ARTICLES

How choice of mouse may affect response timing in psychological studies

RICHARD R. PLANT, NICK HAMMOND, and TOM WHITEHOUSE
University of York, York, England

Mice from the early 1990s seemed to offer a cheap and viable alternative to more expensive response boxes, with fairly consistent results being found between studies. However, has anything changed in the intervening decade? Are newer mice technologies necessarily better? Is USB a better mouse interface than the old-fashioned serial interface? With such questions in mind, we outline a method for bench-testing the timing characteristics of mice outside of a PC, in order to predict their contribution to response timing. A sample set of mice was tested under a visual stimulus–response paradigm, using E-Prime to compare predicted performance with measured response registration. A representative range of mice technologies was tested alongside a standard keyboard and an E-Prime deluxe response box. The implications for using any response device other than a recognized response box are discussed.

Anecdotal evidence from the community suggests that many researchers believe that a stationary mouse, with the ball removed, is the next best alternative to a more expensive response box as a response device or to a standard keyboard. Most appreciate that using a mouse will add a "moderate" amount to their response measures and that this may be offset by virtue of being a constant error. Segalowitz and Graves (1990), for example, found that Microsoft serial mice of the day, when tested under DOS, added a delay of 31±2 msec to recorded reaction times. Crosbie (1990), using his own Turbo Pascal routines, found that a Microsoft serial mouse took around 34 msec to be detected under DOS (median = 34, $M = 35.15$, $SD = 3.54$). He found a bus mouse to be superior (median = 4, $M = 4.1$, $SD = 1.34$). Later, Beringer (1992), in a more in-depth study of mice and interface alternatives, found that a serial NCE Hypermouse added around 24.4 msec, with an SD of 0.26 msec. He found that a Microsoft bus mouse added around 11 msec, with an SD of 5.69. At the time, the PS/2 mouse was less well understood. Beringer found that an IBM PS/2 mouse added around 23.68 msec, with an SD of 4.74 msec. Beringer pointed out that if the mean and the variance of a given device are known, statistics may be corrected. It is also noted that provided the variance of the response device is small in relation to the sample variance, this does not greatly reduce the statistical power (Ulrich & Giray, 1989).

Mice from the 1990s, then, did seem to offer a cheap and viable alternative to response boxes, with fairly consistent results being found between studies. However, has anything changed in the intervening decade? Researchers are no longer likely to use DOS, removing the ability to "hit the hardware," and the bus mouse is no longer with us. Even the hitherto ubiquitous serial mouse is all but dead, being replaced by PS/2 and USB interfaces. Opto-mechanical ball mice are also on the decrease, with ball-less, camera-based optical mice now being in vogue. The move to Windows has been regarded with suspicion, with researchers assuming that response timings were likely to be worse with standard input devices. Some of the latest mice offer a dual-interface mode with which, chameleon-like, they can change from PS/2 to USB or from PS/2 to serial. Many later papers have focused on whether Windows itself is capable of millisecond accuracy, rather than on the variability of associated hardware (e.g., MacInnes & Taylor, 2001; Myors, 1998, 1999).

We believe that researchers may be interested to learn just how variable the humble mouse of today can be, both in terms of its inherent variability and in terms of differences between models and manufacturers. We also hope to dispel the myth that newer mice are somehow *better* than older ones for timing purposes, that leading brands are better than the unknown, that expensive is better than cheap, and that certain interfaces are better than others. We also make illustrative comparisons of mouse timing performance with commercial button boxes and with typical keyboards.

Our methodology is markedly different from that of previous studies in the respect that we have tested a range of mice, both of the same make and model and across different manufacturers. All currently available interface

types have been tested. We have taken a two-step approach to testing. First, we have bench-tested each mouse independently of Windows at the hardware level, using a signal generator and an oscilloscope. Second, we have retested the same mice again, using a standard Windows-based experiment generator (E-Prime) running a simple reaction time paradigm. Rather than using a fixed simulated response schedule, we used one that randomly varied responses between 1 and 5,000 msec after stimulus onset. Once generated, this response schedule was used for all testing.

METHOD

Equipment

Response devices. A total of eight mice were tested: five Microsoft branded mice (http://www.microsoft.com), one Trust (http://www.trust.com), one LogiTech (http://www.logitech.com), and one unbranded OEM/white box mouse that transpired to have been made by Sysgration Ltd. (http://www.sysgration.com). These eight were chosen so that the pool had representation from a leading brand manufacturer, as well as from one of the bulk OEM suppliers. There was also a representative sample of interfaces, reflecting those of the mouse population in general: five PS/2, four serial, and one USB. Two of the mice offered alternative interfaces: The AMI wheel mouse offered serial and PS/2, and the Microsoft optical mouse offered PS/2 and USB. An approximate technology timeline was also established by using the FCC ID (where available) to check when licenses were granted. These ranged from 1993 (OEM unbranded) to 2000 (Microsoft optical; date taken from the circuit board).

