Response time accuracy in Apple Macintosh computers

Ian Neath · Avery Earle · Darcy Hallett ·
Aimée M. Surprenant

Published online: 17 March 2011
© Psychonomic Society, Inc. 2011

Abstract The accuracy and variability of response times (RTs) collected on stock Apple Macintosh computers using USB keyboards was assessed. A photodiode detected a change in the screen's luminosity and triggered a solenoid that pressed a key on the keyboard. The RTs collected in this way were reliable, but could be as much as 100 ms too long. The standard deviation of the measured RTs varied between 2.5 and 10 ms, and the distributions approximated a normal distribution. Surprisingly, two recent Apple-branded USB keyboards differed in their accuracy by as much as 20 ms. The most accurate RTs were collected when an external CRT was used to display the stimuli and Psychtoolbox was able to synchronize presentation with the screen refresh. We conclude that RTs collected on stock iMacs can detect a difference as small as 5–10 ms under realistic conditions, and this dictates which types of research should or should not use these systems.

Keywords Reaction time · Response time · Measurement · Experimentation · Computers

Researchers studying vision or other areas of psychophysics have long been sensitive to issues of measurement, especially response time (RT), accuracy, and various properties of the visual display. For researchers in other areas of experimental psychology, however, many of the issues are less critical. For example, in a standard memory experiment, the researcher is more often interested in relative speeds of responding in two (or more) conditions than in absolute time to respond. Similarly, although precise knowledge of the onset time

of a stimulus is important, the researcher does not necessarily worry about other factors that may be critical in psychophysical studies (e.g., luminance or saturation). Although information is available about individual LCD displays or computers running Microsoft Windows (e.g., Plant & Turner, 2009) or older versions of Apple's operating system (e.g., MacInnes & Taylor, 2001), we could find no publications examining timing accuracy and variability for systems running recent versions of Mac OS X. The purpose of the present work was to answer the following question: How accurate are RTs collected on Apple Macintosh computers?

The basic logic of the tests reported is as follows. In an experiment, the program starts a clock and displays a stimulus on the screen. The subject presses a key on the keyboard, and the clock is stopped. The dependent variable is the RT, the difference between when the clock started and when it stopped. We replaced the subject with a device that always took the same amount of time to press a key on the keyboard once the stimulus was shown. Any variation observed in the measured RTs must be due to variability attributable to a combination of the computer hardware (e.g., display, USB bus, and keyboard) and software (e.g., operating system and specific program running the experiment). This is similar to the method of De Clercq, Crombez, Buysse, and Roeyers (2003), except that we used dedicated hardware expressly constructed for the purpose, whereas they used a second general-purpose computer.

Hardware and software

Technical Services at Memorial University of Newfoundland built a custom testing box that consisted of the following: A photodetector monitored the display for a

I. Neath (✉) · A. Earle · D. Hallett · A. M. Surprenant
Psychology Department, Memorial University of Newfoundland,
St. John's, NL A1B 3X9, Canada
e-mail: ineath@mun.ca

change in luminance on the screen; when this occurred, a relay was activated that in turn activated a solenoid. The solenoid was positioned over the keyboard and pressed a key.

Details and calibration The testing box used a PIC 16F877 microcontroller, and the T2 timer was set to interrupt at 1 kHz. For the purposes of calibration, the testing box also included an LED and a single test button. On a calibration run, the solenoid was positioned over the onboard test button and the photodetector was pointed at the LED. The microcontroller set logical 1 at one of its outputs, which caused the LED to light. The photodetector was monitored at one of the microcontroller's analog inputs. When the photodetector output voltage reached a value chosen to be 2/3 of a typical scale reading, a second output of the microcontroller was set to logical 1. This second output activated a relay, and in turn a solenoid, which pressed the test button. The buttonpress grounded another microcontroller input, which was normally at logical 1. Detection of logical 0 at this input was monitored by the microcontroller, which reported the number of 1-ms periods elapsed.

