Ω potentiometers, accuracy of $\pm 5\%$; with these devices, one unit corresponds to 0.5 deg of rotation. However, the maximum value that can be read (255) is now obtained with an angle of 130 instead of 260 deg. Of course the poten-

tiometers must be accurate and should be calibrated prior to use. A correcting factor may be necessary for further computation.

SOFTWARE

To improve recording speed, instead of using the Applesoft BASIC instruction PDL(n), we use a machine lan-

```

0800          1          TTL "SUBROUTINE"
0800          2          *
0800          3          *****
0800          4          * ----- *
0800          5          * RECORDING OF ARM MOVEMENT *
0800          6          * ----- *
0800          7          *****
0800          8          *
0300          9          ORG $300
001E          10         PADNUMB EPZ $1E
0008          11         TABPOINT EPZ $08
001B          12         TABLELOW EPZ $1B
0009          13         COUNT EPZ $09
FB1E          14         PREAD EQU $FB1E
001A          15         TIMING EPZ $1A
0300          16         *
0300          17         *
0300          18         *
0300          19         *
0300 A6 1E    20         BEGIN   LDX PADNUMB
0302 20 1E FB 21         JSR PREAD
0305 98       22         TYA
0306 85 1A    23         STA TIMING
0308 A4 08    24         LDY TABPOINT
030A 91 1B    25         STA (TABLELOW),Y
030C 20 1F 03 26         JSR PAUSE
030F E6 08    27         INC TABPOINT
0311 D0 ED    28         BNE BEGIN
0313 A5 09    29         LDA COUNT
0315 F0 07    30         BEQ RETURN
0317 C6 09    31         DEC COUNT
0319 E6 1C    32         INC TABLELOW+1
031B 4C 00 03 33         JMP BEGIN
031E 60       34         RETURN  RTS
031F         35         *
031F E6 1A    36         PAUSE   INC TIMING
0321 A5 1A    37         LDA TIMING -
0323 C9 C8    38         CMP #$C8
0325 D0 F8    39         BNE PAUSE
0327 60       40         RTS
0328         41         *
0328         42         END
    
```

```

0300- A6 1E    LDX $1E
0302- 20 1E FB JSR $FB1E
0305- 98      TYA
0306- 85 1A    STA $1A
0308- A4 08    LDY $08
030A- 91 1B    STA ($1B),Y
030C- 20 1F 03 JSR $031F
030F- E6 08    INC $08
0311- D0 ED    BNE $0300
0313- A5 09    LDA $09
0315- F0 07    BEQ $031E
0317- C6 09    DEC $09
0319- E6 1C    INC $1C
031B- 4C 00 03 JMP $0300
031E- 60      RTS
031F- E6 1A    INC $1A
0321- A5 1A    LDA $1A
0323- C9 C8    CMP #$C8
0325- D0 F8    BNE $031F
0327- 60      RTS
    
```

Figure 2. LISA assembler listing the APPLE mini-assembler listing of the machine language subroutine which measures the value of the resistance of the potentiometer and stores it sequentially in RAM memory.

```

10 REM *****
20 REM * ELEMENTARY BASIC PROGRAM *
30 REM * TO RECORD ARM POSITIONING *
40 REM * MOVEMENTS WITH APPLE II *
50 REM *****
60 REM *
70 N = 4: REM * 256*4 MEASURES WILL BE MADE
80 REM * CORRESPONDING TO A TOTAL
90 REM * DURATION OF 2 SECONDS
100 REM *
110 HIMEM: 38399 - 256 * N: REM * PROTECTION OF
120 REM * 256*N MEMORY LOCATIONS
130 A = 38399 - 256 * N: REM * STARTING ADDRESS
140 REM * OF THE PROTECTED TABLE
150 AH = INT (A / 256): REM * HIGH-ORDER BYTE
160 AL = A - AH * 256: REM * AND LOW-ORDER BYTE
170 REM * OF THE STARTING ADDRESS
180 S = 1: REM * PADDLE NUMBER (=RIGHT)
190 REM *CHANGE TO S=0 FOR THE LEFT PADDLE
200 REM *
210 REM * PREPARING PARAMETERS FOR THE
220 REM * MACHINE LANGUAGE SUBROUTINE
230 REM *
240 POKE 30,S: REM * =PADNUMB
250 POKE 9,N: REM * =COUNT
260 POKE 8,0: REM * =TABPOINT
270 POKE 27,AL: REM * =TABLELOW
280 POKE 28,AH: REM * =TABLELOW+1
290 REM *
300 REM * EXPERIMENT *
310 REM *
320 HOME : VTAB (10): HTAB (15)
330 PRINT "*** MOVE ***": REM = ACTION SIGNAL
340 CALL 768: REM = CALL MACHINE LANGUAGE
350 REM SUBROUTINE
360 PRINT " * END OF RECORD *"
370 REM *
380 PRINT "*** DATA RETRIEVAL ***"
390 DIM X(256 * N)
400 FOR I = 0 TO 256 * N
410 X(I) = PEEK (A + I): PRINT X(I)
420 NEXT I
430 END
    
```