It should be noted that although some devices contained electronics designed some 9 years ago, this refers only to the date the license was granted, and not to the manufacturing date. Mouse shells are not licensed, so in theory the same electronics may appear in any shell over a long production period. Indeed, Microsoft's classic intelliMouse shape reinforces this argument, with the model number and shape not guaranteeing the contents of two apparently identical mice. For example, the two Microsoft mice shown in Figure 1 look the same externally, have the same model number, but have completely different timing characteristics. Only the FCC ID denotes that the mice are actually different.

A standard ALPS branded mechanical keyboard was used for comparison. A Psychology Software Tools deluxe response box was used to provide a baseline when each response device was tested with E-Prime.

Bench equipment. In order to test all response devices independently of Windows, two pieces of stand-alone bench equipment were used: a Hewlett-Packard (Agilent) 33120A function/arbitrary waveform generator (http://we.home.agilent.com) and a Tektronix digital phosphor oscilloscope (DPO), Model No. TDS 3034 (http://www.tektronix.com). The function generator was used to generate accurate digital pulses which, via a transistor, were used to simulate button- or key-down events of known frequency and duration. The DPO was used to record both this input pulse and the resulting data generated by the device under test. With both pieces of equipment operating at the nanosecond level, all measures were extremely accurate. The DPO has a built-in floppy disc to which screen images could be saved, which proved ideal for recording tolerances.

Standard experiment generator. A copy of E-Prime v1.0 from Psychology Software Tools was used to test a selected number of response devices. It should be noted that any Windows-based package could have been used.

Presentation PC (PRES). E-Prime ran on a standard PC that is routinely used for benchmarking software as part of the Experimental Timing Standards Laboratory (ETSL) project (http://www.psychology.ltsn.ac.uk/ETSL). Only

Figure 1. Comparison of the internals of two apparently identical mice: Model MS2.0A, serial interface (left, Mouse 2; right, Mouse 1). The right mouse shows far more chip integration, with a vast reduction in component count.

brief details of our benchmarking rig and software are given here (see Plant, Hammond, & Whitehouse, 2002, for further details). The 17-in. monitor was driven at 100 Hz, giving a 10-msec screen redraw. The PRES was a 900-mHz Athlon with 128 MB of RAM, a 20-GB hard drive, and ATI Rage Pro (AGP) with 16 MB of video RAM. Windows 98R2 provided the operating system with MS DirectX8.0a installed. No network card was present, and the task scheduler was disabled. No software other then the operating system was installed.

Control and timing PC (CAT). Our standard ETSL CAT and breakout box was used to simulate responses at predefined offsets relative to the appearance of the white block displayed by E-Prime. CAT itself was a 900-mHz Athlon with 512 MB of RAM, an 80-GB hard drive, and an AGP Leadtek Geforce2Mx with 32 MB. The only PCI slots filled were with the two National Instruments (NI) boards used for timing and control. No network or sound card was present (on-board sound was disabled). Windows 2000 Professional SP1 provided the operating system. NI DAQ drivers, Version 6.8, were used throughout to access the NI cards.

A 1 m × 1 mm diameter fiber-optic cable with suction cup and a 1-m flying lead with 2.5-mm mono jack plugs were used to monitor the screen of the PRES and to interface with a lead soldered to each response device's primary button. Our standard breakout box was used to provide monitoring and response interfaces to the CAT.

From our ETSL benchmarking suite of applications, Visual Stimulus Capture and Response (VSCAR) was used to simulate responses at known offsets relative to the leading edge of each white block that was displayed on the PRES. This has been independently validated as being able to detect a visual stimulus and simulate a response with very high accuracy (jitter, 0.58–0.84 msec).

For more details on the ETSL benchmarking rig and validation studies, see Plant et al. (2002).

Procedure

All the response devices were tested using the 33120A and TDS 3034 in order to establish their timing characteristics independently of Windows and the host PC (a PC did provide power to each device, however). A representative sample of devices was then tested using a simple script in E-Prime.

Bench-testing response devices. Our goal with each response device was to find the delay between the pressing of the primary button or key and the message's being sent to the host PC, plus any variation in this delay caused by its firmware.