This basic cycle—from setting a pin high in order to light the LED to noting that the test button had been pressed—was measured in two independent ways. First, the microcontroller always reported a time of 38 ticks of a 1-kHz timer. Second, a Tektronix oscilloscope (model TDS2012B) reported a value of 38.0 ms. Thus, all results reported below have this 38-ms value subtracted from the measured time. With an ideal system, the resulting latencies should be 0 ms; any value higher than 0 is due to the computer/monitor/software.

Computers Two iMac computers were assessed, a recent model and an eight-year-old model. The newer model was a 24-in. iMac with the model identifier iMac8,1, sold between April 2008 and March 2009.¹ This machine has a 2.8-GHz Intel Core 2 Duo processor, a bus speed of 1.07 GHz, and 2 GB 800-MHz DDR2 SDRAM. The graphics card is an ATI Radeon HD 2600 Pro driving a 24-in. glossy TFT Active Matrix LCD (1,920 × 1,200). This iMac was running Mac OS X 10.6.3 (build 10D573) with all software updates installed as of May 5, 2010.

The older model was a 15-in. iMac with the model identifier PowerMac4,2, sold between January 2002 and February 2003.² This machine has a 700-MHz G4 processor, a bus speed of 100 MHz, and 512 MB PC133 SDRAM. The graphics card is an NVIDIA GeForce2 MX powering a 15-

in. TFT Active Matrix LCD display (1,024 × 768). This iMac was running Mac OS X 10.4.11 (build 8S165) with all software updates installed as of May 5, 2010.

Keyboards Two types of Apple-branded keyboards were tested: The currently available (as of 2010) aluminum USB keyboard (Model A1243), which came with the 24-in. iMac, and the previous generation white USB keyboard (Model A1048). Note that the white keyboard was one generation more recent than that which originally came with the 15-in. iMac. An Apple-branded USB mouse (A1152) remained attached during the tests.

Software Three tests used Psychtoolbox 3.0.8,³ two when running under MATLAB 7.9.0.529 (R2009b)⁴ and the third when running under GNU Octave 3.2.3.⁵ For the 24-in. iMac, the JavaScript, Java, and Flash tests were run via Safari 4.0.5 (6531.22.7), the version of Java was 1.6.0_17, and the version of the Flash player was 10.0 r42. For the 15-in. iMac, the JavaScript, Java, and Flash tests were run via Safari 4.0.5 (4531.22.7), the version of Java was 1.5.0_19, and the version of the Flash player was 9.0 r246. The Flash program was created with Adobe CS4 Professional 10.0.2, and the movie frame rate was left at the default of 12 fps.

While the tests were being conducted, no other user-initiated software was running except for the following: MATLAB requires that the X11 application be running, and Octave runs inside a Terminal.app window. Unnecessary services (e.g., Bluetooth, wireless, Time Machine, and file sharing) were turned off, but the computer remained connected to the Internet via an ethernet cable. There was no antivirus software running.

Basic test The photodetector was placed approximately 5 mm from the surface of the screen in the upper left-hand corner, approximately 5 cm down and 5 cm over. The photodetector remained in this location for all tests on a given computer/display. The display was set to maximum brightness, and all lights in the testing room were extinguished (the room had no windows). The solenoid was positioned such that when it was fully extended, the key below was fully pressed down. For each type of software (e.g., Psychtoolbox, JavaScript, Java, Flash), the program waited a random amount of time between 1 and 1,000 ms. Then, either the entire screen changed from black to white (for Psychtoolbox tests) or a small region changed from black to white (for JavaScript, Java, and Flash tests),

¹ See www.everymac.com/systems/apple/imac/stats/imac-core-2-duo-2.8-24-inch-aluminum-early-2008-pennyn-specs.html.

² See www.everymac.com/systems/apple/imac/stats/imac_700_fp.html.

