UNIVERSITY OF ALBERTA

Consistency in Cognitive Performance: Age-Related Changes in Intraindividual Variability

By

William J. Davis

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

MASTER OF ARTS

DEPARTMENT OF PSYCHOLOGY

EDMONTON, ALBERTA

FALL 1995

ISBN 0-612-06336-4

UNIVERSITY OF ALBERTA

LIBRARY RELEASE FORM

NAME OF AUTHOR: William James Davis

TITLE OF THESIS: Consistency in Cognitive Performance: Age-Related Changes in Intraindividual Variability

DEGREE: Master of Arts

YEAR THIS DEGREE GRANTED: 1995

SIGNED: ....

PERMANENT ADDRESS:

14622 103 Avenue,

Edmonton, Alberta

DATE: Oct. 6/95

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled: "Consistency in Cognitive Performance: Age-Related Changes in Intraindividual Variability" submitted by William J. Davis in partial fulfillment of the requirements for the degree of Master of Arts.

....

Supervisor

....

....

Date: Oct 6, 95

Abstract

Researchers in the field of gerontology have traditionally attempted to investigate age-related changes in cognitive abilities based on differentiating mean performance among age groups. More recently however, the notion that intraindividual variability, traditionally viewed as measurement error, may serve as a valuable measure of the cognitive changes associated with normal aging. The present research tested this hypothesis by differentiating performance variability among older and younger adults on three cognitive tasks: A lexical decision task, a line length estimation task, and a time estimation task. The results showed that older adults displayed much greater intraindividual variability in both lexical decision and time estimation performance, while no differences in line length estimation were observed. A model for age-related changes in cognitive performance variability is proposed, and future directions for employing normative intraindividual variability scores as a diagnostic tool for the detection of Alzheimer's disease are discussed.

Acknowledgements

I would like to thank Dr. Allen Dobbs for his insightful comments and suggestions, and his confidence in my abilities. I would also like to thank Dr. Don Schopflocher and Sean Drake for their programming skills and for generous donations of their time.

Finally I would like to express my gratitude to Laura Copeman and Margret Tulloch for their valuable assistance in participant recruitment.

Table of Contents

List of Tables

List of Figures

I. INTRODUCTION

Over the past several decades, researchers have devoted a considerable amount of their efforts to discovering age related differences in a wide range of cognitive and behavioural abilities. Although much light has been shed on the performance changes associated with aging, most researchers have discussed these age related differences in terms of group mean comparisons. More recently, investigators have expressed their concerns with this methodology, as there is reason to believe that performance variability may also change with age. Two ways in which variability has been discussed are interindividual (within-group) variability, and to a much lesser degree; <u>intra</u>individual (within-subject) variability.

Interindividual variability has often been commented by aging researchers, as it has frequently been found that older adults tend to be a far more heterogeneous group than younger adults in terms of cognitive (Adams, 1991; Scukanek, Petrosino, & Rastatter, 1992), perceptual (Cranford & Stream, 1991; North & Fairchild, 1993; Slawinski, Hartel & Kline, 1993), affect (Lawton, Kleban, Dean & Rajagopal, 1992), motor (Swanson & Lee, 1992), and neurological characteristics (Wauquier, 1991). As a result, researchers often appreciate that group mean comparisons are rarely sufficient for a complete understanding the changes associated with the normal aging process. This is because the larger within-group variability scores for older adults might indicate existing sub-populations within the group. This questions the often implicit assumption underlying the use of group means, namely that the aging process is uniform across individuals.

Unfortunately, most aging researchers still neglect to consider variability as a potentially valuable measure. In a paper reviewing 56 articles from gerontological research, Bornstein and Smircina (1982) reported that only four of these publications included a discussion of the statistical measures of variance. Furthermore, the issue of increased variability was mentioned in only two studies, and only one of these had included an appropriate statistical analysis to test the hypothesis of increased variance. This disappointing lack of attention to measures of dispersion was recently addressed in a more intensive review of gerontological research (Nelson & Dannefer, 1992). They reported that in only about one half of 127 articles reviewed were measures of dispersion mentioned, and the majority of these reported increasing patterns of variability with age. Again, few of these studies included any statistical analyses of the phenomenon. However, the authors were able to perform Bartlett's test for homogeneity of variance on variability measures for ten of the studies in this sample. In half of these, significant differences in within-group variability were observed. However, more research is required to confirm that it is the aging process (rather than factors such as illnesses, motivation etc.) that affects individuals differently, as suggested by interindividual variability differences.

More recently, in an effort to understand the nature of increased heterogeneity among older adults, researchers have proposed that this phenomenon is a reflection of greater intraindividual variability in performance abilities for older adults (Allen, 1991; Allison, 1989; Hertzog, Dixon & Hultsch, 1992; Mantyla

& Craik, 1993; Morse, 1993; Scukanec et al., 1992; Wilkinson & Bornstein & Smircina, 1982). That is, it may be that the performance of older individuals is much more variable than that of their younger counterparts, and this intraindividual variability might contribute to the frequently observed within-group heterogeneity.

This hypothesis is significant for studying individual differences. If the hypothesis is confirmed, a reassessment of many previous observations of group differences and the applications and stereotypes that have been derived from those observations, might be warranted. Further, the hypothesis of age related differences in cognitive performance may lead to a methodological revolution in psychology since intraindividual, or within-subject variability, which has been traditionally viewed as statistical error, may in fact turn out to be a reliable measure of individual differences in cognitive aging and other areas of research.

At present, there exist two schools of thought regarding the underlying source of this variability.

Some researchers believe that older adults exhibit greater variability as a result of greater susceptibility to external influences (Hertzog, Dixon & Hultsch, 1992). Alternatively, Davis and Dobbs (1993) proposed that an increase in <u>intra</u>individual variability is an overt consequence of a more variable internal cognitive system. They suggested that older adults process information in a more inconsistent fashion than younger adults, and this variability in processing is an integral part of the aging process independent of any greater susceptibility to external influences.

In Hertzog et al.'s (1992) study, seven elderly women were tested weekly on text recall performance for a period of 70 weeks. A great deal of variability in performance within subjects was observed across the 70 sessions. They concluded that the increase in performance variability is caused by older adults experiencing greater fluctuations in mood states because of a greater susceptibility to various environmental stimuli. There were, however, two methodological problems with this research. First, their longitudinal approach was not paired with an examination of younger adults for comparison. Thus, their contention that intra-individual variability is greater in an adult's later years is unjustified. Second, no effort was made to assess within-session variability. The observed performance variability across testing sessions might also have existed within the single testing sessions. Therefore, the possibility that intra-individual variability is internally driven was not considered in their research.

Less systematic clinical observations also have attributed observed cognitive performance variability in older adults to greater fluctuations of non-cognitive factors such as physical and mental health (Kaszniak, 1990). Clinicians have, for example, often noted that older individuals show greater behavioural variability between visits. Again, nothing was said regarding possible age related differences in behavioural variability within a single visit.

The advantage of measuring an individual's performance variability within one testing session is that one can minimize the influences from the potentially variable external factors. If significant variability among older adults is observed in this paradigm, its cause might be attributed to the individual's processing inconsistencies as opposed to external factors.

Evidence consistent with the suggestion that increased variability is an internally driven phenomenon was provided by Davis & Dobbs (1993). Intraindividual variability scores for old and young adults were compared for reaction time measures obtained in a single testing session. Older adults exhibited significantly greater variability in performance across trials. Since no changes in their environment during the brief testing session were apparent, we concluded that the observed differences could only have resulted from differences in the consistency of cognitive processing for young and old adults.

This hypothesis is consistent with neurological research that has revealed redundancy in the nervous system of higher order species (Patterson, Foster & Heron, 1980; Glassman, 1987). The loss of redundancy has been shown to result in increased performance variability in both biological systems and in artificial neural networks (Calvin, 1983; Jacobson, 1976; Plotkin & Russell, 1972).

Rationale

The purpose of the present research was twofold. First, we intended to test the hypothesis that age differences in intraindividual variability in cognitive performance exists. Second, we wished to use these results to address the hypothesis that the source of the variability differences is a more variable cognitive system within older individuals. This can be achieved by measuring variability within a single session, since the external influences discussed by Dixon et al. (1992) and Kaszniak (1990) are expected to remain relatively constant for the brief duration of the testing sessions.

In selecting tasks for the present study, two criteria had to be met. First, two tasks in which age differences are known to exist, or suspected to exist were selected. A third task in which mean age differences were not expected was selected. This will enable me to determine if differences in intraindividual variability are concurrent with observed mean differences. Second, paradigms in which confounding due to ceiling or floor effects would be minimized were selected. Ceiling and floor effects are simply upper and lower limits respectively, imposed on the dependent measures in a particular study. These effects are problematic in that an individual's potential range of performance is reduced, and thus any derived statistical measures are invalid. In order to avoid ceiling and floor effects, tasks which do not restrict the range of dependent measures were needed.

The first task involved a lexical decision paradigm in which response latencies in milliseconds were measured. Research has shown that younger adults are significantly faster than older adults at lexical decision performance (Howard, 1988; Howard, Shaw, & Heisey, 1986). Since floor effects are intrinsic to the usual format of this paradigm, these effects were minimized by presenting four letter strings per trial instead of just one. The goal was to increase the potential range of the response time distributions. The easy and hard conditions in this task included groups of low or high frequency words, respectively, as it has been consistently demonstrated that word frequency is one of the most robust factors affecting both reading and lexical decision performances (Morton, 1969; Dobbs, Friedman, & Lloyd, 1985).

The second task chosen further minimized confounding due to ceiling or floor effects. It involved a line length estimation paradigm, where the magnitudes of errors are unlikely to be artificially restricted. Although an exhaustive search of the aging literature failed to discover any previous use of this paradigm, large age differences in mean estimation performance were not expected in this task. Many of the cognitive processes involved have been shown to remain relatively unchanged with age. Salthouse (1985) has discussed age-related changes in working memory primarily in terms of speed of processing differences. Working memory factors such as capacity and accuracy were shown to remain relatively constant throughout the life span. The present line estimation task does not impose time constraints on participants, and it was therefore expected that mean differences due to age will be minimal. The line estimation task required individuals to view a vertically presented line, and reproduce it horizontally. In the easy condition, simple mental rotation and replication skills were employed. In the hard condition, memory processes were also active, as the stimulus line will be removed from view.

Our third task was a dual-task procedure. The primary task involved time estimations in which subjects were not unreasonably constrained in terms of estimation accuracy. The secondary task was a digit tracking task similar to that employed by Dobbs & Rule (1989), whereby older individuals were found to perform as well as younger adults in the digit tracking condition. This task is added to the paradigm to provide easy and harder conditions to the primary task. Research has shown that estimated duration of a given interval becomes shorter with increases in the demands of concurrent cognitive processing (Fortin & Rousseau, 1987). The difficulty of the time estimation task was determined by the condition of the secondary digit tracking task. In the no lag condition, participants simply estimated the amount of time that had lapsed between two tones. In the digit tracking condition, visual presentation of single digits was successively presented, and participants simply reported each digit as they appeared.