The signal generator, together with a simple interface (a transistor being used as a switch), was used to simulate the primary button or key being pressed for 200 msec every second. This signal was fed to the oscilloscope as a trigger, and the data output was observed in order to calculate latencies. For RS-232 devices, the TXD line acted as a trigger; for PS/2 devices, the DATA line was used; and for USB, the D+ line was used.

Data are transmitted in packets of several bytes for RS-232 devices, PS/2 format is used for PS/2 devices, and USB human interface device (HID) format is used for USB devices. The transmission time was measured between the start of the first byte and the end of the last byte of each packet.

Observing the delay from the button- or keypress to the start of data packet transmission gives a latency figure. Since the firmware in each device is not synchronized to the signal generator, the latency will depend on the point in the firmware scanning loop when the signal (buttonpress) arrives. The shortest latency is achieved when the button or key is pressed just before the firmware checks for input changes, and the longest when the button or key is pressed just after the firmware has checked for input changes. Adjusting the button- or keypress rate to around 0.8–1.2 sec, so that the latency slowly increases or decreases, and observing this varying delay over some time (several tens of seconds) provides the maximum and the minimum values (jitter). Examples of minimum and maximum latencies are shown in Figure 2. The sum of the latency and the transmission time gives the delay from pressing the button to the point at which the PC hardware (UART for RS-232, keyboard/mouse controller for PS/2, and USB host controller for USB) has received all of the packet of data. There is a further undetermined, yet small, delay as data are processed by the PC operating system device drivers and passed to the application (e.g., E-Prime).

Testing response devices with a standard experiment generator. A simple visual stimulus–response paradigm was constructed using E-Prime. E-Prime was programmed to display a 40 × 30 × 256 pixel white block centered on a black background (800 × 600 × 24 bit). The block was programmed to terminate on a response. On termination, a black screen was displayed for 500 msec before the next trial in the sequence was displayed. A block with a total of 80 trials was constructed.

A fiber-optic cable was attached to the center of the PRES screen and then hooked up to the optodetector card on the breakout box of the CAT. The appropriate response device was then plugged into the PRES and, in turn, patched to the digital out channel of the breakout box. This connected to a 2.5-mm mono socket that had been soldered to the primary button or key of each response device. Finally, the PRES was booted into Windows, and E-Prime started with the paradigm outlined.

Our CAT was instructed to run our VSCAR software with the simulated stimulus–response schedule, with 80 stimulus–response intervals randomly generated within the range of 1–5,000 msec (see Plant et al., 2002, for more details of our benchmarking software). VSCAR remains in capture mode until it detects the appearance of a visual stimulus. It then waits the appropriate interval before simulating a response. VSCAR is triggered during the first refresh of a stimulus display as the pixels under the fiber are illuminated for the first time (the leading edge). VSCAR simulates the button- or key-
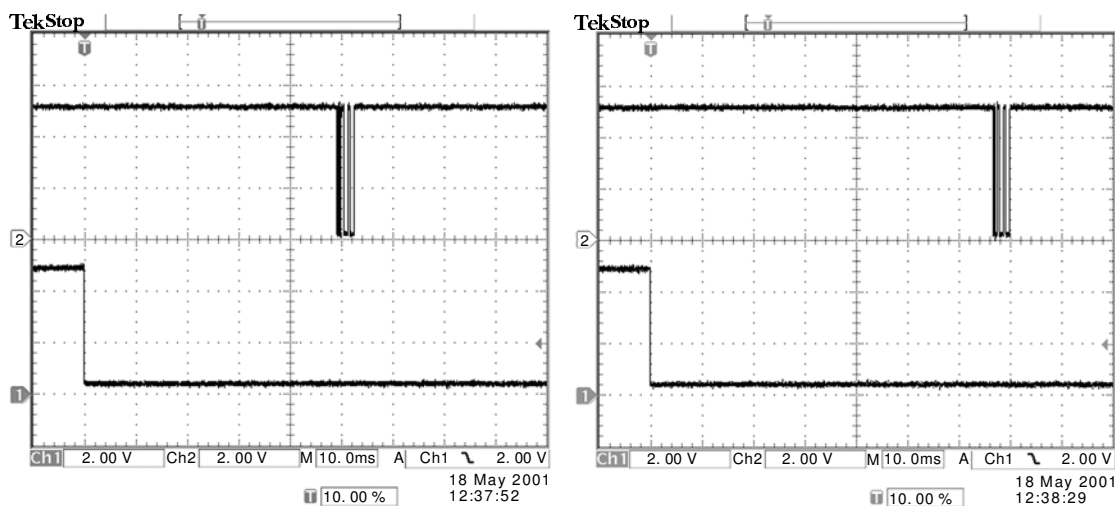
Figure 2. Minimum and maximum button-down latencies for the OEM serial mouse (Mouse 8), measured using our Tektronix Digital Phosphor Oscilloscope. The button-down signal generated by the 33120A is the lower trace; the mouse's firmware response is the upper. Each division represents 10 msec. Left image, minimum button-down registration time, 48 msec; right, maximum, 68 msec.

down event for a constant 50 msec before waiting for a further 500 msec and then rearming for the next trial.

