The Timing Accuracy of General Purpose Computers for Experimentation and Measurements in Psychology and the Life Sciences

Andrew Wallace* and Guy Madison

Department of Psychology, Umeå University, SE-751 42, Sweden

Abstract: General purpose computers find increased use within behavioural, psychological, and neuroscientific experimentation, which raises concern for the timing accuracy that can be obtained with such systems. Here, we assessed the timing accuracy of such machines, considering both differences between different hardware and different versions of the Windows™ operating system (OS); Windows XP, Vista and Windows 7. The variability varied widely across machines and OS versions. The indeterministic variability within each OS and computer combination was mostly within +/-30 ms, and had a non-normal distribution with many small deviations and few large deviations. These large deviations are a characteristic feature that seems to constitute occasional additional delays up to about 150 ms. Thus, although measurements recorded from a general purpose PC running Windows should have an accuracy of -30 to +50 ms, occasionally larger variations suggest that experiments need a large test base to avoid significant distortions of the results.

Keywords: General purpose computers, Timing, Life Sciences, Windows™ operating system.

INTRODUCTION

General purpose computers, such as PCs and laptops with OSs targeted for business or entertainment offer a cost efficient way to conduct laboratory research. In fact, the lion's share of experiments in psychology, neuroscience, and the life sciences are nowadays conducted with such computers. However, the very attribute that make them a cost effective tool may also make them unsuitable for some specialist research projects; their general purpose nature. Here, we focus on the timing resolution that can be obtained with these computer systems, since this is often a crucial factor for the study of behaviour. Examples of time-critical study tasks include reaction time [1], implicit priming [2], event-related potentials in EEG [3], as well as time interval production [4] and discrimination [5]. In general, the timing resolution of human operators is proportional to the interval being timed, in accordance with Weber's law, on the order of 2-5 percent [6]. Human locomotion and action occurs predominantly on the time scale 0.3 to 1.5 s. This means that one standard deviation corresponds to between 6 and 75 ms for intervals in the range of human behaviour. The temporal resolution of human perception is comparable in magnitude, as indicated by a correspondence between production and discrimination studies [6, 7]. An established norm in behavioural research is therefore that the resolution of the experimental system should be 1 ms or less [8, 9].

Researchers have investigated the real time performance of general purpose computers. [10], for example, concluded that Windows NT has some limited use for real time applications but reported also indeterministic behaviour for some aspects of the OS. In [11], for example, the timing performance of the Ethernet card used within a general purpose PCs running different OSs was investigated. It was found that the OS could affect the utilisation of the available hardware which might affect timing depending on what hardware used in experiments. In [12], experiments were conducted to evaluate the real time performance of various systems including using general purpose PCs running Windows NT and Windows 2K and it was concluded that specifically designed real time OS performed better than general purpose OSs such as Windows.

Previous research into the performance of general purpose machines tends to evaluate the system from an engineering perspective where the manifest timing of the system is measured. The purpose of the present paper is to investigate the timing performance of general purpose computers within the realm of behavioural, psychological, and neuroscientific experimentation. In these applications, time measurements from participants' interaction with the computer's I/O is obtained from within computer programs and used as a dependent variable. Here, we also measure timing from within computer programs, recognising that such values should be most accurate as they do not include delay and variability from the I/O devices.

METHOD

Three different computers and three different versions of the Windows™ were used. All of these are presently in use at several sites in business and academic institutions. The computers were HP Compaq 8510p, HP Compaq DX MT

*Address correspondence to this author at the Department of Psychology, Umeå University, SE-751 42, Sweden; Tel: (0)73-620 5455; Fax: (0)90-786 66 95: E-mail: guy.madison@psy.umu.se

1874-3501/12 2012 Bentham Open

2000 and HP Omnibook 6100. The specifications of these computers are listed in Table **1**.

Each computer was combined with each of three different versions of the Windows™, namely Windows XP Pro SP 3, Windows Vista Pro, and Windows 7 Beta. In addition, the .NET framework 3.5 was also installed on all computers running Windows XP and Vista. Windows Vista and Windows 7 would not install on the HP Omnibook 6100.

Each computer ran "as is", with only those programs running that started as part of the OS after it was installed as well as the test program. In other words, no additional programs ran on the computer beyond the test software.

Four timing tests were developed, believed to tap different important aspects of timing accuracy in real-world tasks. The software was written in Visual Basic.Net using Microsoft Visual Studio 2008, and measured the duration of executing a loop, the sleep delay accuracy, the inter-thread message delay accuracy, and the time loop between pressing a key on the keyboard and an associated audio output.