³ See <http://psychtoolbox.org/>.

⁴ See www.mathworks.com.

⁵ GNU Octave is a freely available numerical computing language that is mostly compatible with MATLAB. See www.gnu.org/software/octave/.

and the software started a timer. The photodetector triggered the solenoid, which pressed a key. The software stopped the timer as soon as any key was pressed, and the difference between the start and stop times was recorded. The screen (or small region) then changed from white to black, and the software waited 4 s before starting the next trial. The purpose of this 4-s waiting period was to allow the solenoid to come to a complete stop. This loop was executed 1,000 times for each assessment.

Assessment 1

The first assessment addressed reliability. The 24-in. iMac and the default aluminum keyboard that came with it were used, and the software was Psychtoolbox running under MATLAB. For each of five runs, the computer was first rebooted under a nonadministrator account, and only MATLAB was running.

Cumulative frequency distributions for each set of 1,000 observations are shown in the top panel of Fig. 1, and descriptive statistics for each run are shown in the top half of Table 1. As is readily apparent, there was little variation from one run to the next. The mean RT varied from a low of 39.456 ms to a high of 39.711 ms, a difference of 0.255 ms. The standard deviation was quite small, ranging from a low of 2.631 ms to a high of 2.777 ms, a difference of 0.146 ms. Importantly, the skewness was always close to 0, varying from -0.025 to 0.049 . The standard error of the skewness statistic (which, because it is completely dependent on sample size, is the same for all of our tests) was 0.077. Because the skewness figures are well within two standard errors of 0, this suggests a largely unbiased distribution.

We compared the five observed distributions to five generated from a normal distribution. Five cumulative frequency distributions, each with 1,000 values drawn randomly from a normal distribution with a mean of 39.570 and a standard deviation of 2.704, were generated using the following MATLAB function:

```
random('norm', 39.570, 2.704, 1, 1000)
```

Descriptive statistics for each are shown in the bottom half of Table 1. There is little difference between the observed and generated RTs except for the range, which increased from a mean of 13.605 for the observed to a mean of 18.047 for the generated. The means, standard deviations, quartile, and skewness measures are all comparable; if anything, the generated distributions show a slightly wider range and more variability in skewness than the observed distributions.

Fig. 1 Cumulative frequency distributions for five replications measuring RT on an Intel iMac computer using MATLAB and Psychtoolbox (*top figure*). The bottom figure shows all 5,000 observations in one cumulative frequency distribution and also plots a cumulative frequency distribution generated by sampling randomly from a normal distribution using the MATLAB command `random('norm', 39.570, 2.704, 1, 1000)`

The bottom panel of Fig. 1 shows the cumulative frequency distribution for all 5,000 measured RTs combined and all 5,000 generated RTs combined. While there are a few minor differences, the observed function approximates the generated function quite closely. The right-most column in Table 1 shows the descriptive statistics for these combined functions. With the exception of the range, the numbers are quite comparable.

This assessment shows that RTs collected on a 24-in. iMac and default aluminum keyboard with MATLAB and Psychtoolbox produce RTs that are, on average, approximately 40 ms too long. However, there are no noticeable differences between different runs, the standard deviation is quite small, and the cumulative frequency distribution is similar to one generated from a normal distribution.

Table 1 Measures of central tendency, range, and standard deviation for each of five replications using Psychtoolbox running under MATLAB on a 24-in. iMac with an aluminum USB keyboard (top) and for five distributions randomly generated from a normal distribution using MATLAB (bottom)
(See the text for details)