By comparing the response time recorded by E-Prime (E-Data) with that generated according to the schedule, we are able to calculate the response target error (RTE) that is attributable to a combination of response device, Windows, and E-Prime itself.

RESULTS

Bench-Tested Response Devices

We found that there was a huge variation between the eight mice tested, using standard bench hardware, in terms of minimum and maximum button-down latencies, of button- or key-down jitter, and of transmission time. For

Table 1
Summary Results From Bench-Tested Mice, Keyboard, and Response Box

Our Ref.	FCC ID	Model	Country of Origin	Interface	Minimum Button Down	Maximum Button Down	Button Down Jitter	Transmission Time	Typical Delay
				Mice					
1	C3KSS1	MS Mouse 2.0A (ETSL)	Taiwan	serial	13.70	14.30	0.60	24.00	38.00
2	C3KSS2	MS Mouse 2.0A	China	serial	4.90	9.60	4.70	25.00	32.25
3	C3KKS2	MS Mouse 2.0A	Ireland	serial	4.80	9.60	4.80	25.00	32.20
4	C3KKMP3	MS Mouse 2.1A	China	PS/2	11.40	35.00	23.60	4.20	27.40
5a	none visible	MS optical wheel mouse	China	PS/2	10.50	39.20	28.70	5.00	29.85
5b	none visible	MS optical wheel mouse	China	USB	9.00	18.00	9.00	0.10	13.60
6a	none visible	Trust AMI dual scroll	China	serial	1.70	2.40	0.70	33.00	35.05
6b	none visible	Trust AMI dual scroll	China	PS/2	0.40	1.10	0.0007	6.00	6.55
7	JNZ201213	Logitech wheel mouse	China	PS/2	19.20	27.20	8.00	6.16	29.36
8	HQXPC93010-12	OEM unbranded	Taiwan	PS/2	48.00	68.00	20.00	3.60	61.60
				Keyboard					
KB	FMA44HKB6251	102 key ALPS mechanical keyboard (Nan Tan Computer Co.) 1988	Thailand	5 pin DIN with PS/2 adapter	22.50	31.80	9.30	1.10	28.25
				Response Box					
BB1	n/a	Psychology Software Tools deluxe response box	USA	serial	0.005	1.33	1.28	−0.5	1.17

Note—FCC ID was taken from each device when available. This was looked up on the Federal Communications Commission Web search: http://www.fcc.gov/oet/fccid/. Timings are in milliseconds.

Table 2
Response Target Error (in Milliseconds) as Recorded by E-Prime 1.0

Device	Our Reference	Interface	Mean	SD	Minimum	Maximum	Range
MS Mouse 2.0A (ETSL)	1	serial	42.81	0.58	42.00	44.00	2.00
AMI wheel mouse	6a	serial	39.20	0.46	38.00	40.00	2.00
AMI wheel mouse	6b	PS/2	10.15	0.53	9.00	11.00	2.00
MS optical wheel mouse	5a	PS/2	35.13	8.46	20.00	50.00	30.00
OEM mouse	8	PS/2	66.30	6.48	56.00	80.00	24.00
MS optical wheel mouse	5b	USB	20.09	2.25	15.00	25.00	10.00
PST deluxe response box	BB1	serial	6.44	0.99	4.00	9.00	5.00
ALPS keyboard	KB	PS/2	34.35	3.39	28.00	40.00	12.00

example, between the best and the worst performing mice, there was an absolute maximum difference of 66.9 msec and a minimum difference of 47.6 msec. The best performing mouse (Mouse 6b) had a minimum button-down latency of a mere 0.40 msec, as compared with the worst mouse (Mouse 8), with a latency of 48 msec. Maximum button-down latencies for these two mice were 1.1 and 68 msec, respectively. A summary of the results for all eight mice, as well as equivalent data for the keyboard and the response box, is shown in Table 1.

