The Accuracy of Response Timing by Authorware Programs

Mark D. Tew and Kenneth O. McGraw

University of Mississippi

Correspondence Address:
Kenneth O. McGraw
Dept. Of Psychology
University of Mississippi
University, MS 39677
pymcgraw@olemiss.edu (e-mail)
662-915-5192 (voice)
662-915-5398 (fax)

Abstract

A programming option for researchers wishing to develop web-delivered research experiments is Macromedia's Authorware. As a first step in measuring the accuracy with which Authorware running under Windows can measure event times, keystroke pairs at precisely timed intervals of 150, 200, 250, and 1000 ms were sent via an external device to computers that used an Authorware program to detect the keystrokes and measure the interval between them. Measurement accuracy was compared among experimental conditions defined by the time interval, the mode of execution (in Netscape, in Internet Explorer, or as an executable file), and the CPU load. Results showed that accuracies were stable for typical CPU loads, but declined significantly in one extreme condition. Also, accuracies declined with time interval. Mode of execution was not a factor of importance. A comparison of an executable Authorware program with E-Prime, an experiment generator that is the gold standard for measurement precision, showed that Authorware's accuracy was comparable to E-Prime's except in the heavy CPU load condition.

The Accuracy of Response Timing by Authorware Programs

A technical challenge facing cognitive scientists wishing to use computers as laboratory instruments has been finding ways to achieve sufficient accuracy for computer-based measurements of event times. These problems have become less pronounced with the development of ever faster processors, video cards, and input devices; but with the faster computers have come multitasking operating systems, which introduce a novel challenge. The challenge results from the fact that the operations of Windows or any multitasking operating system are not under the control of the programmer. This results in unwanted intrusions into the performance of programs. The precision of event measurements suffers to the extent that the operating system intrudes during the measurements.

One solution to the intrusion problem has been to run programs in DOS, where programmers have more control (Myors, 1999). Another has been to build external devices (McKinney, MacCormac, & Welsh-Bohmer, 1999). A third has been to use software specifically designed to overcome Windows intrusions. The first of this type was MEL (Schneider, 1988) which has been superceded by E-Prime (Psychology Software Tools, n.d.). For most researchers, the third alternative is the most attractive because it allows researchers to take full advantage of the many useful features of Windows while not compromising event measurements.

In addition to dedicated experiment generator software such as MEL and E-Prime, there are off-the-shelf application programs that are useful for developing interactive experiments that run on Windows computers. One is Macromedia's Authorware (Macromedia, n.d.) . The advantage of off-the-shelf packages over speciality software packages are price, training support, and in the case of Authorware, web-deliverability. Unlike MEL and E-Prime, Authorware programs can be executed in a browser (either Internet Explorer or Netscape) using a browser plug-in that is equivalent to a run-time player. For researchers wanting to conduct Web-based experiments, therefore, Authorware offers an appropriate development tool. The question is whether Authorware is resistant to operating system intrusion errors. If not, it could not be used for developing experiments with time-critical measurements.

A strong suggestion that Authorware is reasonably resistant to intrusion errors comes from data gathered using Authorware experiments at PsychExperiments, an on-line psychology laboratory (http://psychexps.olemiss.edu). Many of the experiments at the site employ event times as either the independent or dependent variable. Although the accuracy of these measurements has not been determined directly, data from the experiments indicate accuracy sufficient to replicate time-dependent phenomena such as the Stroop effect, the gender difference in mental rotation, both semantic and attentional priming, and a time-dependent line motion illusion (McGraw, Tew, & Williams, 2000).

The purpose of this report is to take a first step at directly assessing Authorware's accuracy in measuring event times. Rather than look at all relevant event times--displays, reaction times, and time intervals--we restricted our observations to the last of these, time intervals, by assessing Authorware's accuracy in measuring intervals between two keypresses when the presses were separated by times ranging from 150 to 1000 milliseconds (ms). The measurements were made in a variety of experimental conditions defined by mode of execution (in a browser or as an executable file) and CPU load (the number and nature of simultaneous tasks executed by the Windows or NT operating system). The measurements included conditions in which Authorware's accuracy for measuring timed intervals could be compared to the accuracy of E-Prime.

Method

Machines

Five PC computers running Windows served as "subjects" in the core design used to assess Authorware's accuracy in measuring keystroke intervals. These computers were sampled from publicly available computers located on the University of Mississippi campus. Treating these machines as subjects, a set of repeated measures were obtained on each according to a 3 x 4 x 4 within subjects experimental design. Six other machines were also used to make some measurements but the full experimental design was not implemented on these machines. The five primary computers used in this research are numbered 1 to 5. Their operating system, processor, and random access memory capacity are given in Table 1.

Design

The first factor in the core 3 x 4 x 4 design was execution mode with levels labeled Executable (the Authorware program packaged as an *.exe file and run under Windows), Netscape (the Authorware program packaged to run in a browser and run in Netscape using Macromedia's Netscape web player plug in), and IE (the same web-packaged files executed in Internet Explorer using Macromedia's Internet Explorer web player plug in). The plugins are available at http://macromedia.com .