	1	2	3	4	5	Combined
Mean	39.456	39.633	39.711	39.495	39.557	39.570
Stdev	2.720	2.777	2.712	2.631	2.676	2.704
Skewness	0.025	0.049	0.005	0.029	-0.025	0.019
Range	13.509	13.478	13.946	13.667	13.425	15.907
Min	32.873	33.079	33.485	31.524	33.416	31.524
Max	46.382	46.557	47.431	45.191	46.840	47.431
1st Quartile	37.441	37.640	37.570	37.638	37.424	37.558
2nd Quartile	39.706	39.340	39.818	39.024	39.695	39.711
3rd Quartile	41.456	41.594	41.458	41.354	41.856	41.493
Mean	39.649	39.421	39.584	39.382	39.671	39.541
Stdev	2.702	2.706	2.721	2.752	2.725	2.723
Skewness	0.064	-0.091	0.076	-0.040	0.055	0.012
Range	18.726	17.732	17.070	18.326	18.381	19.200
Min	29.774	31.192	31.295	30.648	30.087	29.774
Max	48.500	48.925	48.364	48.973	48.468	48.973
1st Quartile	37.836	37.572	37.735	37.525	37.899	37.702
2nd Quartile	39.517	39.556	39.681	39.401	39.600	39.535
3rd Quartile	41.474	41.348	41.343	41.254	41.545	41.390

Assessment 2

The second assessment compared two different Apple-branded keyboards and also compared Psychtoolbox running under MATLAB and Octave. One set of tests was run using the same keyboard as in [Assessment 1](#) (i.e., the aluminum USB keyboard that was still shipping in 2010); a

second set of tests were run using the previous-generation white USB keyboard. In addition, one set of tests ran Psychtoolbox under MATLAB, which is proprietary commercial software, and the second set ran the same Psychtoolbox code under Octave, which is mostly compatible with MATLAB but is freely redistributable under the terms of the GNU General Public License.

Fig. 2 Cumulative frequency distributions for measured RTs using Psychtoolbox under MATLAB (*top figures*) and Octave (*bottom figures*) and using Apple's current aluminum USB keyboard (*left figures*) and previous-generation white USB keyboard (*right figures*)

Table 2 Measures of central tendency, range, and standard deviation for RTs measured on two types of keyboards (aluminum or white) and two kinds of software running Psychtoolbox (MATLAB or Octave)

	Aluminum		White	
	MATLAB	Octave	MATLAB	Octave
Mean	39.456	40.360	19.689	19.576
Stdev	2.720	2.666	2.668	2.732
Skewness	0.025	0.061	0.027	0.095
Range	13.509	13.189	12.631	11.878
Min	32.873	33.591	13.365	13.811
Max	46.382	46.779	25.996	25.689
1st Quartile	37.441	38.668	17.464	17.753
2nd Quartile	39.706	40.222	19.719	19.297
3rd Quartile	41.456	42.576	21.833	21.675

Cumulative frequency distributions are shown in Fig. 2, and descriptive statistics are shown in Table 2.⁶ Surprisingly, there was a large difference in accuracy between the two Apple-branded keyboards: RTs collected via the white keyboard were on average 20 ms faster than those collected on the aluminum keyboard; the standard deviations were approximately the same.⁷ There was no consistent difference between MATLAB and Octave. Although the difference in skewness between MATLAB and Octave with the white keyboard looks large (0.027 vs. 0.095, respectively), the latter value is well within the range of values observed when generating distributions (see the bottom part of Table 1), and both are well within two standard errors of 0.

There were no differences observable between running Psychtoolbox code under MATLAB or under Octave, but there was a large difference between the two most recent Apple-branded keyboards, with the older keyboard yielding more accurate RTs.

Assessment 3

One advantage of Psychtoolbox is that the presentation of stimuli on a display can be synchronized with the vertical refresh. To the extent that the software is successful, the start of the timer coincides with the actual display, and therefore most of the remaining timing noise may be attributable to issues with the USB keyboard. The third assessment examined accuracy

and variability in RTs when the stimuli were shown on an external CRT rather than on the built-in LCD.