Loop Test

In the loop test, the system executed a loop 9999999 times where it added 1 to a counter. The loop test was repeated 2000 times.

```
For i = 0 To 2000 - 1
 QueryPerformanceCounter(nTime)
 LoopTimes(inx) = nTime
 For j = 0 To 9999999
 Cnt = Cnt + 1
 Next
 QueryPerformanceCounter(nTime)
 LoopTimes(inx) = (CDbl(nTime) /          CDbl(nFrq)) _
 - (CDbl(LoopTimes(inx)) /                CDbl(nFrq)) _
 inx = inx + 1
Next
```

Sleep Test

The sleep test used the Windows Sleep function that forms part of the Thread class. Software engineers can encounter situations where a part of a program needs to wait for a certain amount of time before continuing. This sleep test measured the difference between the required delay and the actual delay recorded. The delay was incremented between each test. This test was repeated 2000 times.

```
For i = 0 To 2000 - 1
 QueryPerformanceCounter(nTime)
 SleepTimes(inx) = nTime
 Thread.Sleep(i)
 QueryPerformanceCounter(nTime)
 SleepTimes(inx) = (CDbl(nTime) /
 CDbl(nFrq)) _
 - (CDbl(SleepTimes(inx)) /               CDbl(nFrq)) _
 inx = inx + 1
Next
```

Delay Test

In multi-threaded programs, threads sometimes need to synchronise with each other. The various versions of the Windows provide a message passing mechanism to enable such synchronisations. The inter-thread message delay test tested the delay between the time asked for and the time measured. This test was repeated 2000 times.

```
For i = 0 To 2000 - 1
 nRand = Fix((1001) * Rnd())
 DTime.SetDelay(nRand)
 DTime.Run()
 QueryPerformanceCounter(nTime)
 m_TimeSem.waitone()
 QueryPerformanceCounter(Cnt)
 DelayTimes(i, 0) = nRand
 DelayTimes(i, 1) = (CDbl(Cnt) /          CDbl(nFrq)) _
 - (CDbl(nTime) / CDbl(nFrq))
 DTime = New TimeDelay
Next
```

Keyboard Test

For the keyboard test, the delay from issuing a sound to entering a key on the computer's keyboard was measured. The test used a hardware set up where the sound output from the computer automatically triggered the keyboard input.

Each test used the QueryPerformanceCounter function which provides the most accurate time measure available from Windows. These three software-based tests can inform about timing variability, but not about the timing delay. To obtain an objective measure of delay, we employed a cus-

Table 1. Basic Properties of the Three Computers

Computer	Processor	Cock f (GHz)	RAM (Mb)	Graphics Proc.	Graphics RAM (Mb)
HP Compaq 8510p	Intel Core 2 Due T8300	2.40	4,000	ATI Mobility Radeon HD2600	256
HP Compaq DX2000 MT	Celeron	2.53	1,000	Intel 8258G	??
HP Omnibook 6100	Intel Mobil III - M	1.00	500	ATI Mobility Radeon	16

tom-built hardware that triggers a keyboard response within 1 ms of an alternating voltage signal reaching above a 30 mV threshold. This device was connected to the computer's sound output and PS/2 keyboard input, and the following code was run:

```
For i = 0 To 99
 m_nStartTime = GetTime()
 m_BufferISIP.Play(0,               BufferPlayFlags.Default)
 Thread.Sleep(1000)
Next
Protected Sub SoundGen_KeyPress(ByVal          sender _
As Object, ByVal e As _
System.Windows.Forms.KeyPressEventArgs) _
Handles MyClass.KeyPress
 m_Times(m_nIdx) = GetTime() -              m_nStartTime
 m_nIdx = m_nIdx + 1
End Sub
```

We only tested the Compax DX 2000 MT desktop computer, because it was the only one to have a PS/2 keyboard connector. Given the large differences between the Oss obtained for the other tests, all three Oss were compared.

Data Analysis

The 2000 data points from each test were used "as-is" for obtaining descriptive statistics. Although few of the data sets exhibited a normal distribution, we provide standard deviations and variances to give an impression comparable to real data. A few singular extreme values were eliminated, according to the logic that they might have been noted and eliminated as outliers in behavioural data. In addition to this, the same data were arranged in bins with numbers typical for those used in behavioural research when e.g. groups of participants are compared. Hence, we computed means and variances for 40 bins containing 50 data each, considering each bin a random sample of the entire population of timing performance for the respective test. This procedure will tend to decrease overall variance for flattened non-normal distributions (variances of variances) and yield more conservative estimates. Likewise, the non-normality of many bins would yield outlier values that inflate variance. We nevertheless considered it appropriate to examine interactions on this basis, as the binning procedure would provide a more conservative test.

RESULTS

The main goals of the present study are to help researchers assess the feasibility of using general purpose computers for their research task, and to provide examples of how different hardware, OS versions, and types of timing procedures interact. Results are presented for each OS and computer combination separately, but arranged so as to facilitate comparisons among these. Finally, interactions among computer and OSs are examined when applicable.