The second factor in the design was operating system load (OS Load). In the minimum load condition, labeled "Dedicated" no programs were open on the computer other than the measurement program, the operating system, and–where required–a browser. The Dedicated results show the capabilities of Authorware programs under preferred conditions, which is to say the condition defined by a computer dedicated solely to the experimental task. The second level of load was labeled “Default” because the additional programs open while running the Authorware program were those initiated at start up by Windows. The actual programs varied across machines but included an antivirus program in every case. The Default OS Load was used to simulate the conditions that would typify the conditions for machines that might be used for data collection in a campus computer laboratory. The third level of OS Load was labeled “WebTraffic” because the computer received a RealAudio stream during the time measurements were made. The WebTraffic results give an indication of the effect of background internet traffic on measurement accuracy during otherwise dedicated conditions. The fourth level of load was the most extreme. Labeled “Time Killer,” this condition consisted of a second Authorware program packaged as an *.exe file that continuously incremented the value of a variable, compared the new value to 10,000, and reset the variable to zero when it reached 10,000. This Authorware program ran minimized while the measurement program was executing. The Time Killer results give an indication of the effect of clock-cycle-intensive programs running simultaneously with the measurement program.

The third factor in the core design was the length of the target interval. The intervals were 150 ms, 200 ms, 250 ms, and 1000 ms. These values were chosen as representative of response times that might be obtained in choice reaction time experiments.

The dependent measure in this experiment was the absolute deviation between the known target value for each of 50 keystroke intervals and the Authorware measurement of the interval.

Apparatus

A laptop computer (Winbook) and a stand-alone data acquisition system (National Instruments DaqPad 1200 communicating with the laptop through the laptop's parallel port) were used to create a keystroke generator. This combination creates a programmable system capable of providing keystroke signals at precisely timed intervals.

A keyboard communicates to the Windows operating system through two lines called the Clock line and the Data line. Communication is bidirectional in that either the computer or the keyboard can begin communication and control the clock and data lines to transmit data. Each keystroke signal consists of a sequence of 11 bits: one start bit, 8 bits denoting the key, one parity bit (odd parity), and a stop bit. In the quiescent state, both the clock line and the data line "float" high. To communicate to the computer that a key has been pressed, the keyboard (and the keystroke generator) commences a communication by pulling the data line low and then the clock line low—this is the start bit. During communication of the data bits, parity bit, and stop bit, the clock line is used to signal the presence of new data on the data line. One may view the clock line as a negative edge trigger signal that denotes the presence of new data on the Data line. Each new data bit must be provided within 10 microseconds of the rising edge of the clock line. Time to transmit the complete 11-bit keystroke signal is about 1 millisecond. Each keystroke consists of two separate communications: the bit sequence just described, which denotes a key-down condition, and the same bit sequence followed by a key-up signal when the key is released (the key-up code is 00000111111). According to the specifications, the computer can override a keyboard generated communication by holding the clock line high, and the keyboard should check for that state prior to communicating each bit. In this experiment we implemented only a one-way communication from the keystroke generator to the computer and did not check for a computer override signal. This does not affect the results reported here, because if the condition described were to occur it would result in a test that was not completed; that is, fewer than 50 keystroke pairs would be transmitted.

In order to generate the keystroke signals, the DaqPad's digital waveform generation functions were employed. In this mode, 8-bit data is transferred from the computer to the DaqPad and placed on one of the DaqPad's 8-bit TTL ports. The DaqPad communicates the data to the outside world asynchronously using the handshaking signals OBF (Output Buffer Full) and ACK (Acknowledge). Every time new data is placed on the port the OBF line is pulled low, signaling the presence of new data. A receiving device pulls ACK low after reading the data, and the OBF is automatically reset high by the negative edge on ACK. In order to achieve the maximum speed for these experiments, the OBF and ACK handshaking lines were tied together. With this configuration, the OBF going low provides the negative edge on ACK which then sets both OBF and ACK high until the next new data is output. With OBF and ACK tied together the DaqPad is capable of providing new data at slightly longer than 13 microsecond intervals, which this paper will call a "DaqPad cycle". Two of the eight lines of the DaqPad's output port are wired into a standard PS2 keyboard connector to the Clock and Data Lines , and the keystroke generator replaces the keyboard for the computers tested, as shown in Figure 1.

A Visual Basic program (Figure 2) running on the laptop computer accepts input from the person conducting the measurements and calls driver functions to control the DaqPad. An experimenter can set two parameters, the interval between the onset of key-down signals (in milliseconds) and the number of keystroke pairs to be generated. The measurements reported here used a keypress duration of about 53 milliseconds (from the start bit of key down to the start bit of key up), a duration typical of keystrokes we observed using a Tektronix 1200B Logic Analyzer. The entire keypress cycle (key-down followed by key-up) consists of the key signal (~1ms), followed by a 53 ms delay (4000 DaqPad cycles), then another key signal (~1ms), about a 4 ms delay (320 DaqPad cycles), and finally the key-up signal (~1ms). While a key signal by itself represents a key-down event, the combination of a key signal followed shortly (< 5 ms) by a key-up signal represents the key being released. Keystroke interval choices provided are 150 milliseconds (6750 DaqPad cycles), 200 milliseconds (10525 DaqPad cycles), 250 milliseconds (14325 DaqPad cycles), and 1000 milliseconds (71000 DaqPad cycles). The conversion between milliseconds and DaqPad cycles was empirically determined by observing the output signal with the logic analyzer, and adjusting the cycle count until the desired interval was obtained. During this calibration, the logic analyzer was running in asynchronous mode using its internal clock with a sample rate of 100 microseconds for the 150 ms, 200 microseconds for the 200 ms and 250 ms intervals, and a sample rate of 1 millisecond for the 1000 ms mode. The sampling interval of the logic analyzer sets an upper bound on the error in presentation of keystroke intervals.