The 24-in. iMac supports a second display, and a ViewSonic Professional Series PF790 CRT was connected via a mini-DVI to a VGA adapter. The CRT was set to a resolution of $1,024 \times 768$ at 85 Hz. In addition, we compared mirrored (i.e., both displays set to the same resolution of $1,024 \times 768$ and showing the same images) and nonmirrored (i.e., the built-in LCD set to $1,920 \times 1,200$ and the CRT set to $1,024 \times 768$, with the MATLAB main window displayed on the LCD and the “stimuli” shown on the CRT) modes. Both the white and aluminum keyboards were tested.

Cumulative frequency distributions are shown in Fig. 3, and descriptive statistics are shown in Table 3. RTs were reduced by approximately 8–9 ms by using a CRT rather than the built-in LCD display. The marked difference in accuracy between the two keyboards remained, and the standard deviations and skewness were similar to those observed previously. As can be readily seen, running as mirrored or not mirrored had no observable effect.

The RTs were faster when an external CRT was used rather than the built-in LCD display. The fastest RT detected with the white keyboard was just 5.6 ms with the CRT, compared with 13.4 ms with the built-in display. The comparable values for the aluminum keyboard were 23.9 and 32.9 ms.

Assessment 4

The first three assessments all used Psychtoolbox running under MATLAB (or Octave). The remaining three assessments looked at three different languages for delivering Web-based experiments: JavaScript, Java,⁸ and Flash. For each, we tested both the recent 24-in. iMac used in the three previous assessments and an eight-year-old 15-in. iMac. Once again, we compared the newer aluminum and older white keyboards. **Assessment 4** examined JavaScript.

Cumulative frequency distributions are shown in Fig. 4, and descriptive statistics are shown in Table 4. Despite the huge difference in processor speed between the two iMacs, the RTs were only about 10 ms slower on the older G4 iMac than on the newer Intel iMac. In contrast, the difference between the two keyboards was twice as large, with the same 20-ms difference seen in previous assessments.

Unlike in previous assessments, the skewness measures for the RTs collected on the G4 iMac were well outside the

⁶ The RTs listed as “MATLAB – Aluminum” are the same as those shown as Replication 1 in **Assessment 1**.

⁷ We used three different aluminum keyboards (all Model A1243) and two different white keyboards (all Model A1048). We detected no differences between keyboards of the same model, but always observed the same magnitude difference between keyboard models.

⁸ JavaScript and Java are not related, despite some superficial similarities, including use of the word *Java* in both names. JavaScript is a dynamic object scripting language developed by Netscape, whereas Java applications and applets are compiled bytecode, and the language is developed by Sun Microsystems (now part of Oracle).

Fig. 3 Cumulative frequency distributions for measured RTs using Psychtoolbox under MATLAB using an external CRT that mirrors (*top figures*) or does not mirror (*bottom figures*) the built-in display and using Apple's current aluminum USB keyboard (*left figures*) and previous-generation white USB keyboard (*right figures*)

range of those from the generated distributions in Assessment 1, and were also more than two standard errors from 0. This was due primarily to a larger range at the higher end of the distribution than in the distributions from the Intel iMac. For both computers and both keyboards, the standard deviations were almost twice as large as those seen in previous assessments.

Assessment 5

The fifth assessment focused on collecting data over the Internet using Java applets (e.g., Stevenson, Francis, &

Kim, 1999) and again compared performance on the recent 24-in. iMac to that on an eight-year-old 15-in. iMac.