Loop Test

Fig. (**1**) shows the means of bins for each machine and OS. Values measured for the loop test tended to return at approximately the same value for each run of the test but with some notable exceptions.

All three OSs appear to yield the same average loop time of 36 +/- 1.5 ms when running on the 8510p. All three OSs also exhibit the same minimum time of 35 ms, but differ from 47 to 114 ms in the maximum times measured.

More variation was observed with regards to the mean value for the DX 2000 machine when compared with the 8510p machine. The mean loop duration varied from about 110 to 125 ms, and maximum durations reached from 169 to 395 ms. The 6100 exhibited a mean loop duration intermediate to those of the other computers. A remarkable finding was that although the mean loop duration was large for all OSs when they ran on the DX 2000, the associated variability was one order of magnitude smaller under XP! The interactions between the two computers and the three OSs were highly significant, as shown by the ANOVAs reported in Table **2**.

Delay Test

The measured delay was mostly close to the required delay for the 8510p, but it was too short for the other com-

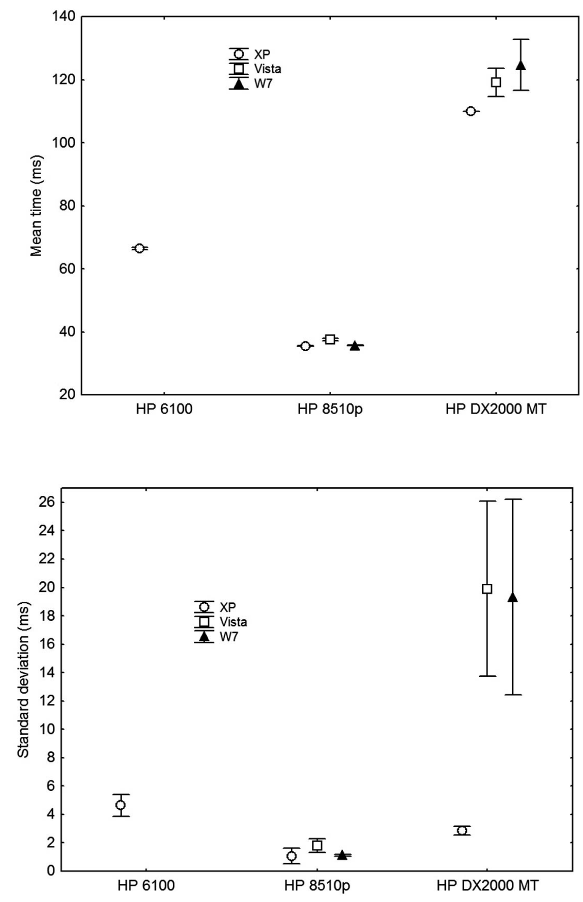
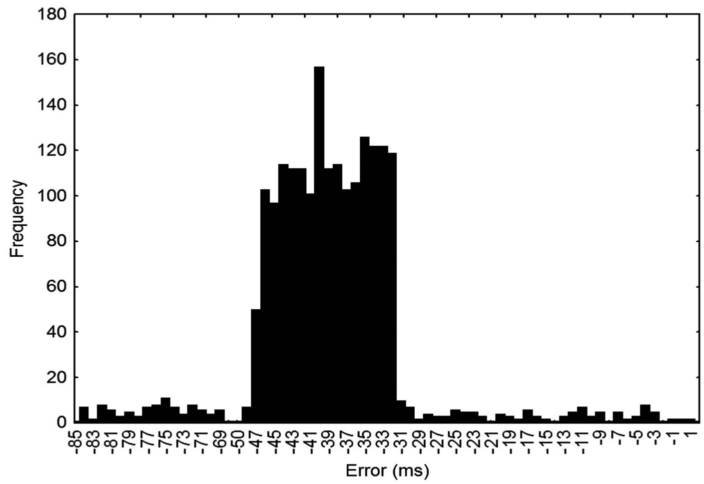


Fig. (1). Loop test performance as a function of computer and OS across bins. **A**. Mean means of loop times. **B**. Mean SDs of loop times.