After the experimenter sets the two parameters, the Visual Basic program controls the DaqPad as follows. Eight bit data words that generate (1) the 11-bit key down signal, (2) the 53 ms keypress duration, (3) the 11-bit key code plus the key-up signal, (4) enough data words to account for the keystroke interval, (5) the 11-bit key signal, (6) the 53 ms keypress duration, and (7) the 11-bit key code plus the key-up signal are downloaded to the DaqPad's memory. When the experimenter clicks on start, the DaqPad is directed to send the precisely timed keystroke pair. Then an internal Visual Basic function provides a random delay of between 0.5 seconds and 1 second, before sending the next pair of keystrokes. The keystroke pair/random delay sequence is repeated until the desired number of signals of keystroke pairs have been generated. We believe that this keystroke signal best emulates the kind of measurements of intervals between keystrokes that would be obtained in actual experiments. The random delay between keystroke pairs was included to preclude the possibility that the keystroke generator and the computer being measured would chance upon some particularly advantageous periodicities.

Other Custom Software

The Authorware program used to detect the keyboard signal (a number “2") consisted of a decision icon set to repeat 50 times. A map icon hanging off the decision icon contained sequential keypress interaction icons. The first was set to watch for a “2" and to start the system clock when it was detected. Authorware’s system clock has a 1 ms resolution. The second interaction watched for a second “2" and on detection computed the difference between the current clock reading and the starting clock reading. Courtesy of Walter Schneider, staff at Psychology Software Tools provided us with an E-Prime program (*.ebs) to perform the same operations as the Authorware program. The program was executed using E-Prime’s runtime player, E-Run, v 1.0 (Beta 5.0).

Procedure

The laptop computer and DaqPad make an easily transportable system which was carried to the computers reported in Table 1. Each computer was equipped with a browser and had a network connection to the Internet. The experimenter loaded the Authorware Web Players for Netscape and Internet Explorer on the computer, the E-Prime runtime player E-Run, the

Authorware and E-Prime measurement programs, and the Authorware Time-Killer program onto the computer's hard disk. The keyboard connection to the computer under test was replaced by the keystroke generator apparatus described above.

Each measurement took place as follows. The measurement program on the computer being tested was begun, the keystroke generator was set for a 150 ms interval and 50 pairs of keystrokes were sent and measured by the computer under test, and then the measurement program paused. The keystroke generator was then configured for a 200 ms interval and 50 pairs of keystrokes were sent and measured, and the measurement program again paused. This sequence was repeated for the 250 ms and 1000 ms intervals.

Results

To conduct the data analysis, we converted the observed interval measurements (50 per experimental condition) into deviation scores, which were computed by subtracting the true interval length from the observed length. These deviations represent errors. Positive values for error represent intervals measured as longer than they in fact were and negative values represent measurements shorter than they in fact were. We next defined on-target values as values that deviated no more than 1 ms from their targets. Table 2 gives the percentage of on-target values per experimental condition. A three-way repeated measures analysis of variance showed that OS Load was the only significant source of variability in these percentages when alpha was set at .05. Overall on target percents were 89% and 91% for the Dedicated and Default load conditions, 85% for the Web Traffic condition, and 38% for the Time Killer condition. Obviously, Time Killer was a condition that degraded Authorware's measurement accuracy. The Time Killer mean (38%) differed significantly from all others with no significant differences among the other three means. Expressed in terms of effect size (sample η^2), OS Load accounted for 58% of the total variance and the contrast between the TimeKiller mean and the other three combined accounted for 99% of the OS Load variance.

Although interval was not a significant source of variance in the above analysis (p=.058), mean percent on target did decline systematically with the length of the interval to be measured. Looking just at the three conditions in which good accuracy was achieved (i.e., the Dedicated, Default, and Web Traffic conditions), the mean percent on target were 92% , 91% , 91% and 81% for the 150, 200, 250, and 1000 msec intervals. The decline in accuracy with interval length is consistent with the fact that longer intervals are more likely to have included operating system activities that would have caused measurement errors.

Given that the percent of Authorware's on-target measurements were quite good on average in three of the four load conditions leaves open the question of how bad the off-target measurements were. The box plots in Figure 3 address this question and provide data on machine-to-machine variability in measurement accuracy. The machine-to-machine variability went unanalyzed in the initial analysis which treated machines as "subjects." The data chosen for display in Figure 3 are for the Executable mode. Box plots for the other modes–IE and Netscape–would look essentially the same. (Recall that Mode was not identified as a significant source of variance in the three-way anova.).

The boxes in Figure 3 are drawn using the 25^{th}, 50^{th}, and 75^{th} percentile values in the distributions of 50 measurements taken in each condition. For many of the conditions, these values are the same, which accounts for the fact that the "boxes" are single, superimposed lines.