In each of the three tasks, we expected to observe significant main effects of age for intraindividual variability measures. Significant differences between the group means of individual standard deviation scores would support this hypothesis. Further, we expected to observe (age)x(condition) interactions such that the increased cognitive demands associated with the hard conditions will more greatly affect older individuals' mean and variability scores.

II. METHOD

PARTICIPANTS

Forty community-dwelling adults volunteered to participate in this research. Recruitment of participants was primarily by oral solicitation. Additional participants for the older group were solicited from the Edmonton Good Time Rovers (a local recreational vehicle travelling organization) by means of a posted advertisement.

Twenty adults with ages ranging from 25-33 years, and a mean age of 27.6 years represented the younger group, and 20 adults with ages ranging from 65 to 75 years, and a mean age of 68.5 years represented the older group. Subjects in the younger group had a mean of 15.8 years of education, and the older adults had a mean of 13 years of education. Groups were also matched for sex, with 11 females and 9 males in each. Further, participants were screened for normal physical health through self ratings, and matched on verbal skills using the Mill-Hill vocabulary test (Raven, 1958). All participants were native English speakers, and had normal or corrected-to-normal vision (either with prescription eyeglasses or contact lenses), and all participants were unaware of the hypothesis under investigation.

APPARATUS

All tasks were presented on a Fujikama (IBM compatible) 33 MHz, 486dx personal computer. Visual stimuli were presented on a Goldstar VGA colour monitor. An IBM extended keyboard served as the input device for all tasks. The keys to be used by participants were covered with adhesive colour labels to eliminate potential confusion with inappropriate keys. Sessions were run in a semi-darkened room to minimize any possible contextual distractions.

PROCEDURE

Task 1: Lexical Decision. Eighty trials were presented in which subjects were presented with four letter strings per trial. On one half of the trials, four words appeared in the centre of the monitor. On the other half of the trials, three words plus one pronounceable nonword were presented. The position of the nonword among the three words was counterbalanced across the 40 trials. Subjects were instructed to decide as quickly and accurately as possible whether a nonword was absent or present by pressing the blue (z) or red (/) key respectively, on the computer keyboard. See Appendix B for complete subject instructions.

Each trial commenced with simultaneous presentations of a central fixation box in the centre of the computer screen and a 500-Hz tone lasting for 500 msec. After 1000 msec, the four letter strings appeared in the box. The display remained on the screen until a response was made. Successive trials were presented following a 1500 msec inter-trial interval. The 80 item lists were divided into two blocks, and the order of block presentation was counterbalanced across subjects. Each participant was presented with 10 practice trials to familiarize themselves with the task demands and the response keys. An accuracy criterion of 80 per cent was required before moving on to the experimental session. None of the participants required additional practice. Response times in milliseconds, and accuracy were recorded by the computer.

One hundred forty high-frequency and 140 low-frequency words were selected from Kucera & Francis (1969). High frequency words had a frequency of 250 or more, and low frequency words had a frequency less than or equal to five. Forty nonwords were created by changing 1 letter in each of 40 additionally chosen words. An additional 40 words of both high and low frequency were chosen for the 10 practice trials. See Appendix A for a complete listing of stimulus items.

Note that spoiled trials in this task were defined as response latencies under 1000 msec. The computer programming for this task would interpret the second of an accidental double button push as the response for the successive trial. Such responses were recorded from 0 to just under 1000 msec. There were very few spoiled trials of this type.

Task 2: Line Length Estimation. Forty trials were presented in which subjects were instructed to horizontally replicate the length of a vertically presented line. On each trial, a vertical line was presented onscreen with the starter segment of a horizontal line appearing below it. The length of the vertically presented lines ranged from 180 and 220 pixels, and had a mean length of 200 pixels. The set of stimulus items was specifically determined as opposed to random length generation. Subjects adjusted the length of the horizontal line by pressing blue key with their right index finger or the red key with their left index finger to respectively lengthen and shorten their estimates. Pressing the spacebar indicated they were satisfied with their estimation and completed a trial. On one half of the 40 trials, the vertical line remained onscreen until the response was complete. These trials were labelled the easy condition.

On the other half of the trials, the vertical line was presented for a 3000 msec memorization period, and would then disappear from view before the participants began their responses. The two types of trials were randomly arranged within two blocks, and the order of block presentation was counterbalanced across subjects.

Participants were initially presented with 10 practice trials to familiarize themselves with the task demands and the input controls before proceeding on to the experimental session.

The difference between the actual and estimated line lengths in pixels were computed and served as the dependant measure. Note that computer terminal pixels are longer vertically than horizontally, and error magnitudes may be misleading. However the relationship in error performances between individuals remains unchanged.

Task 3: Time Estimation. A dual-task paradigm was employed in which the primary task involved a time estimation task whereby each of the 40 trials commenced with a 1000-Hz tone lasting for 1500 msec, and terminated with a 1000 msec tone of the same frequency. Participants were asked to estimate the total trial time by tapping the spacebar once to represent the start of the trial, and once again to represent their estimation of total trial time. The actual time from offset of the first and second tone presentations varied from 1000 to 1600 msec. The real difference in milliseconds between the actual and estimated times served as the dependent measure.

Hard and easy conditions were defined by the load of the secondary task. For the easy, or no lag condition, no secondary task was presented. Participants simply devoted all of their attention to the primary task. For the hard, or simple tracking condition, successive onscreen presentations of single digits were presented for the duration of time between the start and end tones. Participants were instructed to verbally report each digit as it appeared until the onset of the second tone.

The hard and easy trials were presented in two separate blocks, and the order of block presentation was counterbalanced across participants. Justification for segregating the two trial types this way came from pilot participant verbal reports. For those participants, the two trial types were mixed together. Each of the participants reported that at the onset of each trial, much of their attention was devoted to determining whether it would be an easy or hard condition as they awaited the possible first digit presentation. Consequently, the additional cognitive processes involved with this anticipation period might disrupt their estimation of time. In order to remove the uncertainty period confound, we presented the two trial types separately, and therefore the participants would be aware of the procedure before the onset of each trial.

Participants were presented with 5 practice trials for each condition to familiarize themselves with the task demands and the input keys before proceeding on to the experimental trials. Only two of the elderly participants required additional practice.

It was necessary to define both upper and lower outlier scores in this task as participants occasionally reported losing sight of exactly where they were in the sequence of trial events. For example, participants occasionally thought they were waiting for a tone to be presented, when in fact the computer was waiting for an input response. Such an incident would result in an extremely large overestimation score for that trial. A criterion level of 15 or more outliers would exclude a participant's data from further analyses. Only one of the elderly subjects was removed for meeting this exclusion criterion.

III. RESULTS

Mean Performance

Task 1: Lexical Decision. The group means for both conditions are presented in Figure 1. Group mean reaction times were computed and analyzed using a 2(group: old, young) x 2(frequency: high, low) analysis of variance. Response latencies from trials in which nonwords were present were excluded from the analysis, since the number of words the participant was required to read varied from trial to trial and responding to nonwords involves less well understood processing mechanisms. Results from the analysis showed a significant effect of age, $F(1, 35) = 29.11$, $p < .001$. Older adults produced response latencies that were greater than those of younger adults (Mean Old = 3216.5 msec; Mean Young = 1940.6 msec). A significant main effect of frequency was also obtained, $F(1, 35) = 32.32$, $p < .001$, whereby high frequency word were responded to more quickly than low frequency words. The analysis failed to reveal a significant interaction, demonstrating that frequency did not differentially affect response latencies for young and older adults.

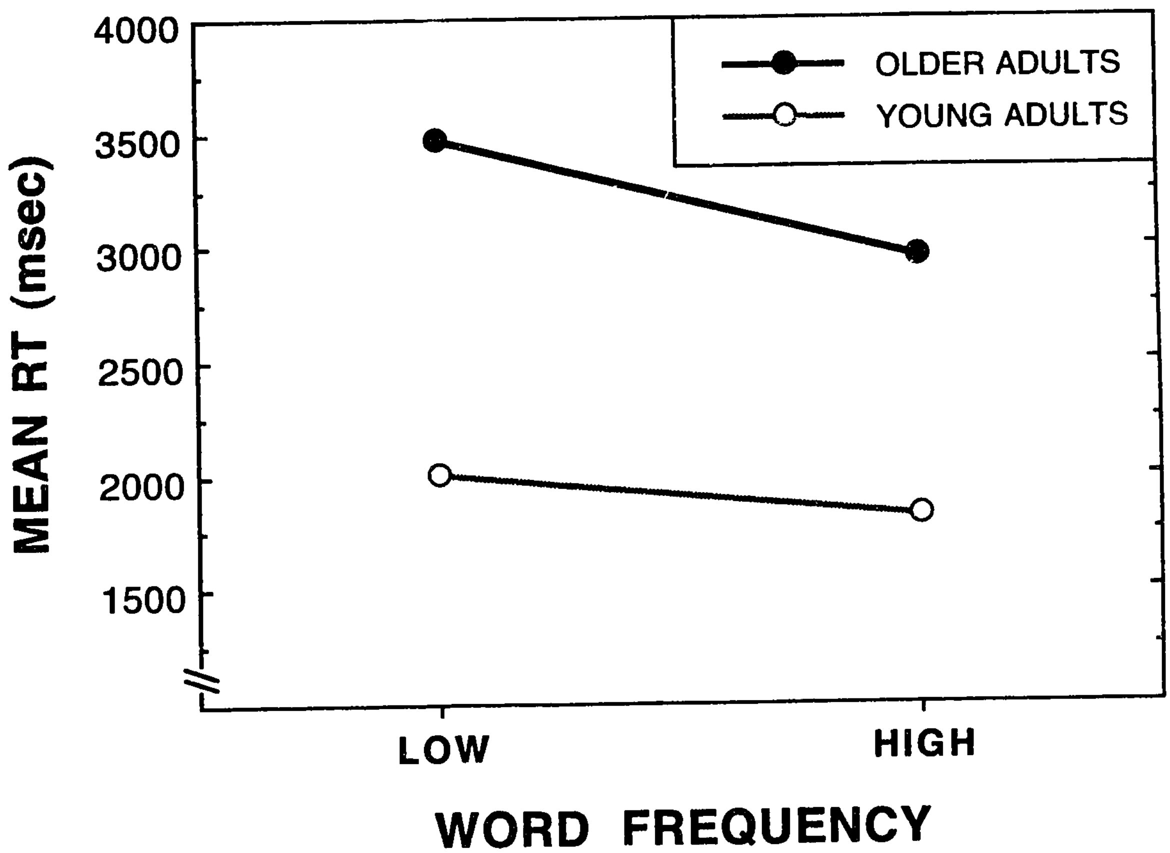

Figure 1. Mean Response Latencies for the Lexical Decisions Task.

Mean group accuracy scores for both conditions are presented in Table 1. Performance accuracy was analyzed in a 2(age: old, young) x 2(frequency: high, low) ANOVA. A significant main effect of frequency was observed, $\underline{F}(1, 36) = 46.6$, $\underline{p} < .001$. No main effect of age was obtained, demonstrating that more errors occurred for lists of low frequency words than for those of high frequency words equally for both young and older adults. This pattern of errors also demonstrates that speed-accuracy trade offs in the reaction time data were unlikely.