Table 2. Descriptive Statistics for Loop Execution Times

		Min	Max	Median	IQR	M	SD	S	Skewness	Kurtosis
Compaq 8510p	XP	35	114	35	1.0	35.52	2.00	4.01	30.88	1187.76
	Vista	35	86	37	1.0	37.54	2.56	6.55	8.23	114.05
	W7	35	47	35	2.0	35.70	1.16	1.35	1.71	4.69
Compaq DX200 MT	XP	108	169	109	1.0	109.94	2.98	8.86	5.66	79.36
	Vista	108	686	109	0.0	119.18	30.59	935.98	7.68	92.76
	W7	108	395	109	2.0	124.62	37.89	1435.45	3.11	10.40
Omnibook 6100	XP	60	153	65	8.0	66.43	5.33	28.44	5.52	53.70

Note. For the loop time means, 2-way 3 OS x 2 computer ANOVAs with means as dependent variable showed significant effects of both OS (F2, 78 = 9.82, p < 0.0005), computer (F1, 39 = 2405.3, p < 0.0000001), and the OS x computer interaction (F2, 78 = 8.62, p < 0.0005). For the loop time SDs, the same kind of ANOVA showed significant effects of both OS (F2, 78 = 14.45, p < 0.00001), computer (F2Note. For the loop time means, 2-way 3 OS x 2 computer ANOVAs with means as dependent variable showed significant effects of both OS (F2, 78 = 9.82, p < 0.0005), computer (F1, 39 = 2405.3, p < 0.0000001), and the OS x computer interaction (F2, 78 = 8.62, p < 0.0005). For the loop time SDs, the same kind of ANOVA showed significant effects of both OS (F2, 78 = 14.45, p < 0.00001), computer (F2, 78 = 65.69, p < 0.0000001), and the OS x computer interaction (F2, 78 = 13.82, p < 0.000005)., 78 = 65.69, p < 0.0000001), and the OS x computer interaction (F2, 78 = 13.82, p < 0.000005).



Note that the abscissa displays extant values in order of magnitude rather than according to scale of magnitude. The largest value was 46 ms, which cannot be seen because only every second value is displayed.

Fig. (2). Distribution of delay errors found for HP DX 2000 MT and Vista.

puters. This was particularly pronounced for Vista and W7, returning on the order of 40 ms early. In addition, some delays exceeded substantially the requested delay. For example, a requested delay of 73 ms resulted in a measured delay of 215 ms for XP. The results are summarised in Table **3**, and an example of the distribution is shown in Fig. (**2**).

Finally, Fig. (**3**) shows the means of bins for both means and standard deviations for each machine and OS. These interactions were significant, as shown by the ANOVAs reported in Table **4**.

Sleep Function

Visual inspection of the target and actual intervals given by the sleep function indicated that the latter were quantised with respect to the former. Specifically, target intervals from 0-15 gave a sleep interval of 16, 16-32 gave an interval of 32 and so forth. These target-actual interval differences (in the following called errors) are exemplified in Table **5**. This phenomenon yielded an almost quadratic distribution of errors, exemplified for computer 8510 with Vista in Fig. (**4A**). The expected error was for the 8510p and DX 2000 computers not exactly the modulo of 16, but was best described by

The results also show a number of sleep durations that exceeded the typical amount for each block, and also some where the computer returned before requested. For example, a requested sleep delay of 422 ms resulted in a delay of 449 ms, one test on the DX 2000 machine using Windows XP, 24 ms off from the expected 422 ms for the block of values from 407 ms to 422 ms. A requested 76 ms sleep delay returned a value of 69 ms, 8 ms short of the expected value.

Even larger deviations were observed for Vista. A requested delay of 49 ms resulted in a measured delay of 153 ms, 105 ms off. For Windows 7, the measured sleep delay tended also to fall within a band of requested sleep delays. However, some large deviations were also observed. A request for 1685 ms produced a delay of 2154 ms, which is 467 ms off, and a request for 1693 ms produced a delay of 7059 ms, 5366 ms off. Some delays also returned early e.g. a requested delay of 1698 ms resulted in a delay of 1676 ms, 50 ms early.

Table 3. Descriptive Statistic of Keyboard Delays

		Min	Max	Median	IQR	M	SD	S	Skewness	Kurtosis
HP Compaq DX2000 MT	XP	66	81	76	4.0	75.62	3.73	13.89	-1.22	1.13
	Vista	75	363	85	4.0	87.36	28.14	791.67	9.68	95.76
	W7	66	81	76	4.0	75.62	3.73	13.89	-1.22	1.13

Differences Between means were significant according to one-way ANOVA ($F2, 297 = 16.82$, $p < 0.00001$)

Table 4. Descriptive Statistics of Delay Test Errors

		Min	Max	Median	IQR	M	SD	S	Skewness	Kurtosis
Hp Compaq 8510p	XP	-4	142	0	0.0	0.26	5.58	35.71	22.30	497.34
	Vista	-19	134	0	0.0	-0.11	3.29	10.80	33.29	1390.28
	W7	-7	303	0	0.0	0.20	7.29	53.15	38.10	1526.79
HP Compaq DX2000 MT	XP	-63	271	-23	0.0	-23.27	8.59	73.73	20.07	690.89
	Vista	-85	46	-39	8.0	-39.45	11.23	126.19	-0.74	7.14
	W7	-94	359	-39	8.0	-38.55	13.99	195.68	20.59	564.47
HP Omnibook 6100	XP	-45	234	-15	8.0	-15.01	6.75	45.52	24.87	929.62