Extending from the boxes are whiskers that are drawn to the largest and smallest errors that meet a statistical definition of non-extreme (i.e., less than 1.5 times the interquartile range above or below the box). For most conditions there are no whiskers because there were no off-target values that met the statistical definition. Circles mark individual measurements that do meet the statistical definition for outliers (greater than 1.5 times the interquartile range above or below the box.

One obvious effect in the Figure 3 data is the OS Load effect described above. The Time Killer condition stands out as the one that produced the largest and most frequent errors. A second visually apparent effect is that more outliers are observed at longer intervals than at shorter ones. This effect was not significant in the overall analysis but, nonetheless, appears reliable, as noted above. The one novel effect in these data is an apparent machine effect. In particular, Machine 5 made very accurate measurements even in the Time Killer condition. The fact that this machine had the fastest processor of any we tested (930 MHz) suggests that the faster processor speed accounts for the more accurate results.

To put the on-target data for Authorware into perspective, we compared Authorware measurements to ones obtained using E-Prime. This comparison could only be conducted on two machines (Machines 4 and 5) because the player file for E-Prime (E-Run) did not work on all the machines we used. The results of the comparison are given in Figure 4, again using box plots. For Machine 5, which was the 930 MHz Pentium machine with 128K of RAM, there was no difference between Authorware and E-Prime in the accuracy with which intervals were measured. On Machine 4, a 450 MHz machine with 128 Mb of RAM, there was a difference, but only in the Time Killer condition. Aside from that, differences are not dramatic, as shown in

Table 3 where on-target percents are given for Authorware and E-Prime as a function of OS Load and length of interval.

The present study was not designed to assess time as variable. To do so would require taking repeated measures on single machines under otherwise constant conditions. In the course of our observations, however, we did make some isolated measurements in which time of day was the only factor and we did observe strong hints of substantial cross-time variability in accuracy of measurement. Table 4, for example, gives on-target data for Authorware measurements made on Machines 1 and 2 on two separate measurement occasions when running in Dedicated mode. Table 5 gives on-target data for E-Prime running on Machine 1 in Default mode on three separate occasions. Of particular note in these data is the variability in on-target values for Machine 2. On the second measurement occasion, on-target values were all above 90% whereas on the first occasion only a third of the measurements achieved this level of accuracy with one on-target value being just 18%. This raises the issue whether the machine variability observed in the study, albeit limited, is substantially due to measurement time.

Discussion

The present study gives quantitative indications of the accuracy of measurement that can be achieved using Authorware programs in three modes of use–running in Netscape with the Netscape web player, running in Internet Explorer with the IE web player, and running as an executable file. There was no indication that execution mode was a variable of any importance; therefore, the accuracy findings generalize across all three modes.

We also assessed the accuracy for time intervals of different length. In general, accuracy was greater for shorter than for longer intervals, an observation which is consistent with the fact that inaccuracies are introduced when Windows gives priority to operations that steal clock cycles from the operations called for by the application program. The likelihood that a longer interval will include stolen clock cycles is greater than a shorter one. However, even though we believe interval length is a valid source of variance in measurement accuracy, it is not a large one. Using the on-target data that have been the primary focus of our analysis, time interval accounted for just 2.2 % of the total sample variance.

The single largest source of variance in our measurements was operating system load. This variance was almost totally due, however, to the one condition–Time Killer-- that represents a load condition that is least likely to occur in Web-based data collection by home or lab computer users. Somewhat more likely are conditions simulated by Web Traffic in that it is conceivable that research participants would be receiving e-mail, getting anti-virus updates, and perhaps even listening to streamed music, while participating in a research experiment. Our results indicate that such activities do not seriously degrade the ability of Authorware programs to obtain measurements that are within 1 ms of their true target values. Ideally, of course, one would want research participants to shut down unneeded application programs when collecting time-sensitive data in an experiment, but the results here show that there is no meaningful difference between Default and Dedicated conditions. Therefore, it may be unnecessary to have users close all unwanted programs before starting an experiment. Default conditions which typically include an anti-virus program, e-mail program, and a firewall do not degrade performance to any measurable degree.

An important point yet to be made is that the accuracy of measurements reported here are for single measurements. Thus a report of 90% on target measurements indicates that 45 of the

50 measurements we made in an experimental condition were within 1 ms of their target value. Researchers rarely use single measurements. They use averages. If we re-address the issue of accuracy by asking what percent of the 50 measurement averages were on target and if we drop the Time Killer condition which clearly produced unacceptable Authorware measurements, we find that the means for all 36 sets of measurements made at 150, 200, and 250 ms in the core design were on-target and 29 of 36 means were on-target at the 1000 ms interval. Off-target means ranged from 2 to 5 ms off target. The point of this observation is that when the data from studies are means, the inaccuracies found in a few aberrant values will be masked by their being averaged with predominately accurate values.

The final observation of note is the comparison of Authorware's accuracy in measuring keypress intervals to the accuracy of E-Prime. In all but the Time Killer condition, they are essentially equivalent. E-Prime has been developed with particular attention to timing issues, as is evident in the Time Killer data on Machine 4, which show that E-Prime was able to command processing time more effectively than Authorware. However, in the Default, Dedicated, and Web Traffic conditions, which represent the user conditions most likely to be sampled in Web-based research, the E-Prime advantage in accuracy was quite small.