There was also a smaller, but still significant, variation in button-down jitter—that is, the amount timings varied in terms of minimum and maximum latencies (range). Mouse 5a had the largest internal variation of some 28.7 msec, whereas the best (Mouse 6b) varied only by an infinitesimal 0.0007 msec.

In terms of transmission time, again there was a wide variation depending on the response device and the interface it used. It should be noted that transmission times were based on the hardware and the interface used. In terms of transmission time, the best mouse (Mouse 5b USB) had a transmission time of a mere 0.10 msec. The worst devices were all serial (Mice 1, 2, 3, and 6a), with transmission times between 24 and 33 msec. PS/2 mice (Mice 4, 5a, 6b, 7, and 8) ranged between 3.6 and 6.16 msec.

Two of the mice tested offered a choice of interface by means of an in-line adapter. Mouse 5a/5b, for example, could switch between PS/2 and USB interfaces. Mouse 6a/6b could switch between PS/2 and serial. Surprisingly, changing the interface not only affected transmission time, but also markedly affected the internal timing characteristics of the mouse itself. Effectively, you have two mice in the same shell, since two pieces of microcode within the mouse firmware independently take over mouse control according to the chosen interface. The biggest difference in maximum button-down latencies occurred with Mouse 5. When it was operating in USB mode, latencies rose from 18 to 39.2 msec. Minimum latencies largely remained unchanged at 9 and 10.5 msec, respectively. Within Mouse 6, changing the interface from PS/2 to serial had little effect, with the serial interface raising minimum and maximum latencies only by just over a millisecond in both cases.

By way of a comparison, a standard ALPS branded keyboard was also bench-tested. This had a minimum key-down latency of 22.5 msec, a maximum delay of 31.8 msec, and a theoretical transmission time of 1.1 msec. As the keyboard scans a matrix, all keys should be within these limits, with a key scan occurring approximately every 10 msec.

Since E-Prime was to be used to test each mouse under a real experimental paradigm, Psychology Software Tool's own deluxe response box was also bench-tested. This was done to establish a yardstick by which mice and other input devices could be judged. The results are also shown in Table 1.

Testing Response Devices With a Standard Experiment Generator

Of the original pool of eight mice, four were tested as response devices in the simple E-Prime paradigm. In addition, the keyboard and PST's own response box were used. As can be seen from Table 2, the results from each of the four mice closely mirrored their timing characteristics, already established during the hardware bench-testing. Similarly, both the keyboard and the PST response box mirrored their bench-tested characteristics. The poorest mice remained the poorest, and the best remained the best. Differences between the last column of Table 1, which identifies the typical device delay, and the first row of Table 2, which identifies the mean delay measured by E-Prime, highlight the contribution made to timing over and above that inherent in the device itself

Table 3
Corrected E-Prime Response Target Error Times as Compared With Benchmark Predicted Timings (in Milliseconds)

Our Reference	Bench Test Prediction	E-Prime (Corrected)	Difference
Mouse 1	38.00	37.81	0.19
Mouse 6a	35.05	34.20	0.85
Mouse 6b	6.55	5.15	1.40
Mouse 5a	29.85	30.13	−0.28
Mouse 8	61.60	61.30	0.30
Response box (BB1)	1.17	1.44	−0.27
Keyboard (KB)	28.25	29.35	−1.10

Note—The bench test prediction was based on the midpoint of the minimum and the maximum measures plus the transmission time (see Table 1). The E-Prime (corrected) was based on the mean of 80 trials minus a correction factor of 5 msec (see Table 2).
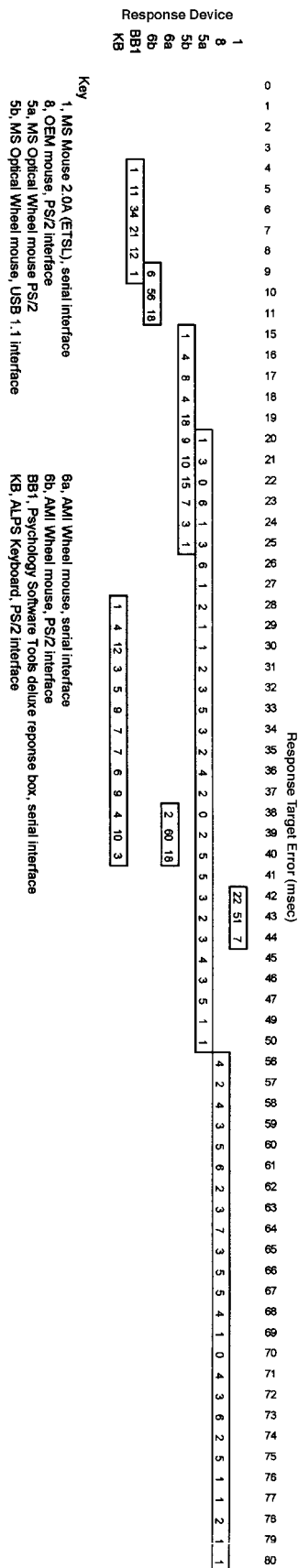
(see Table 3 for a direct comparison). In short, these differences could be due to anything "downstream" of the device—for example, the version of MS Windows used, the interface type, the chipset, the device driver, and not least, the software package itself.