Note. For the delay error means, 2-way 3 OS x 2 computer ANOVAs with means as dependent variable showed significant effects of both OS ($F2, 78 = 935.14$, $p < 0.000001$), computer ($F1, 39 = 34569.27$, $p < 0.000001$), and the OS x computer interaction ($F2, 78 = 1477.82$, $p < 0.0000001$). For the error SDs, the same kind of ANOVA showed a significant effects of computer ($F2, 78 = 173.81$, $p < 0.0000001$), and OS x computer interaction ($F2, 78 = 5.66$, $p < 0.01$), but not of OS ($F2, 78 = 2.085$, $p = 0.13$).

The 6100 did also exhibit temporal quantisation for the sleep function. Requests for 1–10 ms returned as a block of 10 ms, whereas on the DX 2000 and 8510p this block was approximately within the range of 16 ms. Thus, the 6100 computer with XP exhibited a modulo 10 function:

The sleep errors were adjusted by the values given by equations 1 and 2 to minimise variability. The rationale for this was that predictable errors are deterministic and could therefore in principle be eliminated or compensated for.

However, equations 1 and 2 could apparently not capture the underlying behaviour completely because of an unpredictable fluctuation in the position of the quantisation points. A small number of roll-overs therefore remain in the adjusted series, close to 10 ms for the 6100 computer and close to 16 ms for the other computers. Fig. (**4B**) shows the distribution after adjustment for computer 8510 with Vista.

$$e=15-mod16(round(1.024*target)) \quad \textbf{(1)}$$

Table **6** shows descriptive statistics for the non adjusted sleep errors, and Table **7** for the adjusted sleep errors. The differences in variability are small, which is explained by the fact that the original distribution was rectangular, and that the variance is mostly driven by large values in the tails of the distribution, which are only marginally affected by the adjustment.

Fig **5** shows the variability between OS and machines for the sleep test. Although all three OSs on each machine exhibited the same block behaviour variations do occur between OSs and between machines.

Keyboard Test

The keyboard test emitted a sound which triggered an emulated mouse button response. This test used a specially

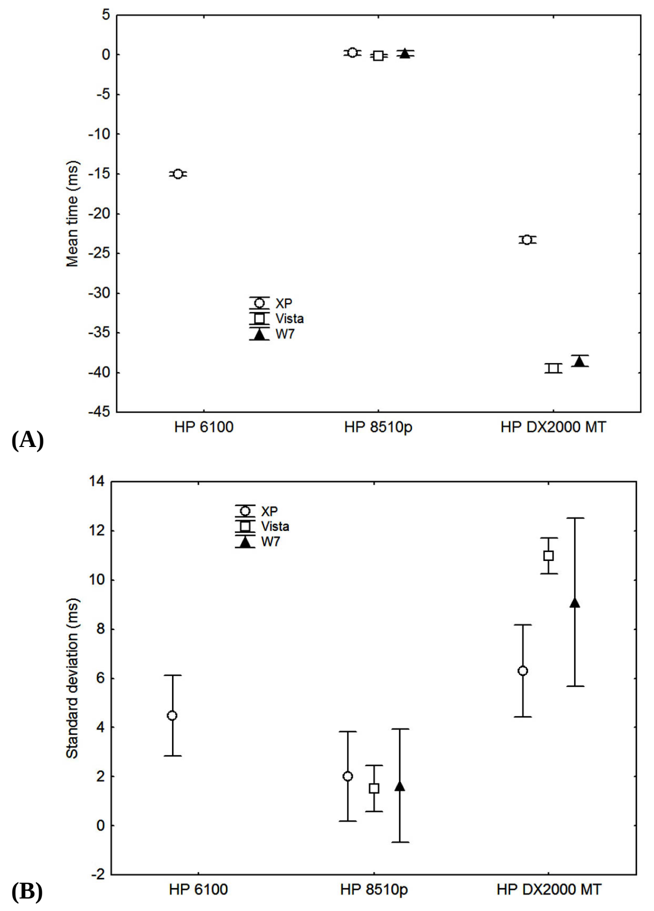


Fig. (3). Delay test performance as a function of computer and OS across bins. **A**. Mean means of delay errors. **B**. Mean SDs of delay errors.