References

Macromedia (n.d.). Macromedia Authorware 6.5: Product overview. Available online at http://www.macromedia.com/software/authorware/productinfo/product_overview/ .

McGraw, K.O., Tew, M.D., & Williams, J.E. (2000). The integrity of web-delivered experiments: Can you trust the data? *Psychological Science*, *11*, 502-506.

McKinney, C.J, MacCormack, E.R., & Welsh-Bohmer, K.A. (1993). Hardware and software for tachistoscopy: How to make accurate measurements on any PC utilizing the Microsoft Windows operating system. *Behavior Research Methods, Instruments, & Computers, 31,* 129-136

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

Psychology Software Tools (n.d.). *E-Prime: User's guide beta 5 (rev. 5.0)*. Available for download from http://www.pstnet.com/E-Prime/beta/e-beta5.htm.

Schneider, W. (1988). *Micro Experimental Laboratory Language Reference Manual.* Pittsburgh, PA: Psychology Software Tools.

Table 1

Description of the Computers Used to Make Measurements

Machine	Operating System	Processor (Mhz)	RAM (Mb)
1	Win 95	166	32
2	Win 98	450	64
3	Win 98	450	128
4	Win 98	450	128
5	Win 2000	930	128

Table 2

Proportion of On-Target Measurements Broken Down by Mode, OS Load, and Target Interval Length for Each of 5 Machines

			Machine				
Mode	OS Load	Interval (ms)	1	2	3	4	5
Netscape	Dedicated	150	1.00	0.88	0.92	0.96	0.96
		200	1.00	0.64	0.92	0.98	0.92
		250	0.96	0.90	0.88	0.94	0.96
		1000	0.90	0.50	0.78	0.98	0.94
	Default	150	0.98	0.96	0.84	0.93	0.74
		200	1.00	0.96	0.80	0.92	0.88
		250	0.96	0.94	0.88	0.93	0.80
		1000	0.94	0.92	0.92	0.93	0.94
	T-K	150	0.08	0.28	0.26	0.30	0.98
		200	0.44	0.16	0.30	0.12	0.96
		250	0.32	0.20	0.28	0.10	0.98
		1000	0.36	0.14	0.36	0.18	0.66
	W-T	150	0.96	0.78	0.92	0.88	0.88
		200	0.96	0.66	0.92	0.90	0.96
		250	0.94	0.74	0.88	0.86	1.00
		1000	1.00	0.72	0.78	0.76	0.72
I-E	Dedicated	150	1.00	0.88	0.86	0.90	0.98
		200	1.00	0.96	0.90	0.96	0.94
		250	1.00	0.86	0.82	0.96	0.98
		1000	0.92	0.92	0.94	0.92	0.66
	Default	150	1.00	0.88	1.00	0.96	0.94
		200	1.00	0.96	0.98	0.98	0.88
		250	0.96	0.86	0.92	0.92	0.94
		1000	0.94	0.92	1.00	0.86	0.96
	T-K	150	0.22	0.22	0.28	0.28	1.00
		200	0.42	0.10	0.36	0.08	0.96
		250	0.50	0.26	0.34	0.14	0.94
		1000	0.16	0.12	0.32	0.10	0.62

Mode	OS Load	Interval (ms)	Machine 1	2	3	4	5
	W-T	150	0.86	0.76	0.90	0.98	0.98
		200	0.92	0.46	0.88	0.94	0.92
		250	0.98	0.76	0.86	0.88	0.90
		1000	0.88	0.72	0.84	0.96	0.70
Exe	Dedicated	150	0.94	0.84	0.92	0.86	0.96
		200	0.94	0.84	0.98	1.00	0.94
		250	0.98	0.86	0.84	0.90	0.98
		1000	0.98	0.18	0.84	0.92	0.62
	Default	150	1.00	0.94	1.00	0.96	0.98
		200	1.00	0.78	0.96	0.96	0.96
		250	0.94	0.90	0.96	0.92	0.88
		1000	0.90	0.02	0.98	0.90	0.50
	T-K	150	0.46	0.22	0.34	0.28	1.00
		200	0.44	0.10	0.34	0.14	0.98
		250	0.34	0.26	0.36	0.08	0.96
		1000	0.50	0.12	0.40	0.20	0.60
	W-T	150	0.84	0.76	0.92	0.98	0.90
		200	0.94	0.74	0.92	0.96	0.98
		250	0.94	0.70	0.80	0.96	0.94
		1000	0.86	0.18	0.86	0.92	0.86

Table 3

Proportion of On-Target Measurments Obtained in Executable Mode on Machines 4 and 5 Using Authorware and E-Prime to Capture Interval Lengths