In order to illustrate the absolute error and variability of each device, we calculated the frequency of each RTE value (to the nearest whole millisecond) recorded by E-Prime for each device. These frequencies are shown in Figure 3. Each row in the figure represents one device, with the spread of values indicated by the length of the enclosing box and the frequencies for each RTE value given as numbers within the box.

Figure 4 illustrates the variability of the best and worst mouse performances across the 80 trials. Data for the keyboard and the PST response box are included for comparison.

Next, basic statistics were applied to the E-Prime RTE data. As was expected, a one-way analysis of variance (with trials as the sampling variable) confirmed that there was a highly significant difference between all the response devices tested with E-Prime ($p < .001$). An independent t test confirmed that there was a significant difference between the best (Mouse 6b) and worst (Mouse 8) performing mice ($p < .001$).

A second independent t test confirmed there was a significant difference between the worst mouse (Mouse 8) and the standard keyboard ($p < .001$), with the standard keyboard proving significantly better than this mouse in terms of RTE. However, the MS optical mouse in PS/2 mode (Mouse 5a) was not significantly better than the keyboard, although it was different when in USB mode (Mouse 5b; $p < .001$).

The two mice (5 and 6) that could operate via an alternative interface were compared in each mode. An independent t test showed a significant difference in RTE between the MS optical mouse in serial (5a) and USB (5b) modes. Furthermore, there was also a significant difference between the AMI mouse operating in serial (6a) and PS/2 (6b) modes.

Given the remarkable performance of the AMI mouse in PS/2 mode (Mouse 6b), an independent t test was carried out to see whether this was on a par with the PST response box. However, this suggested that there was still a significant difference ($p < .001$), with the response box still beating even the best mouse in absolute terms.

Finally, we sought to explore and clarify how closely these error timings related to the specific device characteristics. In order to do so, we applied several manipulations to these data. First, a top-of-screen to mid-screen factor was calculated and then subtracted from the E-Prime mean RTE. Since the monitor of the PRES was driven at 100 Hz, a screen refresh took 10 msec. For the beam to scan to the fiber-optic detector mid-screen and illuminate a few pixels, this would take around 5 msec. Hence, 5 msec was subtracted to give an E-Prime (corrected) RTE. Next, a mean predicted RTE was calculated
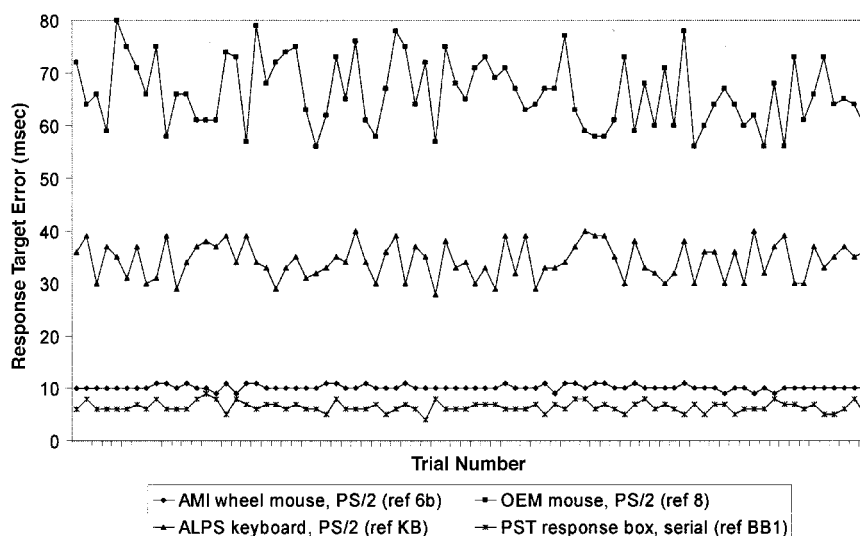

Figure 3. A graphical representation of the frequency and spread of response target errors recorded by E-Prime 1.0.