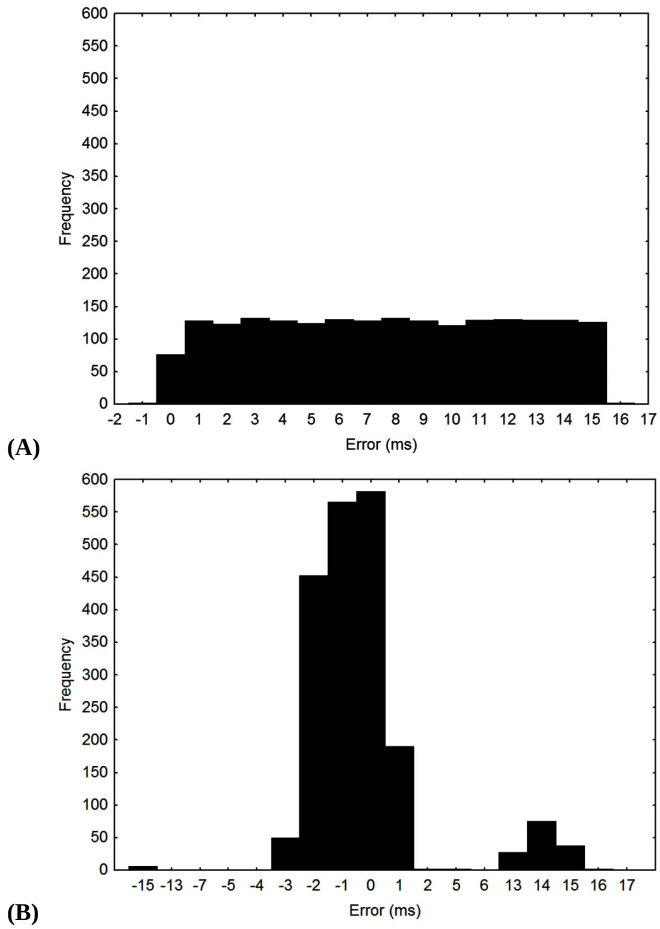


Fig. (4). Distribution of sleep errors found for HP 8510p and Vista. A. Original errors. B. Errors adjusted according to equations 1 and 2.

designed hardware with a response delay of less than 1 ms from the onset of the sound.

Spikes

On some occasions spikes were noticed on the first value measured. For example, using the 8510p with Windows XP Note that the abscissa displays extant values in order of magnitude rather than according to scale of magnitude. the first value measured was 114 ms on the loop test, where as the next values were 57 and 35 ms. The 114 ms was the highest value recorded for that test. On the DX 2000 using Windows XP a requested delay of 664 ms resulted in a measured delay of 935 ms, 271 ms off for the delay test. On the 6100, also on the delay test, a requested delay of 626 resulted in a measured delay of 860 ms, 234 ms off. However, not all tests had such a spike and other large values were observed at other points in the test.

DISCUSSION

Given that a PC with the Windows™ was not designed as a real time system it should come as no surprise that it does not have a fixed response time. The measured times exhibit considerable variability in the time it takes to run the same loop, for example. But this variability was on the same order of magnitude even for Windows functions intended for precise timing, and which do conceivably not exhaust the system's processing capacity.

Different computers can take different amounts of time to run the same code, as evident from the inter thread messaging delay between the DX 2000 and Compaq machines.

Most variations tend to lie within +/-30 ms of the required value. So, as long as accuracies of -30 to +50 ms lie within the acceptable range of experimental error, a PC running any general purpose Windows OS would return acceptable results. However, if we include the keyboard input the machines have an accuracy of within 100 ms so researchers should quote results within -30 to 100 ms.

However, the timing of the system exhibits large variability. Deviations go above and below the requested value (in the case of the sleep and delay functions), but are also characterised by few and large positive variations. As such variations occur less often than the more typical variation, an experiment using a large enough sample would effectively ignore larger spikes. How large a sample an experiment needs to average out large variations depends on the specific combination of machine and OS.

It should be noted that the sometimes very large variability within a machine and OS combination (e.g. HP Compaq DX2000 with Vista or W7) is indeterministic, and add to the error variability in the studies in which the data are used, which decreases the statistical power and the possibility to draw conclusions about differences between means.

In contrast, the differences between different computers and between different OSs are deterministic. This was indicated by the statistically significant differences even for relatively moderate sample sizes, observed in the present study. It should be noted that these differences, regardless of their significance from a computing perspective, are sometimes very large in relation to the group differences one normally observes between individual participants or groups of participants, or between other conditions in behavioural experiments. For example, the 6100 machine often returned before the requested time on the inter-thread communications delay but the 8510p tended to return about the right value. This raises concerns regarding studies where different machines and/or OSs might be employed at different times. If, for example, the same reaction time test was taken on the 6100 machine at one time, and the same participants returned to take the test one year later, but this time on an 8510p machine, this would falsely indicate a substantial increase in reaction time. This difference has a large probability of appearing to be statistically significant, because the deterministic nature of the measurement error provides small variability within each bout of measurements.