Figure 3. Listing the APPLESOFT BASIC main program designed to control the experiment.

guage subroutine which returns a number indicating the potentiometer position with an interval of 2.5 msec between two successive measures (400 measures per sec) and stores the value sequentially in RAM memory. The listing of this subroutine is given in Figure 2. It begins at location \$300 (768, decimal) and must be CALLED from a BASIC program (Figure 3). Before CALLing the subroutine, the BASIC program specifies the number of the involved paddle (0 or 1) and stores it at location \$1E (decimal = 30). The number (N) of 256 measure-blocks is stored in memory location 9. The memory locations that are reserved for data storage must be protected from BASIC writing by lowering the highest memory location which can be addressed by the language with the instruction HIMEM: 38399 - 256 x N (memory 38400 is the starting address of the DOS). The beginning of the protected table is then A = 38399 - 256 x N. The high-order byte of the starting address is AH = INT(A/256), and is stored in location \$1C (decimal = 28). The low-order byte is AL = A - AH x 256, and is stored in location \$1B (decimal = 27). Finally, the main BASIC program sets the table pointer in memory location 8 to zero.

MACHINE LANGUAGE SUBROUTINE

The paddle number (PADNUMBER, in memory location \$1E) is loaded in the X-register, and the firmware monitor subroutine PREAD (at location \$FB1E) is CALLED. This subroutine returns with a number in the Y-register corresponding to the paddle position. The content of the Y-register is transferred to the accumulator and stored in memory \$1A (TIMING) to be used later by the PAUSE subroutine. The TABPOINTER in memory \$08 is loaded in the Y-register, and the measured value is stored in the Y-indexed table.

The PAUSE subroutine is then CALLED. This subroutine is necessary to provide an equal time interval between each successive pair of readings because of the APPLE computer's way of measuring the resistance of the potentiometer. The variable potentiometer is connected to a timing circuit. When a measure has to be made, a memory location (49252 to 49255, according to the paddle number) becomes greater than 128, and drops below 128 within a time proportional to the value of the actual resistance. This time is measured by a loop in the PREAD subroutine. Thus, a low resistance needs a shorter time to be measured than a high resistance; the longer duration is 3.06 msec for 150k Ω . The PAUSE subroutine equalizes the interval between two readings by means of a loop, the duration of which is inversely proportional to the measured value. The duration of this loop has been adjusted to provide a 2.5 msec interval. However, if the highest value of 255 is reached during the subject's movement, the duration of the loop must be adjusted to provide a 3.06 msec interval by changing the reference number \$C8 to \$F0.

After return from the PAUSE subroutine, a value of 1 is added to the TABPOINTER (\$08) which is then compared to 0. If the figures are not equal, the program jumps to a new reading. If TABPOINTER is equal to 0, the program indicates that a block of 256 measures has been made. The COUNTER (\$09) is then compared to 0. If the figures are equal, control is given back to the main

BASIC program. If COUNTER is not equal to 0, 1 is subtracted from the counter, then 1 is added to the high-order byte of the table (TABLELOW + 1 in \$1C) and a new series of 256 measures is initiated.

Once the recording is achieved, the stored data can be collected by the BASIC main program by PEEKing at the memory locations from A to A + 256 \times N, and storing in a file for further computation.

APPLICATION

The stored measures can be used to compute a set of parameters of the displayed movement. The computed reaction time is the number of measures equal to the value corresponding to the starting position. To have these results in msec, we multiply by 2.5. The movement extent corresponds either to a stabilization of successive measures or to the last recorded measure. The movement extent is transformed in degrees with the correcting factor (0.5 in our case). Movement time is computed between the beginning of the movement (the first measure different from the starting position) and the end of the movement (the first measure equal to the previously computed movement extent). Speed is computed by dividing the difference (in degree) between two given positions by the corresponding time interval. The movement trajectory can be visualized in high-resolution graphics on the video screen, or with an X-Y plotter.

This simple method is used in our laboratory in experiments on motor learning and specific preparation. The programs can be easily adapted to the recording of successive movements displayed on successive trials. Instructions dealing with the parameters of the movement to be produced as well as knowledge of results at the end of a given trial can be given to the subject on the video screen.

(Manuscript received March 30, 1984;
revision accepted March 26, 1985.)