Table 1

Mean Proportion of Correct Responses

Word Frequency	Age Group	
	Young	Old
High	0.98	0.98
Low	0.78	0.79

Task 2: Line Length Estimation. Group mean errors in real values for both conditions are reported in Table 2. The group mean errors were computed and analyzed using a 2(age: old, young) x 2(condition: present, absent) analysis of variance. No significant effects were observed for this task.

Table 2

Mean Error Scores as a Function of Age X Condition

Condition	Age Group	
	Young	Old
Easy	59.1	59.7
Hard	60.0	61.3

Note. Error values are expressed in terms of number of pixels.

Task 3: Time Estimation. Group mean errors for each condition are presented in Figure 2. Mean errors were computed and analyzed using a 2(age: old, young) x 2(condition: Easy, Hard) x 2(order: Easy first, Hard first) ANOVA. A significant main effect of condition was obtained, $F(1, 35) = 101.6$, $p < .001$, indicating that both groups were equally affected by task difficulty.

Scatterplots of mean scores for each of the three tasks were also created, and the distribution patterns confirmed linearity of the data and did not indicate any obvious floor or ceiling effects.

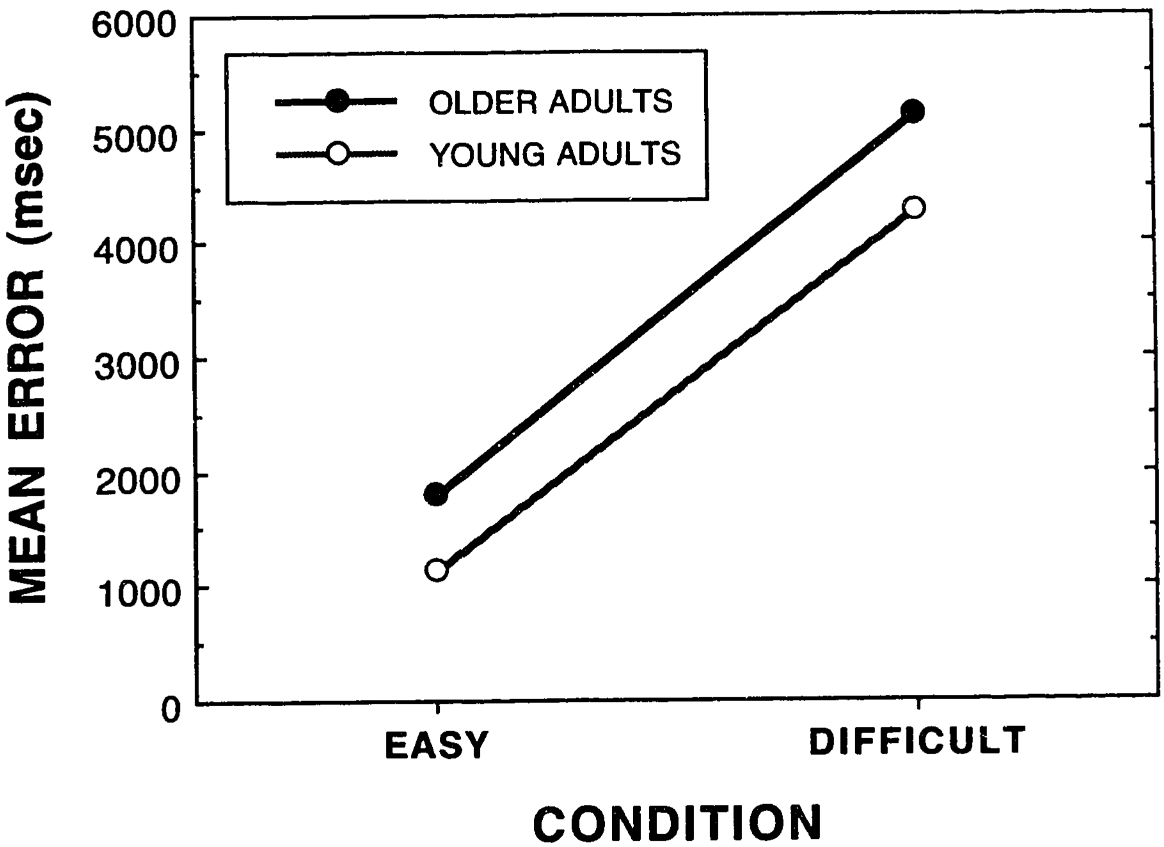

Figure 2. Mean Error Scores for the Time Estimation Task.

Intra-Individual Variability

Task 1: Lexical Decision. Individual standard deviations were computed for response latencies. Since it can be argued that greater variability scores might naturally be associated with larger mean values, coefficient of variability (CV) scores are occasionally computed and employed as an unbiased measure of variability (Davis & Dobbs, 1993; Morse 1993). Each participant's CV is computed by dividing their standard deviation by the group mean, thus parsing out any potential confounding due to mean effects. In order to determine whether computations of CV measures were warranted for comparisons, standard deviations were correlated with mean scores. Although no a priori threshold was established to define a minimum Pearson's r value that would indicate a biased estimation of variability, post hoc decisions were made regarding the importance of the obtained correlations.

Group mean standard deviations for each condition are presented in Figure 3. Results showed that mean performance did not correlate highly with variability, r = 0.43, $p < .01$ for low frequency trials and r = 0.48, $p < .01$ for high frequency trials. Since these correlations are large enough to account for over 20 per cent of the variability, separate ANOVAs were performed on both CV and standard deviation scores. Since the results from the two ANOVAs were the same, analyses of the standard deviation scores are reported in order to maintain consistency throughout this paper.

Group mean standard deviations were analyzed using a 2(age: old, young) x 2(frequency: high, low) ANOVA. A significant main effect of age was observed, $F(1,35) = 10.33$, $p < .003$. Older adults displayed greater variability in response times across both conditions. No other significant effects were obtained.

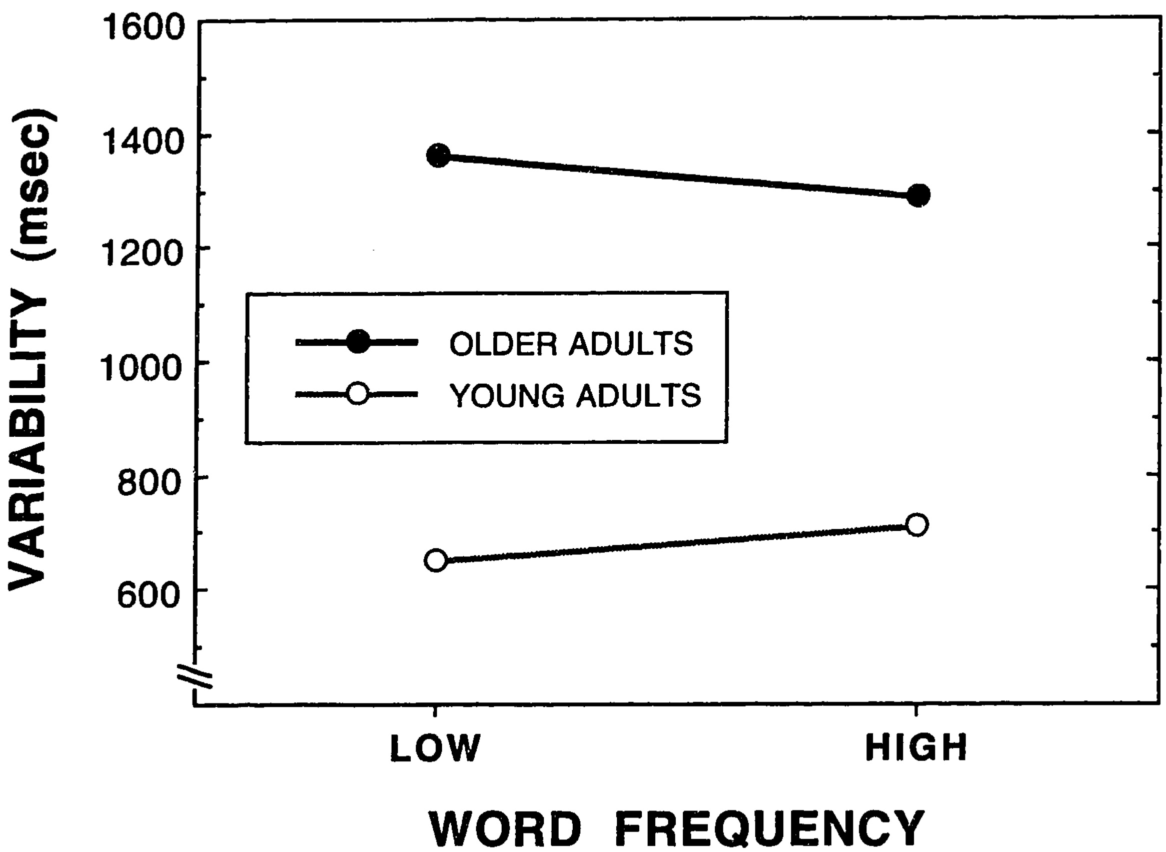

Figure 3. Mean Intraindividual Standard Deviation Scores for the Lexical Decision Task.

Task 2: Line Length Estimation. Each subject's standard deviation scores were computed across estimation errors. Group mean standard deviation scores are presented in Table 3. Correlations between standard deviation scores and mean scores were again small enough that the use of CV measures was unjustified, $r = -0.24$, (n.s.) for easy trials and $r = -0.29$, $p < .05$ for hard trials. Mean group standard deviations were analyzed using a 2(age: old, young) x 2(condition: present, absent) analysis of variance. As with the mean score data for this task, no significant effects were obtained.

Table 3

Mean Intraindividual Standard Deviation Scores as a Function of Age X Condition.

Condition	Age	
	Young	Old
Easy	12.9	13.1
Hard	12.4	13.8

Note. Standard deviations of errors are expressed in terms of number of pixels

Task 3: Time Estimation. Each subject's standard deviation scores were computed across estimation errors in each condition. Once again, standard deviation scores did not correlate highly with mean scores, r = 0.36, $p < .05$ for the easy trials and r = 0.18 (n.s.) for the hard trials. Therefore the unbiased individual standard deviation scores were chosen for comparison. The mean standard deviations for each condition are presented in Figure 4. Mean group standard deviations for both conditions were analyzed using a 2(age: old, young) x 2(condition: Easy, Hard) x 2(order: Easy first, Hard first) analysis of variance. A significant main effect of age was obtained, $F(1,35) = 16.86$, $p < .001$. Older adults produced greater variability scores in both conditions than younger adults. A significant effect of condition was also obtained, $F(1,35) = 6.23$, $p < .017$. In general, the Hard, or simple tracking condition yielded greater variability scores than did the easier condition.

Scatterplots of standard deviation scores for each of the three tasks were examined, and again linearity of the data were confirmed. There was also no indication of obvious ceiling or floor effects.