		OS-Load			
Application	Interval(ms)	Dedicated	Default	T-K	W-T
		Machine 4			
Authorware	150	.91	.96	.29	.92
	200	.98	.97	.11	.93
	250	.93	.92	.11	.90
	1000	.94	.88	.16	.88
	Mean	**.94**	**.93**	**.17**	**.91**
E-Prime	150	.90	.82	.92	.90
	200	.96	.98	.98	.86
	250	.96	.94	.96	1.00
	1000	.92	1.00	.96	.90
	Mean	**.94**	**.94**	**.96**	**.91**
		Machine 5			
Authorware	150	.97	.89	.99	.90
	200	.93	.91	.97	.95
	250	.97	.87	.96	.95
	1000	.74	.80	.63	.76
	Mean	**.90**	**.87**	**.89**	**.89**
E-Prime	150	.84	.88	.90	.82
	200	.92	.82	.92	.84
	250	.88	.86	.86	.84
	1000	.92	.82	.74	.92
	Mean	**.89**	**.85**	**.86**	**.86**

Table 4

Proportion of On-Target Measurements as a function of Time for Machines 1and 2 in Dedicated Load Condition

		Machine 1		Machine 2	
Mode	Interval	Time 1	Time 2	Time 1	Time 2
Netscape	150	1.00	.98	.88	1.00
	200	1.00	.98	.64	.98
	250	.96	.96	.90	.92
	1000	.90	.96	.50	.94
Internet Explorer	150	1.00	.98	.94	.92
	200	1.00	.96	.74	.90
	250	1.00	1.00	.96	.98
	1000	.92	.92	.94	.96
Executable	150	.94	.80	.84	.96
	200	.94	.94	.84	.96
	250	.98	.92	.86	.82
	1000	.98	.98	.18	.92

Table 5

Proportion of On-Target Measurements as a function of Time for E-Prime on Machine 1 in Default Load Condition

	Machine 1		
Interval	Time 1	Time 2	Time 3
150	1.00	.96	.98
200	.98	.92	.98
250	.98	.90	.92
1000	.98	.94	.96