Experiments may therefore end up with significantly distorted results due to unexpected large variations resulting in the behaviour of the OS and/or machine. On the basis of this variation we can question results obtained from any use of a general purpose computer/OS combination. Some software providers argue that they can reach considerably smaller timing variability than we have reported here by special programming that presumably by-passes native OS routines. This remains to be proven, since to the best of our knowledge such claims have never been documented. Pending such empirical tests for realistic research applications, we must conclude that time-critical behavioural experimentation

Table 5. Typical Values for the Sleep Function Test for Windows 7 on the 8510p e=9-mod10(round(0.998859*target) (2)

Required (ms)	Measured (ms)	Difference (ms)
0	0	0
1	5	4
2	16	14
3	16	13
4	16	12
5	16	11
6	16	10
7	16	9
8	16	8
9	16	7
10	16	6
11	16	5
12	16	4
13	16	3
14	16	2
15	16	1
16	31	15
17	31	14
18	31	13
19	31	12

Table 6. Descriptive Statistics of Sleep Test Errors

		Min	Max	Median	IQR	M	SD	S	Skewness	Kurtosis
Hp Compaq 8510p	XP	0	16	8	8.0	7.70	4.50	20.29	-0.01	1.20
	Vista	-5	57	8	8.0	7.71	4.66	21.70	0.56	
	W7	-14	50	8	7.8	4.93	24.23	0.76	4.77	
HP Compaq DX2000 MT	XP	-8	24	8	8.0	7.76	4.83	23.31	-0.05	-0.69
	Vista	-8	105	8	8.0	8.85	7.98	63.74	3.42	23.39
	W7	-20	5366	8	8.0	8.57	13.09	171.37	23.87	783.13*
HP Omnibook 6100	XP	-6	18	5	6.0	5.00	3.05	9.33	0.00	-0.60

*All statistics except max computed without the extreme value 5366, in which case max was 456 ms

should not be controlled directly by a general purpose computer/OS combination. A real-time operating system on a dedicated computer is required, or some kind of external hardware in which the time-critical operations are programmed to run independently of the processor and the OS of any possible controlling host computer. At any rate, researchers must measure and report the actual delay and variability in their applications, rather than take for granted that they do not exist or are so small as to be inconsequential, as is the present custom. As our results have demonstrated, such ignorance can lead to fundamentally incorrect conclusions about the behavior in question.

SUMMARY

As general purpose computers find use within psychology, tests were conducted to ascertain the reliability of such machines for real time experiments. Three computers were used with three different versions of Windows; Windows XP, Vista and Windows 7. A set of tests were used to evalu-

Table 7. Descriptive Statistics of Sleep Test Errors Adjusted According to Equations 1 and 2.

		Min	Max	Median	IQR	M	SD	S	Skewness	Kurtosis
Hp Compaq 8510p	XP	-15	15	1	6.0	0.22	2.81	7.88	-5.12	24.89
	Vista	15	47	-1	9.0	0.27	4.20	17.65	3.05	13.86
	W7	-16	50	-1	9.0	0.35	4.54	20.63	3.28	17.95
HP Compaq DX2000 MT	XP	-15	11	1	9.0	0.32	3.25	10.54	-3.16	14.29
	Vista	-19	91	1	9.0	1.40	7.17	51.41	4.41	35.25
	W7	-26	5353	1	10.0	1.12	13.76	189.21	20.27	621.73*
HP Omnibook 6100	XP	-9	13	0	10.0	0.50	1.66	2.75	3.77	23.96

*All statistics except max computed without the extreme value 5353, in which case max was 460 ms