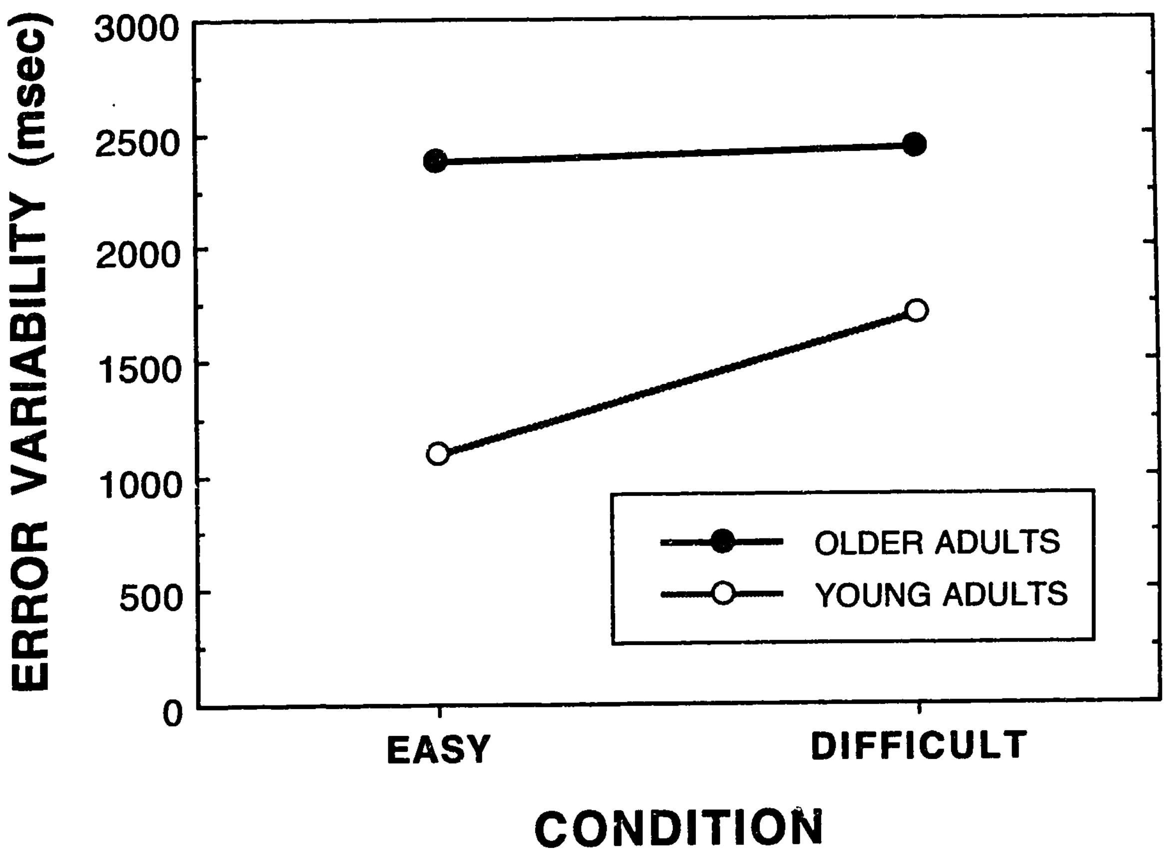

Figure 4. Mean Intraindividual Standard Deviations of Estimate Errors for the Time Estimation Task.

IV. DISCUSSION

Mean Scores

The results from the lexical decision task demonstrate the general slowing of cognitive processing associated with normal aging that is generally agreed upon in aging literature (Howard, 1988). Since no age differences in accuracy were observed, the idea that an overall decline in lexical decision performance occurs with age is not likely. Both young and older adults were equally affected by the word frequency effect, indicating that the semantic memory system appears to remain intact throughout the normal aging process.

The line estimation task failed to reveal mean differences due to age or condition. The absence of age differences supports earlier findings that the cognitive processes involved in this task such as perception of length and a minimal taxing of working memory appear not to change significantly with age.

The non-significant effects of condition may in some part be a reflection of a lack of sensitivity to the dependent measure in this paradigm. Responses in the difficult condition were made immediately after the offset of the vertical line, and the proximity of the two lines might have contributed to the light demands of the task. A conscious effort was made however, to avoid creating conditions that would be too difficult, since future application of this paradigm will involve participants with impaired cognitive abilities (see Directions for Future Research section of the General Discussion).

As with the line length estimation task, the cognitive processes involved in the perception of time intervals appear to remain quite stable throughout the adult lifespan. As discussed by Salthouse (1985), the non-speeded components of working memory such as capacity and accuracy were shown to remain stable across age groups. Results from the time estimation task demonstrate that an increase in cognitive load has an enormous effect on one's perception of time, and this phenomenon equally affects both young and older adults. While both groups showed moderate underestimations of time intervals in the easy condition, the seemingly untaxing secondary demands of the simple tracking conditions were sufficient to cause a great reduction in temporal perception.

Intraindividual Variability Scores

Results from reaction time data from the lexical decision task demonstrate that, independent of mean performance, older adults tend to be much more variable in performance than are younger adults. Unlike mean performance, standard deviations of response latencies were unaffected by word frequency. This is an indication that the source of the variability differences lie in the pre- or post-lexical processes in the cognitive system. In other words, older adults display greater inconsistencies in either input processes, output processes, or both. Kausler (1982) has argued that older adults display deficits in the qualitative aspects of memory systems. While encoding and retrieval processes tend to decline with age, the crystallized information tends to remain quite stable.

The absence of age and condition effects on intraindividual variability scores for the line length estimation task again suggest that the processes involved in estimation of line length remain unchanged throughout the adult lifespan. Task difficulty also proved to have no differential effect on variability scores. Paired with the results from the mean data analysis, these results appear to indicate that the condition differential was not sufficient to produce changes in cognitive performance in a healthy cognitive system.

Results from the time estimation task show that overall, older adults were far more variable in temporal estimation accuracy than were younger adults. The results also indicate that variability in estimation accuracy increased with task difficulty. Looking closely at Figure 3 however, we can see that this condition effect is slightly misleading. The magnitude of variability for the group of older adults is large yet stable across conditions. It is the increase in variability among younger adults which yield the condition effect. This increase, however, was not great enough to produce an interaction between age and condition.

V. GENERAL DISCUSSION

The purpose of this research was to assess the hypothesis that _intra_individual variability in cognitive processes increases as a result of the normal aging process. Further, attempts were made to identify which cognitive processes would show this pattern of increased variability, and which would remain stable. Finally, the hypothesis that increases in _intra_individual variability associated with old age is the result of internally-driven factors was investigated.

Both the lexical decision task and the time estimation task revealed that significant increases in performance variability was associated with the normal aging process. An interesting result from both of these tasks was that the older adults displayed high, yet consistent patterns of variability across the task conditions. In other words, the results indicate that while _intra_individual variability in cognitive performance increases with normal age, this increase reaches a plateau that cannot be breached as a result of at least moderate increases in cognitive load. This might imply that the increase in _intra_individual variability associated with normal aging has a measurable maximal point. Additional research employing greater levels of task difficulty manipulations is required to properly address this possibility.

The results from the line length estimation task indicate that one area of cognitive processing may have been identified that remains unaffected by the normal aging process. Although the nature of the paradigm may have been too insensitive to produce significant effects of task condition, the more relevant issue is that older adults produced the same patterns of variability as the younger adults.

In contrast to Herzog et al.'s (1992) contention that age-related changes in intraindividual variability are the result of external influences, this experiment demonstrates that the internally-driven causes is quite likely. The repeated-measures design in this study serves to minimize potential external fluctuations by gathering measures in a stable context, and in a short period of time. This research is consistent with the notion that the internal cognitive processing system operates far less consistently as a function of the normal aging process.

The Redundancy Hypothesis

One possible explanation for the reason people's cognitive systems become more variable with age is the redundancy hypothesis derived from biological research on the nervous system and other biological systems (Glassman, 1987). Redundancies in the human nervous system have been identified, and viewed by Glassman as being primarily a compensatory system. He cites evidence of behavioural recovery from brain damage as a strong indication that the human brain, being at least twice as large as is necessary for short-term survival, possesses the means for neural "Backups" to compensate for damage due to injury or lesions.

It is also likely that these redundancies in a healthy nervous system are not merely compensation agents. These redundant neural pathways might process information independently, and in parallel, and thus result in more consistent patterns of overt behaviour. Given any cognitive task, a healthy, redundant processing system will activate not one, but several parallel neural pathways committed to processing that information. Just prior to the output stage of processing, the information produced by each of these redundant pathways is averaged in an output buffer, the result is available for a response. Imagine the same problem in a nervous system that has a number of its redundant pathways severed, perhaps due to injury, disease, or loss of cells due to normal aging. This would result in fewer signals reaching the output buffer of the model, and the variability of the available response alternatives would increase. Since fewer signals are submitted to the output buffer, the computed average for response will be much more sensitive to extreme signals, much like mathematical mean computations are more greatly affected by extreme, or outlier values as the number of values decreases. On repeated exposure to the same cognitive task, the averaging process is likely to compute different responses, or display less consistency across repetitions. As a result, overt cognitive or behavioural performance would be highly variable.

Research on brain damaged patients, and patients suffering from the demyelinization caused by multiple sclerosis (MS) have repeatedly been observed to display this type of variable behaviour (Clark, Gardner, Brown, & Gummow, 1992; Patterson et al., 1980). For example, Patterson and his colleagues have demonstrated that the <u>intra</u>individual variability in visual threshold among MS patients was far greater than that of a control group.

It is possible that humans are born with little redundancy in their nervous systems, and that the redundancies develop over time as a result of learning and experience. Consistent with this, developmental research has demonstrated that the cognitive performance of younger children is highly variable, and becomes more consistent in the late teen years. (Dayon, & Thomas, 1994; Hale, Fry, & Jessie, 1993). As a result of the normal aging process, neurons begin to deteriorate and loss of redundancy results, and cognitive performance once again increases in variability. Although further research is warranted to properly investigate this possibility, the differential results from the present research suggest that the increase in <u>intra</u>individual variability (perhaps resulting from a reduction in neural redundancy) is not entirely random. The absence of variability differences between young and older adults in the line length estimation task for example, suggest that neural atrophy progresses in a somewhat systematic fashion.

Directions for Future Research

Recent behavioural and cognitive studies in the area of Alzheimer's disease and Dementia have also made mention of increased intraindividual variability in performance among patients (Haupt, Kurz, & Pollman, 1992; Nichelli, Vernneri, Molinari, & Tavani, 1993). Attempts to diagnose patients based on mean performance measures for various cognitive abilities has proved to be quite unsuccessful. Measures of variability, however, may prove to be more useful in early detection of Alzheimer's disease.

Both brain imaging techniques and postmortem analysis has also proved to be disappointing, in that no distinct pattern of neural degeneration has been identified. The senile plaques and neurofibrillary tangles found in the central nervous systems of Alzheimer's patients appear to occur systematically, but researches tend to disagree on which areas of the brain are commonly affected and which are spared (Brun & Englund, 1986; Giannakopoulos, Hof, & Bouras, 1994).

In terms of diagnostics, it is possible that measures of cognitive variability will prove to be useful in supplementing further neurological revelations.

It is possible that the loss of redundancy and subsequent increase in <u>intra</u>individual variability in cognitive performance associated with normal aging is accelerated in people suffering from Alzheimer's disease. This hypothesis may now be addressed using the tasks employed in the present research. Results from the healthy elderly participants will serve a baseline measures against which the performance variability of patients diagnosed with Alzheimer's disease can be compared. It is my hope that results from this research will eventually serve to establish norms that can be employed as a novel, reliable procedure in the diagnosis of Alzheimer's disease.