Key
1, MS Mouse 2.0A (ETSL), serial interface
8, OEM mouse, PS/2 interface
5a, MS Optical Wheel mouse PS/2
5b, MS Optical Wheel mouse, USB 1.1 interface
6a, AMI Wheel mouse, serial interface
6b, AMI Wheel mouse, PS/2 interface
BB1, Psychology Software Tools deluxe reponse box, serial interface
KB, ALPS Keyboard, PS/2 interface

Figure 4. Variability of the best and worst mice, together with PST response box and standard keyboard for comparison.

from the hardware bench test data. This involved applying the following formula:

[(max button latency − min button latency)/2]

+ min button latency + transmission time.

Table 3 shows the E-Prime (corrected) RTE times and the predicted times according to the formula.

We can see that the predicted and the corrected timings are virtually identical, with an overall average difference between the two of only 0.16 msec when the mice alone are considered.

DISCUSSION

The variation between the best and the worst mice is striking, and not a little worrying for those who believe the mouse to be a reliable and consistent device for accurate timing. It implies that if one used one mouse in one study and another in a replication, this could result in a sizable absolute difference. There were large differences in variability too, with a standard deviation of 0.46 msec for Mouse 6a and 8.46 msec for Mouse 5a. Some mice were significantly worse than the keyboard (Mouse 8), whereas others were just as good (Mouse 5a). Complicating the picture still further is the fact that there were significant differences between the way the internal timing characteristics changed on the dual-interface mice (Mice 5 and 6). So, does this mean that all mice are now so variable that they should no longer be used as response devices when accurate timing is required? Not necessarily, since some mice are superb performers, with Mouse 6b achieving near button box latencies with extremely low variability, at a cost of around $15.

Many researchers have assumed that using standard Windows devices would add a disproportionate amount to the response time error over and above that contributed by the response device hardware. However, we have shown that the response time errors are almost exactly accounted for by the response device, and not by Windows or, in this case, by E-Prime. Using a reasonably fast PC (an Athlon 900) with Windows 98R2, the contribution to response time error from both Windows and E-Prime was around a millisecond, on average. It should be noted that E-Prime uses Microsoft DirectX (http://www.microsoft.com/directx) and so makes use of DirectInput API calls. Thus, the low response time error may be due, in part, to using DirectInput over standard methods. Even more remarkable is that this consistency is not affected by changing the interface or the response device. That is, changing from a PS/2 mouse to a serial mouse—or to a keyboard, for that matter—does not alter the contribution to response time error attributable to Windows or E-Prime, which remains constant at around a millisecond. In all cases, the response time error is always equal to that predicted by bench-testing the hardware, using a signal generator and a scope.

With the disappearance of the bus mouse, many researchers have looked to the serial mouse as a consistent replacement. However, these mice are not as consistent as originally thought, and these too are fast disappearing, with many newer laptops forgoing serial ports for PS/2 and USB (a fact that will affect many response boxes too). Originally, it was thought that PS/2 mice, by virtue of the interface, offered a poor solution. However, we have shown that certain PS/2 mice can achieve remarkable performance, with the best performing AMI mouse being a good example. However, these too may soon disappear, leaving us with USB peripherals across the board, covering both PC and Mac platforms. USB devices are accessed via the HID drivers in Windows. Internally, these check for signal changes at very high rates. However, the USB device in

question dictates when signals are sent to the HID drivers. As a result, there will continue to be differences in the performance of any USB peripheral. This is likely to affect mice, since many engineers rightly assume that it is not necessary to send out positional and button information at higher rates than necessary for normal mouse operation.

Newer optical mice also seem to be taking over from simple opto-mechanical ball mice. However, it is thought that these newer movement detection technologies actually affect the overall mouse performance, due to lower sampling rates within the mouse. Within the firmware, snapshots of the mouse mat/desktop are taken by a tiny camera many times per second. To calculate movement, each image is compared with the previous one, and within this processing loop, button-down events are detected. A second disadvantage is that one cannot remove the ball to prevent movement, which as Segalowitz and Graves (1990) have shown, more than doubles latencies for buttonpresses.