Cumulative frequency distributions are shown in Fig. 5, and descriptive statistics are shown in Table 5. RTs collected using Java were slower than those collected using JavaScript, and the pattern was also quite different. With the white keyboard, there was no difference between RTs collected on the Intel versus the G4 iMac, but with the aluminum keyboard, there was a small difference of approximately 3 ms. However, the skewness measures were all much larger for the slower computer (between 0.255 and 0.872), exceeding two standard errors away from 0. Like the JavaScript results for the G4 iMac in the previous assessment, the large skewness values

Table 3 Measures of central tendency, range, and standard deviation for RTs measured with MATLAB and Psychtoolbox with an aluminum or white keyboard when stimuli were shown on a CRT set at 85 Hz and the built-in display either mirrored or did not mirror the CRT

	Aluminum		White	
	Mirrored	Not Mirrored	Mirrored	Not Mirrored
Mean	30.408	30.246	11.869	11.882
Stdev	2.751	2.651	2.612	2.642
Skewness	-0.026	0.001	-0.008	-0.108
Range	13.346	12.776	12.368	12.616
Min	23.929	23.942	5.570	5.825
Max	37.274	36.718	17.939	18.441
1st Quartile	28.519	27.973	10.335	9.785
2nd Quartile	30.139	30.278	11.763	12.065
3rd Quartile	32.430	32.403	14.093	13.768

Fig. 4 Cumulative frequency distributions for measured RTs on a G4 iMac (*top figures*) or Intel iMac (*bottom figures*) using JavaScript running in Safari and using Apple's current aluminum USB keyboard (*left figures*) and previous-generation white USB keyboard (*right figures*)

were due to a larger range at the higher end of the distribution. The large difference between the white and aluminum keyboards remained.

Assessment 6

The final assessment examined Flash (e.g., Reimers & Stewart, 2007) and once again compared performance on the recent 24-in. iMac to that on an eight-year-old 15-in.

Table 4 Measures of central tendency, range, and standard deviation for RTs measured with JavaScript on two types of keyboards and two different iMac computers

	Aluminum		White	
	Intel	G4	Intel	G4
Mean	55.210	65.514	33.947	45.615
Stdev	5.928	5.866	5.857	6.108
Skewness	0.003	0.302	0.006	0.332
Range	29	36	30	37
Min	41	52	19	30
Max	70	88	49	67
1st Quartile	51	61	30	41
2nd Quartile	55	65	34	46
3rd Quartile	60	70	38	50

iMac. Cumulative frequency distributions are shown in Fig. 6, and descriptive statistics are shown in Table 6. There were considerable differences between the two computers, both in the overall mean as well as in the standard deviation. For some runs, the standard deviations exceeded 10 ms. The difference between the two keyboards remained.

There was an odd pattern observed on the Intel iMac. The cumulative distribution function displayed some scalloping, especially with the newer keyboard. For example, whereas an RT of 56 ms was observed 103 times, the corresponding frequencies for 57, 58, 59, and 60 ms were 83, 39, 2, and 107, respectively. When replotted as a frequency distribution, this pattern is more evident, as shown in Fig. 7. Since this was the only time this pattern was detected, this assessment was rerun. Although the specific values changed, the scalloping remained.

Simulation

One key finding from Assessment 1 is that distributions of RTs collected in each assessment are quite similar to what is expected if the same number of observations are randomly sampled from a normal distribution. We took advantage of this in order to estimate how many observations would be necessary in order to produce a mean RT that was statistically different from a given RT for different standard

Fig. 5 Cumulative frequency distributions for measured RTs on a G4 iMac (*top figures*) or Intel iMac (*bottom figures*) using Java applets running in Safari and using Apple's current aluminum USB keyboard (*left figures*) and previous-generation white USB keyboard (*right figures*)

deviations. Using MATLAB, we randomly sampled from a normal distribution with mean = 40 ms and standard deviation varying from 1.0 to 10.0 ms by 0.1-ms increments. We varied the number of RTs randomly drawn from 3 to 100. We computed whether the mean of the sample was significantly different, by a *t* test, from a mean that differed from 40 ms by 1, 5, 10, and 20 ms. We did this 1,000 times for each standard deviation and for each to-be-detected magnitude. Figure 8 shows the results. The *x*-axis shows the standard deviation (in milliseconds), and the *y*-

axis shows the number of observations necessary to detect a 1-, 5-, 10-, or 20-ms difference consistently.