References

Adams, C. (1991). Qualitative age differences in memory for text: A life-span developmental perspective. Psychology and Aging, 6(3), 323-336.

Allen, P. A. (1991). On age differences in processing variability and scanning speed. Journal of Gerontology, 46(5), 191-201.

Bornstein, R., & Smircina, M. (1982). The status of the empirical support for the hypothesis of increased variability in aging populations. The Gerontologist, 22(3), 258-260.

Brun, A., & Englund, E. (1986). Brain changes in dementia of Alzheimer's type relevant to new imaging diagnostic methods. Progress in Neuro-Psychopharmacology & Biological Psychiatry, 10(3-5), 297-308.

Calvin, W. H. (1983). A stone's throw and its launch window: Timing precision and its implications for language and hominid brains. Journal of Theoretical Biology, 104, 121-135.

Clark, E., Gardner, M. K., Brown, G., & Gummow, L. (1992). Components of analogical reasoning in a mildly head injured population. Current Psychology Research and Reviews, 11(1), 21-35.

Cranford, J. L., & Stream, R. W. (1991). Discrimination of short duration tones by elderly subjects. Journals of Gerontology, 46(1), 37-41.

Davis W. J., & Dobbs, A. R. (1993). Age-related changes in performance variability. Poster session presented at the 2nd annual CARNET conference, Toronto, ON.

Dayan, A., & Thomans, J. R. (1994). Intention to remember spatial location in movement: Developmental considerations. Human Performance, 7(1), 37-53.

Dobbs, A. R., Friedman, A., & Lloyd, J. (1985). Frequency effects in lexical decisions: A test of the verification model. Journal of Experimental Psychology: Human Perception and Performance, 11, 81-92.

Dobbs, A. R., & Rule, B. (1989). Adult age differences in working memory. Psychology and Aging, 4(4), 500-503.

Feingold, A. (1995). The additive effects of differences in central tendency and variability are important in comparisons between groups. American Psychologist, 50(1), 5-13.

Giannakopoulos, P., Hof, P. R., & Bouras, C. (1994). Alzheimer's disease with asymmetric atrophy of the cerebral hemispheres: Morphometric analysis of four cases. Acta Neuropathologica, 88(5), 440-447.

Glassman, R. B. (1987). An hypothesis about redundancy and reliability in the brains of higher species: Analogies with genes, internal organs, and engineering systems. Neuroscience & Biobehavioral Reviews, 11, 275-285.

Hale, S., Fry, A. F., & Jessie, K. A. (1993). Effects of practice on speed of information processing in children and adults: Age sensitivity and age invariance. Developmental Psychology, 29(5), 880-892.

Haupt, M., Kurz, A., & Pollman, S. (1992). Severity of symptoms and rate of progression in Alzheimer's disease: A comparison of cases with early and late onset. Dementia, 3(1), 21-24.

Hertzog, C., Dixon, R. A., & Hultsch, D. F. (1992). Intraindividual changes in test recall of the elderly. Brain and Language, 42, 248-269.

Howard, D. V. (1988). Aging and memory activation: The priming of semantic and episodic memories. In L. L.

Light & D. M. Burke (Eds.), Language, Memory and Aging (pp. 77-99). Cambridge, MA: Cambridge University Press.

Howard, D. V., Shaw, R. J., & Heisey, J. G. (1986). Aging and the time course of semantic activation. Journal of Gerontology, 43(2), 195-203.

Kausler, D. H. (1982). Experimental psychology and human aging. New York: Wiley.

Kucera, H., & Francis, W. N. (1967). Computational analysis of present-day English. Providence, RI: Brown University Press.

Lawton, M. P., Kleban, M. H., Dean, J., & Rajagopal, D. (1992). The factorial generality of brief positive and negative affect measures. Journals of Gerontology, 47(4), 228-237.

Mantyla, T. & Craik, F. I. (1993). Context sensitivity and adult age differences in encoding variability. European Journal of Cognitive Psychology, 5(3), 319-336.

Morse, C. K. (1993). Does variability increase with age? An archival study of cognitive measures. Psychology and Aging, 8(2), 156-164.

Morton, J. (1969). Interaction of information in word recognition. Psychological Review, 76, 165-178.

Nelson, E. A., & Dannifer, D. (1992). Aged heterogeneity: Fact or fiction? The fate of diversity in gerontological research. The Gerontologist, 32(1), 17-23.

Nichelli, P., Vernneri, A., Molinari, M., & Tavani, F. (1993). Precision and accuracy of subjective time estimation in different memory disorders. Cognitive Brain Research, 1(2), 87-93.

North, A. D., & Fairchild, M. D. (1993). Measuring color-matching functions II. New data for assessing observer metamerism. Color Research and Application, 18(3), 163-170.

Patterson, V. H., Foster, D. H., & Heron, J. R. (1980). Variability of visual threshold in multiple sclerosis: Effect of background luminance on frequency of seeing. Brain, 103, 139-147.

Plotkin, H. C., & Russell, I. S. (1972). Compensation of a hemidecorticate learning deficit by manipulation of input redundancy. Physiology and Behavior, 10, 601-606.

Raven, J. C. (1958). Guide to using the Mill Hill vocabulary test with the progressive matrices scale (2nd Ed.), London: H. K. Lewis.

Salthouse, T. A. (1985). Speed of behaviour and its implications for cognition. In J. E. Birren & K. W. Schaie (Eds.), Handbook of the psychology of aging, (2nd Ed.), New York: Van Nostrand Reinhold.

Scukanec, G. P., Petrosino, L., & Rastatter, M. P. (1992). Fundamental frequency variability in elderly women during production of stressed and unstressed words. Perceptual and Motor Skills, 74(3), 1091-1095.

Slawinski, E. B., Hartel, D. M., & Kline, D. W. (1993). Self-reported hearing problems in daily life throughout adulthood. Psychology and Aging, 8(4), 552-561.

Stuss, D. T., Pogue, J., Buckle, L., & Bondar, J. (1994). Characterization of stability of performance in patients with traumatic brain injury: Variability and consistency on reaction time tests. Nueropsychology, 8, 316-324.

Swanson, L. R., & Lee, T. D. (1992). Effects of aging and schedules of knowledge of results on motor learning. Journals of Gerontology, 47(6), 406-411.

Wauquier, A. (1993). Aging and changes in phasic events during sleep. Physiology and Behavior, 54(4), 803-806.

Wilkinson, R. T., & Allison, S. (1989). Age and simple reaction time: Decade differences for 5,325 subjects. Journal of Gerontology, 44(2), 29-35.

APPENDIX A

Stimulus Materials

High Frequency Word List

find	should	white	more	that
water	however	better	perhaps	felt
power	life	thought	even	come
present	such	good	without	want
against	even	from	enough	service
they	before	social	will	around
school	example	city	could	little
would	toward	very	history	what
three	never	public	long	certain
years	been	system	order	second
this	money	house	almost	were
church	action	your	about	point
great	with	between	matter	place
said	young	began	their	look
light	right	also	group	human
night	these	side	state	need
which	always	under	want	because
still	like	sense	back	found
become	began	used	every	away
nothing	family	those	face	another
himself	course	time	small	where
mind	number	most	within	during

just	after	have	think	people
above	problem	other	does	head
night	work	either	thing	first
much	give	there	among	whole
early	when	since	while	both
least	across	over	through	again

Nonwords

quims	tawl	cofend	fanfark	kano
suep	yertz	beed	togu	fuam
joygul	gask	vire	ketchud	hobi
gred	ebsic	winower	yondes	quirb

Low Frequency Word List

mentor	rift	almond	kebob	cask
plasm	envious	thicket	feud	deport
voids	wicket	heel	oxygen	lathe
zombie	usurp	bold	expire	noun
inca	jovial	bison	yolk	felony
majesty	dank	ravine	xenon	annex
paddle	malt	vaccine	snail	avid
beige	quartz	newt	wand	united
dire	eject	khaki	moan	faucet
lurk	yelped	chalky	armory	gloss
quibble	iota	kinder	reigns	clap
yodel	waltz	ascend	oafs	oblique
hash	drowsy	jest	quench	fawn
yoga	torpedo	jade	pews	racquet
deem	zulu	trauma	eave	plum
naughty	glib	coyote	sequel	waive
aghast	oaks	mailbox	levee	yaks
migrate	icing	loser	fads	zipper
veal	irate	outwit	hybrid	cadaver
pout	nope	sausage	epsilon	nigh
wiener	umpire	moot	garish	robot

mangled	lads	swanky	valiant	yearn
haggle	lava	bribe	ritz	nullify
zoned	niece	thug	digress	gambles
uncap	chum	humid	ornate	ferret
jock	idolize	kazoo	rhyme	infer
primate	binge	vise	vocally	tuba
infidel	pickle	enchant	lapel	jostle

ds

scudy	koday		bory	spall
frew	casp		furm	walf
mojent	keup	blought	dorsh	sune
weet	kinc	bast	bejind	tond

APPENDIX B

Instructions to Subjects

Instructions

Today you will be performing three separate tasks, each of which will be explained to you as we come to them. Before we begin, I'll remind you that your anonymity will be protected and any information gathered about you will be coded into the computer by subject number only. Your participation is voluntary, and if at any time during the experiment you feel like quitting, you may do so without penalty.

Lexical Decision Task

In the next task, you will be performing lexical decisions; which simply means reading a string of letters and deciding whether that string is a word or not a word. Each trial will begin with a short warning beep, and a box will appear in the middle of the computer screen that indicates where you should direct your attention. Then a list of four letter strings will appear inside the box. Your job will be to read through that list and decide whether all of the letter strings are words, or if there is a nonword present. You will position the index fingers of both hands over the blue and red keys throughout the experiment. If all of the letter strings are words you will push the blue key with your left hand.

If you encounter a nonword in the list, you will push the red key with your right index finger. Note that on trials that include nonwords, the nonword may appear in any one of the four letter string positions. So if a nonword is located in the first position for example, you won't need to read through the rest of the list; simply make your response as soon as you see a nonword.

I will be measuring your response times and your errors, so I want you to make your responses as quickly and as accurately as possible. There will be 80 experimental trials in this task, and each new trial will begin soon after you make a response. One half of the trials will contain nonwords, and the other half will be lists of four words. The word/nonword trials will be presented to you randomly. If you feel that you've made an error on a trial, try to resist correcting it by pressing the other key, as that keypress will be read into the computer as your response for the next trial. If this does happen, don't panic. You'll see the next trial skip by, then things will return to their normal pace.

Before we begin the experimental trials, I'll give you 10 practice trials so that you can get used to the controls and the procedure. Do you have any questions before we begin the practice session?

Line Length Estimation Task

In this next task, you will be required to estimate the length of a vertically presented line by building a horizontal line to match its length. On each trial, a vertical line will appear somewhere on the top half of the computer screen, along with a starting segment of a horizontal line near the bottom of the screen. You will adjust the length of the horizontal line until you think it matches the vertical line in length. Tapping the red key with you right index finger will increase the length of the line to the right, and tapping the blue key with your left index finger will shorten the line. When you are happy with your estimate, tap the spacebar to indicate that you have finished, and the next trial will begin.