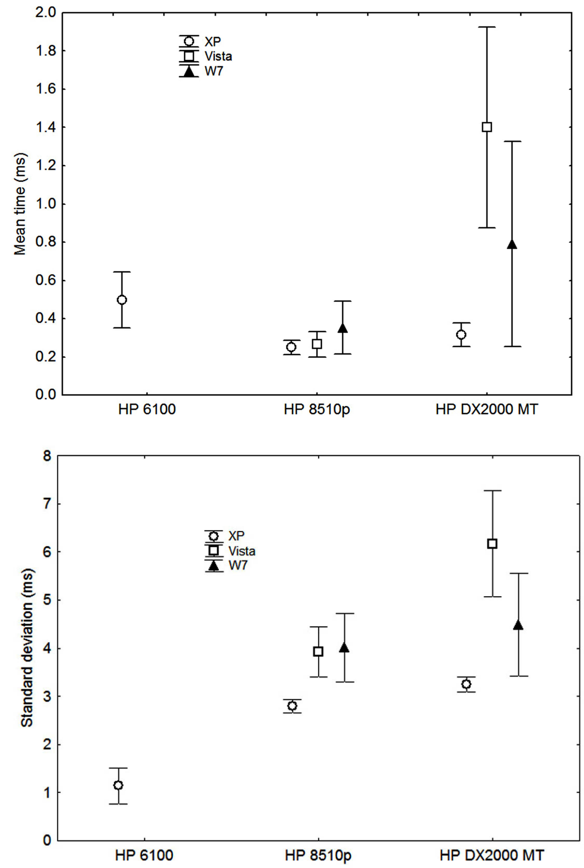


Fig. (5). A. Adjusted sleep test performance as a function of computer and OS across bins. **A**. Mean means of sleep errors. **B**. Mean SDs of sleep errors across bins.

ate the performance of running loops, sleep and inter-process messaging as well as the delays in keyboard response. The results showed that the response time can vary when running the same code on the same machine and between machines. However, most of the fluctuations tend to occur within +/- 30 ms, indicating that measurements recorded from a general purpose PC running Windows should have an accuracy of - 30 to +50 ms, which should be increased to 100 ms if the keyboard is used as in input device. The results also showed that occasionally larger variations occurred, which means that small samples might lead to distorted results and that experiments would need a large test based to ensure the results were not significantly distorted by a few extreme values.

CONCLUSION

We must conclude that time-critical behavioural experimentation should not be controlled directly by a general purpose computer/OS combination. Else, when conducting experimentation using a general purpose computer/OS combination it is important to know the limitation of the machine. Results could vary from machine to machine and between OSs running on the same machine. Experimentation results should have an accuracy of at least +/- 30 ms but 100 ms would represent a safer measure of the accuracy.

When using general purpose PCs, the experiments need to use large sample to avoid the possibility of distortions due to occasional large delays. How large a sample base an experiment needs depends on the OS and machine used. Prior testing or calibration should help to determine the test base needed.

If an experiment needs accuracy greater than +/- 50 ms then the experiment should use specialist real-time hardware and not rely on a general purpose computer. Researchers conducting longitudinal studies, or who otherwise want to compare data across sessions/studies should take care to use the same hardware and OS for all measurements.

FUTURE WORK

The experiments presented used the Visual Basic .NET programming language. Future work could include languages such as C. C++ or Java. The experiments only used one machine from each type of computer make. It could produce variable results if a number of machines were used of the same type. Other possibilities that could form part of future work could involve tests such as mathematical operations and screen operations or the use of real-time OSs such as eCos and Windows CE.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

None declared.

REFERENCES

[1] Jing L, Xudong W. Evaluations of the effects of relaxing music on the recovery from aerobic exercise-induced fatigue. Sports Med Phys Fitness 2008; 48(1): 102-6.

[2] Akrami N, Ekeammar B, Araya T. Category and stereotype activation revisited. Scand J Psychol 2006; 47: 513-22.

[3] Troche S, Houlihan ME, Stelmack RM. Mental ability, p300, and mismatch negativity: analysis of frequency and duration discrimination. Intelligence 2000; 37(4): 365-73.

[4] Madison G, Forsman L, Blom Ö. Correlations between general intelligence and components of serial timing variability. Intelligence 2009; 37: 68-75.

[5] Droit-Volet S, Izanute M. Improving time discrimination in children and adults in a temporal bisection task: The effects of feedback and no forced choice on decisions and memory processes. Q J Exp Psychol 2009; 62(6): 1173-88.

[6] Madison G. Variability in isochronous tapping: higher-order dependencies as a function of inter tap interval. J Exp Psychol Hum Percept Perform 2001; 27; 411-22.

[7] Madison G. Detection of linear temporal drift in sound sequences: principles and empirical evaluation. Acta Psychol 2004; 117(1): 95-118.

[8] Gabrielsson J, Jarvella RJ. How to obtain near millisecond precision on the ibm pc with visual stimuli not exceeding one printed line. Behav Res Methods Instrum Comput 1990; 22: 520-5.

[9] Mates J. A system control program on pc for tapping experiments. Comput Meth Prog Biomed 1990; 33: 43-8.

[10] Baril A. Using windows nt in real-time systems. Proceedings of the fifth IEEE real-time technology and applications symposium 1990; 132-41.

[11] Milanovic A, Srbjic S, Radej J. Performance of distributed systems based on ethernet and personal computers. Proc IEEE Int Symp Indust Electr 1999.

[12] Schmidt DC, Despade M, O'Ryan C. Operating system performance in support of real-time middleware. 7Th IEEE International Workshop on Object Orientated Real-Time Dependable Systems 2002; pp.199-206.

Received: July 26, 2012 Revised: October 01, 2012 Accepted: October 01, 2012