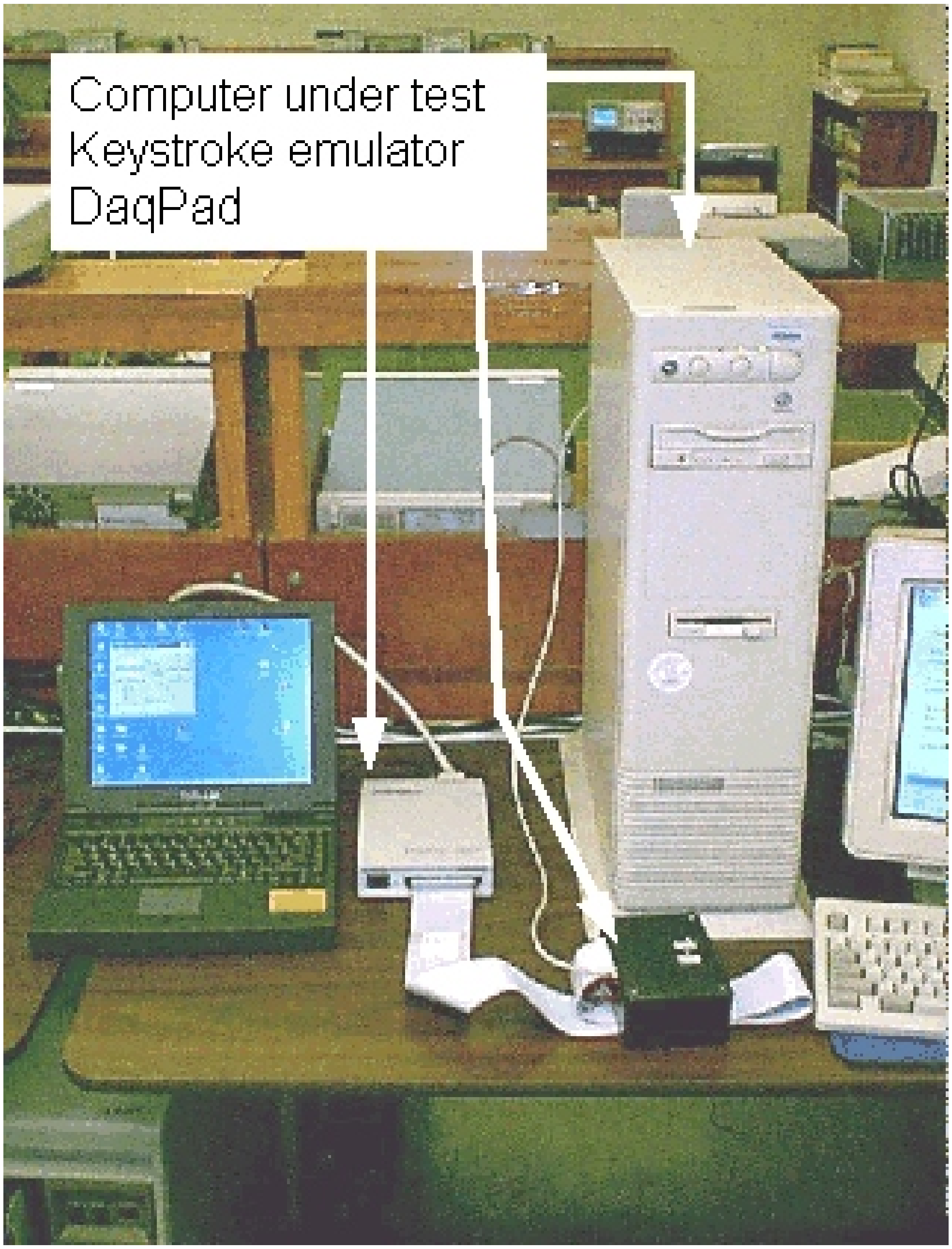
Figure Captions

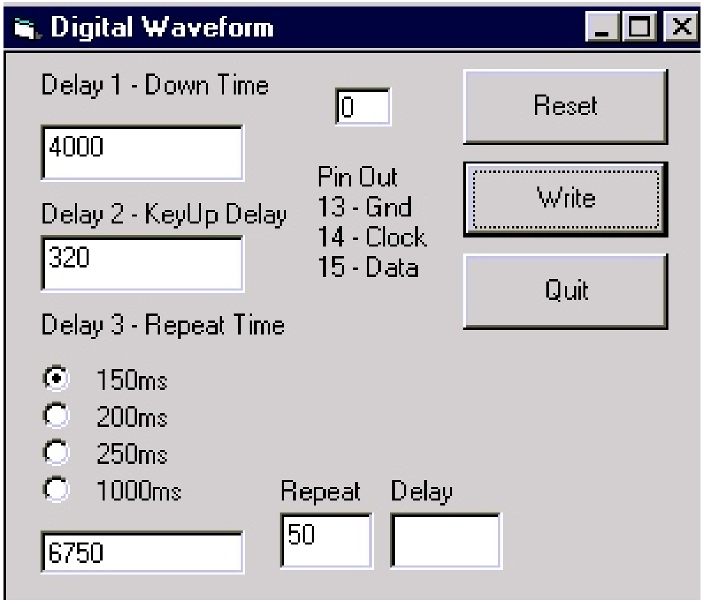
Figure 1. Experimental apparatus.

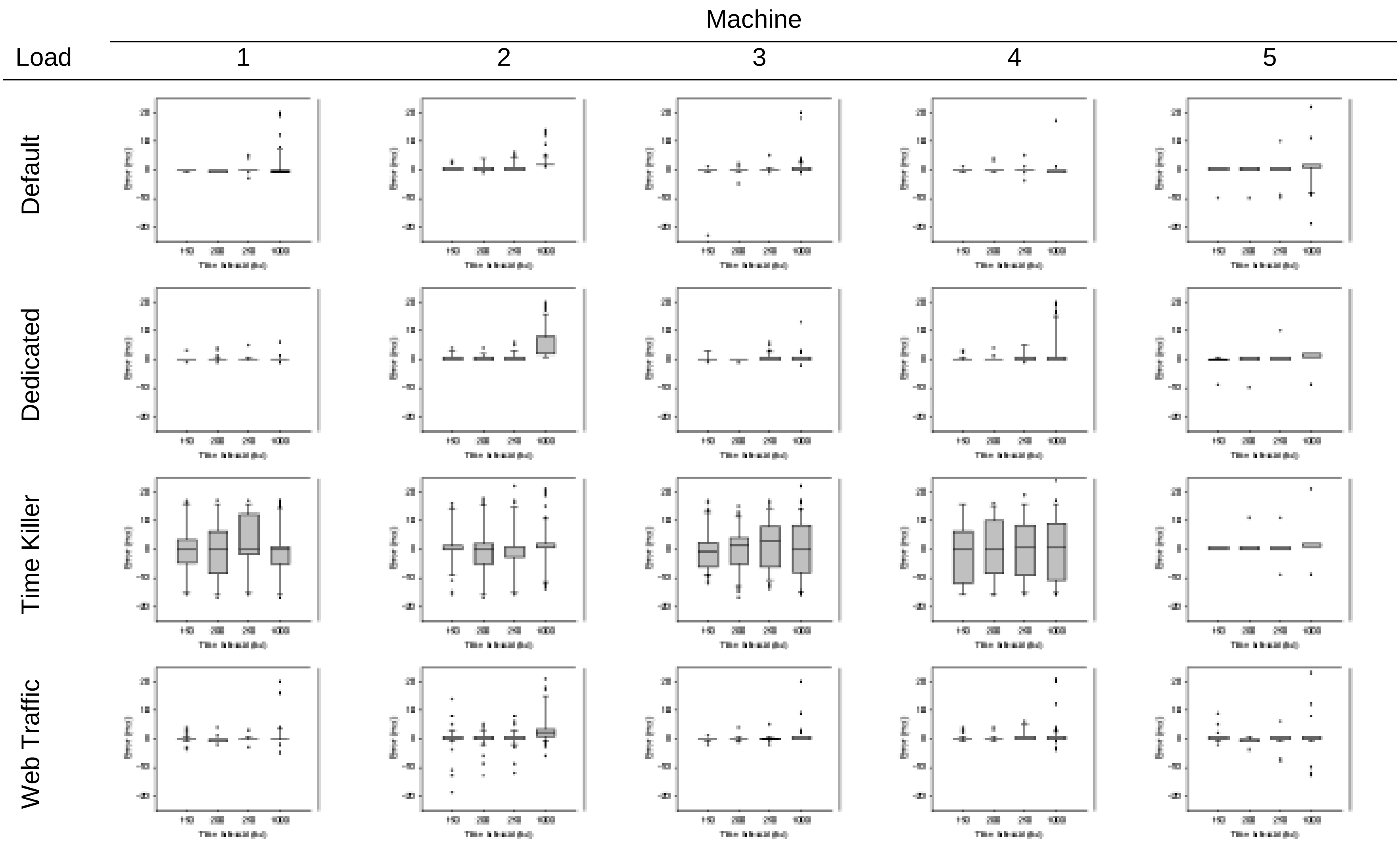
Figure 2. Interface for program used to control DaqPad.