On one half of the trials, the vertical line will only stay on screen for a few seconds, then disappear from view before you begin your estimation. You'll have to remember the length of that line for your estimate, so be sure you get a good look at it before responding. There will be 40 trials in the experimental condition, and the hard and easy trial types will be presented randomly.

I'll give you 10 practice trials before we begin the experimental trials so that you can familiarize yourself with the controls and the procedure. Do you have any questions before we begin?

Time Estimation Task

The following task is a time estimation task in which you will be required to estimate the amount of time that passes between two beep-tones. You will make your estimate by replicating the two beep-tones simply by tapping the spacebar once to create the beginning beep, and once again to replicate the end beep - as if you were controlling the button on a stop-watch. You will be presented with 20 easy, trials, and 20 harder, or simple tracking trials. I will present these blocks of trials separately, so you won't have to worry about which type of trial will be coming up next.

Easy Condition: Each trial will begin with a long beep-tone, along with a message on the screen telling you to begin your estimate. After a period of time, a shorter beep will sound along with a message telling you to stop estimating the time. Now you will estimate that time period by tapping the spacebar to create that first, long beep-tone. A message will appear on the screen asking you to tap the spacebar again to end your estimate. It will remain on the screen until you have tapped the spacebar to indicate the end of your estimate. The next trial will begin on after you've made your estimate.

Simple Tracking Condition: Trials in this condition will be identical to those in the easy condition with one exception. After the computer presents you with the long starting beep-tone, numbers will be flashed in the middle of the computer screen one at a time until the end tone along with the message telling you to stop estimating time. All you have to do is read the numbers aloud as you see them. After the stop tone, your job will be the exact same as it was in the easy condition. You will make your estimate the same way as in the easier trials. Don't worry about the numbers you read. You are not required to remember them. I'm only interested in your time estimates.

Before we move on to the experimental trials, I will give you 5 practice trials of each type to get used to the controls and procedure. Do you have any questions before we begin the practice session?

Appendix C

Computer Programs

LEXICAL DECISION TASK

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <graph.h>
#include "newtime.h"

#define MAX_TRIALS  512

/* Constants for ioboard operation */
#define IBFA         0x20
#define WIN1_BUTTON      0x01
#define WIN2_BUTTON      0x02
#define READY_BUTTON     0x04

struct _videoconfig vc;
struct _fontinfo fi;

/* Parameters are reset in .par file. These are suggested
values. */
int COLOR=14;
int box = 1;
int initbeep=1;
int beepfreq=500;
int beeplength=500;
unsigned long ISI=1500;
char options[20]={"courier\0"};
char font_size[20]= {"h40w32b\0"};
char firstmess[80]= {"Press Space Bar to begin...\0"};
char  lastmess[80]=  {"This  sequence  is  complete.  Thank
you.\0"};

/* The following data structure defines an experiment trial */
typedef struct {
     int trial_number;
   char word1[8];
   char word2[8];
   char word3[8];
   char word4[8];
   char correct_response;
   } expt_trial_type ;

/*  The  following  data  type  defines  responses  to  an
experimental trial */
typedef struct {
     char actual_response;
     int time_in_millisecs;
     }expt_response_type;

#include "nsleep.c"
```

```
#include "io.c"
#include "initgr.c"

void message(char message_text[])
{
     short x,y;

     _clearscreen( _GCLEARSCREEN );
    _x=(vc.numxpixels/2)-(_getgtextextent(message_text)/2);
    y=(vc.numypixels/2);
    _moveto(x,y);
    _outgtext(message_text);
}

void show_words(expt_trial_type trial)
{
     short x, y;

     x=(vc.numxpixels/2)-(_getgtextextent(trial.word1)/2);
    y=(vc.numypixels/2)-(4*fi.ascent);
     _moveto(x,y); _outgtext( trial.word1);
    _moveto(x,y+fi.ascent ); _outgtext( trial.word2);
    _moveto(x,y+2*fi.ascent ); _outgtext( trial.word3);
    _moveto(x,y+3*fi.ascent ); _outgtext( trial.word4);
}

void present_trial( expt_trial_type  trial,int  *timing,char
*response)
{
     _clearscreen( _GCLEARSCREEN );
    sleep(ISI);
    if (box) _rectangle(_GBORDER, 260,150,380, 280);
    if (initbeep) beep(beepfreq,beeplength);
    _clearscreen( _GCLEARSCREEN );
    sleep(500);

     show_words(trial);
     ticks=0;
    *response=getch();
    *timing= (int) ticks;
}

void main(int argc,char *argv[])
{
     /* defaults that can be changed with command line flags
*/
   char *experiment_fname = "lexprac.txt";        /* default
name for experiment trial definition file */
   char *results_fname = "lexprac.dat";      /* default name
for results data file */
   char *par_fname="lex.par";                          /*default
name for parameter file */
```

```
int i, num_trials;

FILE *expt_file_ptr, *results_file_ptr;
expt_trial_type trial_array[MAX_TRIALS];
expt_response_type response_array[MAX_TRIALS];

/* Change defaults from command line */
for (i=1; i < argc; i++) {
 if (!_stricmp(_strlwr(argv[i]), "-p")) par_fname =
argv[++i];
 else if (!_stricmp(_strlwr(argv[i]), "-ef"))
experiment_fname = argv[++i];
  else if (!_stricmp(_strlwr(argv[i]), "-of"))
results_fname = argv[++i];
  else {
     printf("Command line contains invalid flag %s
!\n",argv[i]);
     exit(1);
 }
}

 read_parameters(par_fname);

if ((expt_file_ptr = fopen(experiment_fname,"r")) ==
NULL) {
 printf("Experiment definition file -> %s does not exist
!\n",experiment_fname);
 exit(1);
}

num_trials=read_experiment(expt_file_ptr,trial_array);

fclose(expt_file_ptr);

if ((results_file_ptr = fopen(results_fname,"a")) == NULL)
{
 printf("Cannot open results file: %s \n",results_fname);
 exit(1);
}

prepare_font();
prepare_screen();
start_hires();

message(firstmess);
_getch();


for (i=1;i<=num_trials;i++)
   present_trial(trial_array[i],
```

```
&response_array[i].time_in_millisecs,

&response_array[i].actual_response);

      record_experiment(results_file_ptr,
trial_array,response_array,num_trials);
     fclose(results_file_ptr);

     message(lastmess);
     sleep(5000);

     stop_hires();
     _setvideomode( _DEFAULTMODE );
}

void read_parameters (char par_fname[])
{
     FILE *par;

      if ((par = fopen(par_fname,"r")) == NULL) {
      printf("Parameter   file   ->   %s   does   not   exist
!\n",par_fname);
      exit(1);
     }
     fscanf(par,"%d",&COLOR);
     fscanf(par,"%d",&box);
     fscanf(par,"%d",&initbeep);
     fscanf(par,"%d",&beepfreq);
     fscanf(par,"%d",&beeplength);
     fscanf(par,"%lu\n",&ISI);                    /*
while(fgetc(par) != '\n'); */
     fscanf(par,"%[^\n]\n",options);
     fscanf(par,"%[^\n]\n",font_size);
     fscanf(par,"%[^\n]\n",firstmess);
     fscanf(par,"%[^\n]\n",lastmess);
     fclose(par);
}

int read_experiment(
                                                   F   I   L   E
*exp_file_ptr,
                                                   expt_trial_type
trial_array[]  )
{
   int num_trials, scan_return=0;

    for   (num_trials=1;scan_return!=EOF   &&
num_trials<=MAX_TRIALS;num_trials++){
      scan_return = fscanf(exp_file_ptr, "TRIAL: %d WORDS:
%[A-Z] %[A-Z] %[A-Z] %[A-Z] ANSW: %c\n",
               &trial_array[num_trials].trial_number,
               &trial_array[num_trials].word1,
```

```
                &trial_array[num_trials].word2,
                &trial_array[num_trials].word3,
                &trial_array[num_trials].word4,
                &trial_array[num_trials].correct_response);
     }
     return(num_trials-2);
}

void record_experiment(
                                                            F I L E
*results_file_ptr,

expt_trial_type trial_array[],

expt_response_type response_array[],
                                                            i   n   t
num_trials                             )
{
     int i;
     for( i=1; i<=num_trials;i++){
          fprintf(results_file_ptr, "%d %c TIME: %d RESP:
%c\n",
                         trial_array[i].trial_number,
                         trial_array[i].correct_response,
                         response_array[i].time_in_millisecs,
                         response_array[i].actual_response);
                    }

/* Pauses for a specified number of microseconds. Uses
newtime.h */
void sleep( unsigned long  wait )
{
    unsigned long goal;

    goal = wait + ticks;
    while( goal > ticks)
        ;
}

void beep( int frequency, unsigned long duration )
{
    int control;

    /* If frequency is 0, Beep doesn't try to make a sound. It
     * just sleeps for the duration.
     */
    if( frequency )
    {
        /* 75 is about the shortest reliable duration of a
sound. */
        if( duration < 75 )
            duration = 75;
```

```
        /* Prepare timer by sending 10111100 to port 43. */
        _outp( 0x43, 0xb6 );

        /* Divide input frequency by timer ticks per second
and
         * write (byte by byte) to timer.
         */
        frequency = (unsigned)(1193180L / frequency);
        _outp( 0x42, (char)frequency );
        _outp( 0x42, (char)(frequency >> 8) );

        /* Save speaker control byte. */
        control = inp( 0x61 );

        /* Turn on the speaker (with bits 0 and 1). */
        _outp( 0x61, control | 0x3 );
    }

    sleep( duration );

    /* Turn speaker back on if necessary. */
    if( frequency )
        _outp( 0x61, control );
}

void prepare_font()
{
    unsigned char list[20];

    /* Read header info from .FON files in current directory.
*/
    if( _registerfonts( "*.FON" ) <= 0 ){
        _outtext( "Error: Can't register fonts" );
         exit( 1 );
    }

    /* Build options string. */
    strcat( strcat( strcpy( list, "t'" ), options), "'");
    strcat( list, font_size );

    if( _setfont( list ) >= 0 ){
         if( _getfontinfo( &fi ) ){
              _outtext( "Error: Can't get font information" );
              exit(1);
            }
     }
     else {
          _outtext( "Error: Can't set font ");
          _outtext(list);
          exit(1);
     }
}
```

```
void prepare_screen()
{
    /* Set graphics mode and get configuration. */
    if( !_setvideomode( _MAXRESMODE )){
      _outtext("Graphics Mode unavailable");
      exit( 1 );
    }
    _getvideoconfig( &vc );
     if( vc.numcolors > 2 ) _setcolor( COLOR);
}

/* Using C/C++ p. 828 */
/*Settings for 1000/sec */

#include <stdlib.h>
#include <stdio.h>
#include <dos.h>
#include <conio.h>

#define ICR 0x20
#define EOI 0x20

unsigned long ticks;
unsigned long begin_time;
unsigned long end_time;
unsigned long far *clock = (unsigned long far *)0x0040006CUL;
unsigned long far *oflow = (unsigned long far *)0x00400070UL;
unsigned long save_clock;
unsigned long save_oflow;

void (interrupt far *oldint8) ();

void interrupt far newint8 ()
{
     static unsigned int count = 55;