As can be seen, more than 100 observations are necessary to detect a 1-ms difference once the standard deviation is larger than 2 ms. However, larger differences can be consistently detected with fewer observations, as long as the standard deviation remains small. It should be kept in mind that the criterion of “consistently detected” means that each of the 1,000 statistical tests in one run resulted in a *p* of .05 or less.

Although this simulation depends on certain assumptions, it does offer one way of estimating the number of observations required in order to detect a particular difference in RTs. If the particular combination of hardware and software results in an approximately normal distribution with a small standard deviation, then a 1-ms difference can be consistently detected. However, given that the smallest standard deviation we observed in any of the assessments was on the order of 2.6 ms, the smallest difference in magnitude that a stock iMac could detect under reasonable conditions is approximately 5–10 ms.

One way of reading the figure is to note that with a large difference in magnitude between the two RTs of interest, the standard deviation of the distribution of measured RTs does not matter much; that is, a real difference of 50 ms will be consistently detected with just a few measurements, even with a standard deviation larger than any we observed. While this is a reasonable reading, we prefer to emphasize an alternate view: The smaller the difference in RTs, the

Table 5 Measures of central tendency, range, and standard deviation for RTs measured with Java on two types of keyboards and two different iMac computers

	Aluminum		White	
	Intel	G4	Intel	G4
Mean	70.423	73.621	51.414	51.431
Stdev	5.803	7.288	5.957	7.596
Skewness	0.314	0.872	0.255	0.643
Range	35	38	35	41
Min	59	60	38	36
Max	94	98	73	77
1st Quartile	66	68	47	46
2nd Quartile	70	72.5	51	51
3rd Quartile	75	77	56	56

Fig. 6 Cumulative frequency distributions for measured RTs on a G4 iMac (*top figures*) or Intel iMac (*bottom figures*) using Flash running in Safari and using Apple's current aluminum USB keyboard (*left figures*) and previous-generation white USB keyboard (*right figures*)

more critical it is to know the properties of the timing device used.

Discussion

Some research areas require highly specialized equipment for displaying stimuli and collecting RTs (e.g., tachistoscopes, high-end CRT displays, and dedicated response boxes). Although these tools are essential for many kinds of psychophysical and perceptual work, in other areas many aspects of the display or response apparatus are not critical

to the questions being asked or to the conclusions that are made. Many memory paradigms fall into the latter category. However, prior to this report, there was no available information on the accuracy and characteristics of RTs collected on stock Apple Macintosh computers running Mac OS X and equipped with Apple-branded keyboards. Thus, the quality of RT data collected using these machines was unknown. The assessments presented here allow the researcher to make informed decisions about the types of hardware needed to answer the questions that are being investigated.

We found that RT distributions collected using Psychtoolbox were comparable to distributions generated from

Table 6 Measures of central tendency, range, and standard deviation for RTs measured with Flash on two types of keyboards and two different iMac computers

	Aluminum		White	
	Intel	G4	Intel	G4
Mean	54.350	70.009	37.806	52.089
Stdev	5.338	10.400	5.599	11.232
Skewness	−0.008	0.034	0.076	0.040
Range	27	51	28	49
Min	41	45	24	29
Max	68	96	52	78
1st Quartile	50	61	34	44
2nd Quartile	54	70	38	52
3rd Quartile	58	78	42	61

Fig. 7 Frequency distribution for RTs collected on the Intel iMac with the current aluminum keyboard running Flash

Fig. 8 Simulation results (see the text for details) plotting the number of observations needed to consistently detect a 1-, 5-, 10-, or 20-ms difference as a function of the standard deviation of the normal distribution from which the observations were sampled. Note: This simulation takes into account only hardware/software variability

a normal distribution. The most accurate RTs occurred using Psychtoolbox (running under either MATLAB or Octave), an external CRT, and the older white keyboard, but even when the built-in display was used, the distributions remained largely unchanged except for the mean.