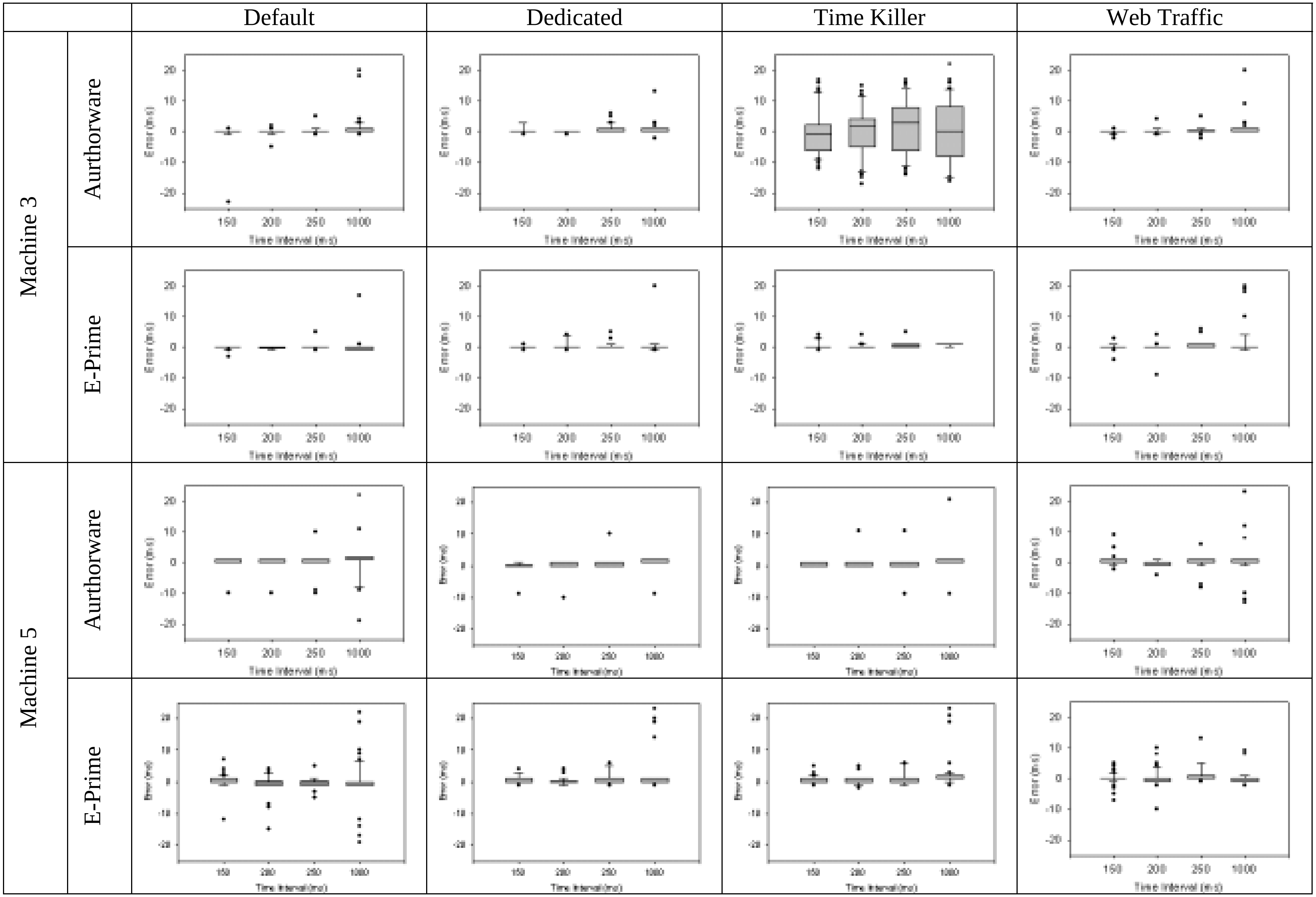
Figure 3. Box plots of error distributions as a function of Machine, OS Load, and Time Interval. Data are for Authorware program running in .exe (executable) mode.

Figure 4. Box plots of error distributions for measurements made using E-Prime and Authorware for Machines 4 and 5


Computer under test
Keystroke emulator
DaqPad


Digital Waveform
Delay 1 - Down Time
4000
0
Reset
Pin Out
13 - Gnd
14 - Clock
15 - Data
Write
Delay 2 - KeyUp Delay
320
Quit
Delay 3 - Repeat Time
150ms
200ms
250ms
1000ms
Repeat
Delay
6750
50


Machine
Load
1
2
3
4
5
Default
Dedicated
Time Killer
Web Traffic


Default
Dedicated
Time Killer
Web Traffic
Machine 3
Machine 5
Aurthorware
E-Prime
Time Interval (ms)
Error (ms)