     ++ticks;
     if (--count == 0 ) {
          (*oldint8)();
          count = 55;
     }
     else outp( ICR, EOI );
}

static void hires_clock (void)
{
     _disable();
     save_clock = *clock;
     save_oflow = *oflow;
     ticks = 0;
     oldint8 = _dos_getvect (8);
     _dos_setvect(8, newint8);
```

```
     outp( 0x43, 0x36);
     outp( 0x40, 0xa9);
     outp( 0x40, 0x04);
     _enable();
}

static void lores_clock( void )
{
     _disable();
     outp( 0x43, 0x36 );
     outp( 0x40, 0 );
     outp( 0x40, 0 );
     _dos_setvect( 8, oldint8 );
     *oflow = save_oflow;
     *clock = save_clock + (end_time -begin_time ) / 55UL;
     _enable();
}

void start_hires(void)
{
     hires_clock();
     begin_time = ticks;
}

void stop_hires(void)
{
     end_time = ticks;
     lores_clock();
}

double hires_duration( void )

     return ( ( (double) ( end_time - begin_time ) / 1000.0)
);
}
```

LINE LENGTH ESTIMATION TASK

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <graph.h>
#include <math.h>
#include <time.h>

/* The following data structure defines an experiment trial */
typedef struct {
   int trial_number;
   int color;
   int leave_on;
```

```
   int length;
   int start_with;
   int add;
   } expt_trial_type ;

/*  The  following  data  type  defines  responses  to  an
experimental trial */
typedef struct {
     int final_setting;
     int moves;
     } expt_response_type;

#define MAX_TRIALS 128

struct _videoconfig vc;
struct _fontinfo fi;

int CENTREX= 320 ;
int  CENTREY= 240 ;

int COLOR=14;
unsigned char *options={"courier"};
unsigned char *font_size= {"h40w32b"};

unsigned char *firstmess= {"Press Space Bar to begin..."};
unsigned char *lastmess= {"This sequence is complete. Thank
you."};

#include "io.c"

/* Pauses for a specified number of microseconds. */
void sleep( clock_t wait )
{
    clock_t goal;

    goal = wait + clock();
    while( goal > clock() )
        ;
}


void prepare_font()
{
    unsigned char list[20];

    /* Read header info from .FON files in current directory.
*/
    if( _registerfonts( "*.FON" ) <= 0 ){
        _outtext( "Error: can't register fonts" );
          exit( 1 );
    }
```

```
    /* Build options string. */
    strcat( strcat( strcpy( list, "t'" ), options), "'");
    strcat( list, font_size );

    if( _setfont( list ) >= 0 ){
          if( _getfontinfo( &fi ) ){
             _outtext( "Error: Can't get font information" );
             exit(1);
            }
     }
     else {
          _outtext( "Error: Can't set font");
          _outtext(list);
          exit(1);
     }
}

void prepare_screen()
{
    /* Set graphics mode and get configuration. */
    if( !_setvideomode( _MAXRESMODE )){
      _outtext("Graphics Mode unavailable");
      exit( 1 );
    }
    _getvideoconfig( &vc );
    if( vc.numcolors > 2 ) _setcolor( COLO
}


void message(char message_text[])
{
     short x,y;

    _clearscreen( _GCLEARSCREEN );
    x=(vc.numxpixels/2)-(_getgtextextent(message_text)/2);
    y=(vc.numypixels/2);
    _moveto(x,y);
    _outgtext(message_text);
}

void addline(int i,int start, int add)
{
     int x,y;
    x=CENTREX/2+start+((i-1)*add);
    y=17*CENTREY/10;

     _moveto(x,y);
     _lineto(x+add,y);
}

void subtractline(int i,int start,int add)
{
```

```
int x,y;

    x=CENTREX/2+start+((i+1)*add);
    y=17*CENTREY/10;
    _moveto(x,y);
    _setcolor(0);
    _lineto(x-add,y);
    _setcolor(COLOR);
}

void jiggle_line_placement()

    int jigx

    jigx=(int) (
                    (
                        (
                         (float)(rand())
                                                            /

(float)(RAND_MAX)

                                                *80.0F

                                                                  )

-40.0F
                    )
                  );
      jigy=(int) ((((float)(rand())/(float)(RAND_MAX)*60.0F)
-30.0F));
     CENTREX=320+jigx;
     CENTREY=240+jigy;
}

void frame(expt_trial_type trial)
{

     jiggle_line_placement();

      _moveto( CENTREX,CENTREY-140);
     _lineto(CENTREX,CENTREY-140+trial.length);

     if (trial.leave_on>0){
          sleep((clock_t)trial.leave_on);
           _setcolor(0);
          _moveto(CENTREX,CENTREY-140);
           _lineto(CENTREX,CENTREY-140+trial.length);
           _setcolor(COLOR);
     }

    jiggle_line_placement();
```

```
     _moveto(CENTREX/2,17*CENTREY/10);
     _lineto(CENTREX/2+trial.start_with,17*CENTREY/10);
}

void rtrial( expt_trial_type trial, int *setting, int *trials)
{
          char dir=0;
          int i=0,n=0;
          int max;

          COLOR=trial.color;
          _setcolor(COLOR);
           _clearscreen( _GCLEARSCREEN );
        max=3*(trial.length/trial.add)/2;

        frame(trial);

        while (dir != ' ' ) {
               dir = _getch(); n++;
          if (dir == '/') {
                 i++;
                if (i > max) i = max;
                else addline(i,trial.start_with,trial.add);
          }
            else  if (dir == 'z' ) {
               i--;
                if (i < 0) i = 0;
                                                       e  l  s  e
subtractline(i,trial.start_with,trial.add);

               }
        }
          _clearscreen( _GCLEARSCREEN );
          *setting=i;
          *trials=n;
}

void main(int argc,char *argv[])
{
     /* defaults that can be changed with command line flags
*/
   char *experiment_fname = "lprac.txt";          /* default
name for experiment trial definition file */
   char *results_fname = "lprac.dat";        /* default name
for results data file */

     int i, num_trials;

     FILE *expt_file_ptr, *results_file_ptr;
     expt_trial_type trial_array[MAX_TRIALS];
     expt_response_type response_array[MAX_TRIALS];
```

```
    /* Change defaults from command line */
    for (i=1; i < argc; i++) {
      if    (!_stricmp(_strlwr(argv[i]),    "-ef"))
experiment_fname = argv[++i];
      else    if    (!_stricmp(_strlwr(argv[i]),    "-of"))
results_fname = argv[++i];
      else {
         printf("Command  line  contains  invalid  flag  %s
!\n",argv[i]);
         exit(1);
      }
   }

   if ((expt_file_ptr = fopen(experiment_fname,"r")) == NULL)
{
     printf("Experiment definition file -> %s does not exist
!\n",experiment_fname);
     exit(1);
   }

    num_trials=read_experiment(expt_file_ptr,trial_array);

    fclose(expt_file_ptr);

   if ((results_file_ptr = fopen(results_fname,"a")) == NULL)
{
     printf("Cannot open results file: %s \n",results_fname);
     exit(1);
   }

    srand(10128U);
    prepare_font();
    prepare_screen();

    message(firstmess);
    _getch();

   for (i=1;i<=num_trials;i++){

      rtrial(trial_array[i],
             &response_array[i].final_setting,
             &response_array[i].moves);

      }

     record_experiment(results_file_ptr,
trial_array,response_array,num_trials);
    fclose(results_file_ptr);

    message(lastmess);
    sleep((clock_t)5000);
```

```
     _setvideomode( _DEFAULTMODE );

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <graph.h>
#include <math.h>
#include <time.h>

/* The following data structure defines an experiment trial */
typedef struct {
   int trial_number;
   int color;
   int leave_on;
   int length;
   int start_with;
   int ada;
   } expt_trial_type ;

/*  The  following  data  type  defines  responses  to  an
experimental trial */
typedef struct {
     int final_setting;
     int moves;
     } expt_response_type;

#define MAX_TRIALS 128

struct _videoconfig vc;
struct _fontinfo fi;

int CENTREX= 320 ;
int  CENTREY= 240 ;

int COLOR=14;
unsigned char *options={"courier"};
unsigned char *font_size= {"h40w32b"};

unsigned char *firstmess= {"Press Space Bar to begin..."};
unsigned char *lastmess= {"This sequence is complete. Thank
you."};

#include "io.c"

/* Pauses for a specified number of microseconds. */
void sleep( clock_t wait )
{
    clock_t goal;

    goal = wait + clock();
```

```
    while( goal > clock() )
        ;
}

void prepare_font()
{
    unsigned char list[20];

    /* Read header info from .FON files in current directory.
*/
    if( _registerfonts( "*.FON" ) <= 0 ){
        _outtext( "Error: can't register fonts" );
         exit( 1 );
    }

    /* Build options string. */
    strcat( strcat( strcpy( list, "t'" ), options), "'");
    strcat( list, font_size );

    if( _setfont( list ) >= 0 ){
          if( _getfontinfo( &fi ) ){
             _outtext( "Error: Can't get font information" );
             exit(1);
            }
     }
     else {
          _outtext( "Error: Can't set font");
          _outtext(list);
          exit(1);
     }
}

void prepare_screen()
{
    /* Set graphics mode and get configuration. */
    if( !_setvideomode( _MAXRESMODE )){
      _outtext("Graphics Mode unavailable");
      exit( 1 );
    }
    _getvideoconfig( &vc );
     if( vc.numcolors > 2 ) _setcolor( COLOR);
}


void message(char message_text[])
{
     short x,y;

    _clearscreen( _GCLEARSCREEN );
     x=(vc.numxpixels/2)-(_getgtextextent(message_text)/2);
    y=(vc.numypixels/2);
    _moveto(x,y);
```

```
    _outgtext(message_text);
}

void addline(int i,int start, int add)
{
     int x,y;
    x=CENTREX/2+start+((i-1)*add);
    y=17*CENTREY/10;

     _moveto(x,y);
     _lineto(x+add,y);
}

void subtractline(int i,int start,int add)
{
 int x,y;

     x=CENTREX/2+start+((i+1)*add);
    y=17*CENTREY/10;
     _moveto(x,y);
     _setcolor(0);
     _lineto(x-add,y);
      _setcolor(COLOR);
}

void jiggle_line_placement()

    int jigx,jigy;


    jigx=(int) (
                      (
                          (
                           (float)(rand())
                                                                   /

(float)(RAND_MAX)

                                                       *80.0F

                                                                        )
-40.0F
                     )
                    );
      jigy=(int) ((((float)(rand())/(float)(RAND_MAX)*60.0F)
-30.0F));
     CENTREX=320+jigx;
     CENTREY=240+jigy;
}

void frame(expt_trial_type trial)
{
```

```
    jiggle_line_placement();

     _moveto( CENTREX,CENTREY-140);
    _lineto(CENTREX,CENTREY-140+trial.length);

    if (trial.leave_on>0){
         sleep((clock_t)trial.leave_on);
         _setcolor(0);
         _moveto(CENTREX,CENTREY-140);
         _lineto(CENTREX,CENTREY-140+trial.length);
         _setcolor(COLOR);
    }

   jiggle_line_placement();

     _moveto(CENTREX/2,17*CENTREY/10);
     _lineto(CENTREX/2+trial.start_with,17*CENTREY/10);
}

void rtrial( expt_trial_type trial, int *setting, int *trials)
{
          char dir=0;
          int i=0,n=0;
          int max;

          COLOR=trial.color;
          _setcolor(COLOR);
          _clearscreen( _GCLEARSCREEN );
        max=3*(trial.length/trial.add)/2;

        frame(trial);

        while (dir != '   ') {
               dir = _getch(); n++;
          if (dir == '/') {
                 i++;
                if (i > max) i = max;
                else addline(i,trial.start_with,trial.add);
          }
            else  if (dir == 'z' ) {
                i--;
                 if (i < 0) i = 0;
                                                  e   l   s   e
subtractline(i,trial.start_with,trial.add);

                }
        }
          _clearscreen( _GCLEARSCREEN );
          *setting=i;
          *trials=n;
}
```

```
void main(int argc,char *argv[])
{
     /* defaults that can be changed with command line flags
*/
   char *experiment_fname = "lprac.txt";          /* default
name for experiment trial definition file */
   char *results_fname = "lprac.dat";          /* default name
for results data file */

     int i, num_trials;

     FILE *expt_file_ptr, *results_file_ptr;
     expt_trial_type trial_array[MAX_TRIALS];
     expt_response_type response_array[MAX_TRIALS];

     /* Change defaults from command line */
     for (i=1; i < argc; i++) {
       if    (!_stricmp(_strlwr(argv[i]),    "-ef"))
experiment_fname = argv[++i];
       else    if    (!_stricmp(_strlwr(argv[i]),    "-of")
results_fname = argv[++i];
       else {
          printf("Command  line  contains  invalid  flag  %s
!\n",argv[i]);
          exit(1);
      }
   }

    if ((expt_file_ptr = fopen(experiment_fname,"r")) == NULL)
{
      printf("Experiment definition file -> %s does not exist
!\n",experiment_fname);
      exit(1);
   }

     num_trials=read_experiment(expt_file_ptr,trial_array);

     fclose(expt_file_ptr);

    if ((results_file_ptr = fopen(results_fname,"a")) == NULL)
{
      printf("Cannot open results file: %s \n",results_fname);
      exit(1);
   }

     srand(10128U);
     prepare_font();
     prepare_screen();

     message(firstmess);
     _getch();
```

```
for (i=1;i<=num_trials;i++){

   rtrial(trial_array[i],
          &response_array[i].final_setting,
          &response_array[i].moves);

    }

  record_experiment(results_file_ptr,
trial_array,response_array,num_trials);
  fclose(results_file_ptr);

  message(lastmess);
  sleep((clock_t)5000);

  _setvideomode( _DEFAULTMODE );
```