So is it simply a question of identifying a good mouse, publicizing its characteristics, and accepting this as the model to use, as was previously the case with the standard Microsoft mouse? Well again, this apparently simple question is fraught with difficulty. As part of the present research, we examined three Microsoft serial mice that were all apparently identical (Model 2.0A). However, on closer examination, each was manufactured in different countries—Taiwan, China, and Ireland—and had a different FCC ID. One might assume that they were manufactured to the same specifications but in different countries, due to cost and distribution factors. However, when we opened each mouse, we discovered different board and component designs (see Figure 1). This also suggests that the microcode in each differed and was likely to have been written by different engineers. Indeed, the timing characteristics of these mice, when bench-tested, confirmed this variability, with button-down latencies ranging from 4.9 to 13.7 msec and jitter ranging from 0.6 to 4.7 msec. Transmission times were fairly consistent at around 25 msec, by virtue of the serial interface. In much the same way, mice with the dual-interface capability have, in effect, two sets of microcode, which may have been written by different engineers, and thus two sets of timing characteristics, despite being run on the same hardware.

A trawl of the FCC database for Microsoft guarantee code (C3K) lists some 85 entries (as of May 2002) for various pieces of Microsoft branded hardware. Unfortunately, the FCC does not require that much information be kept on record and accessible to the public, but it is thought that a large proportion of these are registrations for Microsoft mice of some variety. A conservative estimate would number the mouse population at around 40. This introduces a margin of variability if researchers make use of different mice, or even variants of the same. If timing accuracy is at a premium, being able to replicate results from one year to the next will be difficult if the researcher changes mouse, never mind a change to the PC or the operating system.

The obvious answer to this thorny issue for time-critical research is to use a recognized response box, since this will usually be able to achieve millisecond accuracy internally. It is also likely to have higher quality buttons for faster mechanical action, have better debouncing circuitry, take account of hysteresis, and be better able to cope with short interkey intervals. However, it is not always viable to equip a large teaching lab with response boxes for every machine. In such cases, we think mice with a low variability and low absolute latency should be used; but the timing characteristics of the hardware should be known, so that adjustments to recorded times can be made. We side with Beringer (1992), who suggested that "mouse-dependent results show that differences between mouse hardware must not be neglected" (p. 489). He went on to say that "there is no guarantee that there will be no variations between serial Microsoft mice with the respect to the internal logic." He proposes that "individual testing of installed equipment is therefore recommended . . . regardless of the brand" or, indeed, of the model, as we have shown. We have outlined a relatively simple methodology for doing this with serial, USB, and PS/2 mice, with the aid of a signal generator and an oscilloscope. For those without access to or the knowledge to make use of such equipment, the authors are currently investigating the feasibility of producing a generic timing validation rig for use by the wider research community.

It is worth noting that the first author can identify poorly performing mice by examining their movement and acceleration profiles within Windows, and others should be able to do so with a little training. Mice with a poor internal microcode or slow scanning hardware can be identified by poor movement. Button-down detection and handling takes place somewhere in this movement detection loop, and the best mice have the smoothest movement and the highest sampling rates (in DPI), which tends to give rapid desktop movement to small mouse movements. The on-line gaming community is quite fanatical about mouse performance and shuns all optical mice, favoring good quality opto-mechanical ball mice, such as our best performing AMI mouse.

As the old adage goes, don't judge a book by its cover or, indeed, a mouse by its shell.

REFERENCES

Beringer, J. (1992). Timing accuracy of mouse response registration on the IBM microcomputer family. *Behavior Research Methods, Instruments, & Computers*, **24**, 486-490.

Crosbie, J. (1990). The Microsoft mouse as a multipurpose response device for the IBM PC/XT/AT. *Behavior Research Methods, Instruments, & Computers*, **22**, 305-316.

MacInnes, W. J., & Taylor, T. L. (2001). Millisecond timing on PCs and Macs. *Behavior Research Methods, Instruments, & Computers*, **33**, 174-178.

Myors, B. (1998). A simple graphical technique for assessing timer ac-

curacy of computer systems. *Behavior Research Methods, Instruments, & Computers*, **30**, 454-456.

MYORS, B. (1999). Timing accuracy of PC programs running under DOS and Windows. *Behavior Research Methods, Instruments, & Computers*, **31**, 322-328.

PLANT, R. R., HAMMOND, N., & WHITEHOUSE, T. (2002). Toward an Experimental Timing Standards Lab: Benchmarking precision in the real world. *Behavior Research Methods, Instruments, & Computers*, **34**, 218-226.

SEGALOWITZ, S. J., & GRAVES, R. E. (1990). Suitability of the IBM XT, AT, and PS/2 keyboard, mouse, and game port as response devices in reaction time paradigms. *Behavior Research Methods, Instruments, & Computers*, **22**, 283-289.

ULRICH, R., & GIRAY, M. (1989). Time resolution of clocks: Effects on reaction time measurement—good news for bad clocks. *British Journal of Mathematical & Statistical Psychology*, **42**, 1-12.