Both JavaScript and Java resulted in larger standard deviations, and both showed far larger measures of skewness, always in the positive direction. Some issues were noted with Flash; at least part of this result could be due to the particular implementation. An examination of RTs collected with Flash on an iMac leads to a different conclusion than when RTs were collected on a Windows computer (see, e.g., Reimers & Stewart, 2007).

Surprisingly, we also found a large difference between two Apple-branded keyboards. This difference was larger than the difference in accuracy between a current and an eight-year-old computer. As noted in other studies (e.g., Plant, Hammond, & Turner, 2004), if half of the subjects in a study used one type of keyboard and the remaining half used the second type of keyboard, the RTs of the two groups would be statistically different.

Our simulation results are consistent with previous examinations of clock resolution. For example, Ulrich and Giray (1989, p. 11) concluded that the time resolution of a clock has “almost no effect on detecting mean RT differences even if the time resolution is about 30 ms or worse.” The data in Fig. 8 provide a resource so that a researcher can make a more informed evaluation of whether the likely differences in RTs can be observed with a stock Apple Macintosh computer, and also reveal the increasing importance of validating the timing in a particular experiment as

the magnitude of the RT difference of interest decreases. The smallest magnitude likely to be detected consistently is on the order of 5–10 ms.

Should researchers conduct experiments on stock Apple Macintosh computers when the dependent variable is RT? Given the variability in RTs observed above, we strongly recommend that researchers using *any* computer to collect RTs should assess the accuracy and reliability of their chosen platform. It is always desirable to minimize sources of error, and therefore one should validate the system on which one is collecting data. Given our findings above, we can recommend using the particular hardware/software combinations tested in only some situations. To the extent that the research examines small differences, or that absolute measures of time are important, or that the properties of the visual display are critical, or that synchronizing two or more items is critical, then the answer must be no. However, if a researcher tests all subjects using the exact same hardware, if the focus is on relative rather than absolute RTs, if the differences in RTs in the conditions to be examined are expected to be fairly large (e.g., at least 20–40 ms), if only certain software is used, and if many properties of the visual display are not of critical importance, then the conclusions drawn from RT data collected on a stock iMac are likely to be the same as those drawn from RT data collected on custom or high-end hardware.

Author Note This work was supported, in part, by grants from NSERC to D.H., I.N., and A.M.S. We thank D.J.K. Mewhort for piquing our interest in this topic. Example source code is available from <http://memory.psych.mun.ca/research/code.shtml> or from the first author.

References

- De Clercq, A., Crombez, G., Buysse, A., & Roeyers, H. (2003). A simple and sensitive method to measure timing accuracy. *Behavior Research Methods, Instruments, & Computers*, 35, 109–115.
- MacInnes, W. J., & Taylor, T. L. (2001). Millisecond timing on PCs and Macs. *Behavior Research Methods, Instruments, & Computers*, 33, 174–178.
- Plant, R. R., Hammond, N., & Turner, G. (2004). Self-validating presentation and response timing in cognitive paradigms: How and why? *Behavior Research Methods, Instruments, & Computers*, 36, 291–303.
- Plant, R. R., & Turner, G. (2009). Millisecond precision psychological research in a world of commodity computers: New hardware, new problems? *Behavior Research Methods*, 41, 598–614.
- Reimers, S., & Stewart, N. (2007). Adobe Flash as a medium for online experimentation: A test of reaction time measurement capabilities. *Behavior Research Methods*, 39, 365–370.
- Stevenson, A., Francis, G., & Kim, H. (1999). Java experiments for introductory psychology courses. *Behavior Research Methods, Instruments, & Computers*, 31, 99–106.
- Ulrich, R., & Giray, M. (1989). Measuring reaction times: Effects on reaction time measurement—Good news for bad clocks. *The British Journal of Mathematical and Statistical Psychology*, 42, 1–12.