TIME ESTIMATION TASK

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <graph.h>
#include <bios.h>
#include "newtime.h"

#define MAX_TRIALS 128

struct _videoconfig vc;
struct _fontinfo fi;

/* Parameters are reset in .par file. These are suggested
values. */
int COLOR=14;
int box = 1;
int beepfreq=500;
int beeplength=500;
char options[20]={"courier\0"};
char font_size[20]= {"h40w32b\0"};
char firstmess[80]= {"Press Space Bar to begin...\0"};
char lastmess[80]= {"This sequence is complete. Thank
you.\0"};

/* The following data structure defines an experiment trial */
typedef struct {
     int trial_number;
     unsigned long trial_length;
   unsigned long isi;
   int task_type;
```

```
   int digits_type;
   char numbers[50];
   } expt_trial_type ;

/*  The  following  data  type  defines  responses  to  an
experimental trial */
typedef struct {
     unsigned long actual_time;
     unsigned long time_in_millisecs;
     }expt_response_type;

#include "nsleep.c"
#include "io.c"
#include "initgr.c"

void message(char message_text[])
{
     short x,y;

      while    (_bios_keybrd(_KEYBRD_READY))
_bios_keybrd(_KEYBRD_READ);
    _clearscreen( _GCLEARSCREEN );
    x=(vc.numxpixels/2)-(_getgtextextent(message_text)/2);
    y=(vc.numypixels/2);
    _moveto(x,y);
    _outgtext(message_text);
}

void show_numbers(expt_trial_type trial,int i)
{
     short x, y;
    char word[7] ={"      \0" ;

     x=(vc.numxpixels/2)-6*(_getgtextextent("0")/2);
    y=(vc.numypixels/2)-(fi.ascent);
     _moveto(x,y);

     if (trial.task_type == 1){
          if    (trial.digits_type    ==    1)
word[3]=trial.numbers[i];
          else   if   (trial.digits_type   ==   2)   word[3]
=trial.numbers[i], word[4]=trial.numbers[i+1];
          else   if   (trial.digits_type   ==   2)   word[3]
= t r i a l . n u m b e r s [ i ] ,
word[4]=trial.numbers[i+1],word[5]=trial.numbers[i+2];
    }
    else if (trial.task_type == 2) {
     if (trial.digits_type == 1)  word[0]=trial.numbers[i],
word[5]=trial.numbers[i+1];
        else if (trial.digits_type == 2) {
          word[0]=trial.numbers[i];
          word[1]=trial.numbers[i+1];
```

```
            word[4]=trial.numbers[i+2];
            word[5]=trial.numbers[i+3];
        }
    }
      _outgtext(word);
}

void present_trial(expt_trial_type trial, int i)
{

     float multiplier;
     _clearscreen( _GCLEARSCREEN );

    if (box && trial.task_type!=0) {
          _rectangle(_GBORDER, 260,200,380,270);
        sleep(500);
    }
     _clearscreen( _GCLEARSCREEN );
      if (trial.task_type!=0) {
          show_numbers(trial,i);
      }
     /* getch(); */
     multiplier=  1+   (((   (float)rand()/(float)   RAND_MAX)
-0.50)*0.50);
     sleep((unsigned long) (multiplier*(float) trial.isi) );

}

void get_estimate(unsigned long *millisecs)
{
     while   (_bios_keybrd(_KEYBRD_READY))
_bios_keybrd(_KEYBRD_READ);
     message("Press Spacebar to begin time estimate...");
     _getch();

     beep(beepfreq,beeplength);
     message("Press Spacebar to end time estimate...");
     ticks=0L;
     _getch();
     *millisecs = ticks;
      _clearscreen( _GCLEARSCREEN );
}


void main(int argc,char *argv[])
{
     /* defaults that can be changed with command line flags
*/
   char *experiment_fname = "estprac.txt";       /* default
name for experiment trial definition file */
   char *results_fname = "estprac.dat";      /* default name
for results data file */
```

```
   char *par_fname="est.par";                            /*default
name for parameter file */

     int i, num_trials,j;

     FILE *expt_file_ptr, *results_file_ptr;
     expt_trial_type trial_array[MAX_TRIALS];
     expt_response_type response_array[MAX_TRIALS];

     /* Change defaults from command line */
     for (i=1; i < argc; i++) {
      if  (!_stricmp(_strlwr(argv[i]),  "-p"))  par_fname  =
argv[++i];
      else      if    (!_stricmp(_strlwr(argv[i]),    "-ef"))
experiment_fname = argv[++i];
       else    if    (!_stricmp(_strlwr(argv[i]),    "-of"))
results_fname = argv[++i];
       else {
         printf("Command  line  contains  invalid  flag  %s
!\n",argv[i]);
         exit(1);
      }
   }

      read_parameters(par_fname);

     if  ((expt_file_ptr  =  fopen(experiment_fname,"r"))  ==
NULL) {
      printf("Experiment definition file -> %s does not exist
!\n",experiment_fname);
      exit(1);
     }

       m_trials=read_experiment(expt_file_ptr,trial_array);

     fclose(expt_file_ptr);

    if ((results_file_ptr = fopen(results_fname,"a")) == NULL)
{
      printf("Cannot open results file: %s \n",results_fname);
      exit(1);
     }

    prepare_font();
     prepare_screen();
    start_hires();

     message(firstmess);
     _getch();
     _clearscreen( _GCLEARSCREEN );
     sleep(500);
```

```
    for (i=1;i<=num_trials;i++)  {
        message("Estimate time from beep end to next
beep...");
        sleep(beeplength);
          beep(beepfreq,beeplength);
           _clearscreen( _GCLEARSCREEN );
        ticks=0L;j=0;
        while (ticks<trial_array[i].trial_length) {
            present_trial(trial_array[i],j);

j=j+(trial_array[i].task_type*trial_array[i].digits_type);
        }
        message("Stop estimating time...");
        beep(beepfreq,beeplength/2);
         response_array[i].actual_time = ticks;
           _clearscreen( _GCLEARSCREEN );
        sleep(1000);
        get_estimate(&response_array[i].time_in_millisecs);
    }

      record_experiment(results_file_ptr,
trial_array,response_array, num_trials);
     fclose(results_file_ptr);

     message(lastmess);
     sleep(5000);

     stop_hires();
     _setvideomode( _DEFAULTMODE );
}

void read_parameters (char par_fname[])
{
     FILE *par;

      if ((par = fopen(par_fname,"r")) == NULL) {
      printf("Parameter file -> %s does not exist
!\n",par_fname);
      exit(1);
     }
     fscanf(par,"%d",&COLOR);
     fscanf(par,"%d",&box);
     fscanf(par,"%d",&beepfreq);
     fscanf(par,"%d",&beeplength);
     fscanf(par,"%[^\n]\n",options);
     fscanf(par,"%[^\n]\n",font_size);
     fscanf(par,"%[^\n]\n",firstmess);
     fscanf(par,"%[^\n]\n",lastmess);
     fclose(par);
}
```

```
int read_experiment(                                  F    I    L    E
*exp_file_ptr,                                        expt_trial_type
trial_array[]  )
{
   int num_trials, scan_return=0;

    for   (num_trials=1;scan_return!=EOF   &&
num_trials<=MAX_TRIALS;num_trials++){
       scan_return = fscanf(exp_file_ptr, "%d LENGTH: %lu ISI:
%lu TASK: %d TYPE: %d  DIGITS: %[0123456789]\n",
               &trial_array[num_trials].trial_number,
               &trial_array[num_trials].trial_length,
               &trial_array[num_trials].isi,
               &trial_array[num_trials].task_type,
               &trial_array[num_trials].digits_type,
               &trial_array[num_trials].numbers);
     }
     return(num_trials-2);
}

void record_experiment(                                    F   I   L   E
*results_file_ptr,                                    expt_trial_type
trial_array[],
expt_response_type response_array[],                       i    n    t
num_trials                          )
{
     int i;
     for( i=1; i<=num_trials;i++){
          fprintf(results_file_ptr,  "%d  SB:  %lu  WAS:  %lu
ESTIMATE: %lu\n",
                         trial_array[i].trial_number,
                         trial_array[i].trial_length,
                         response_array[i].actual_time,
                         response_array[i].time_in_millisecs);
     }
}

/* Using C/C++ p. 828 */
/*Settings for 1000/sec */

#include <stdlib.h>
#include <stdio.h>
#include <dos.h>
#include <conio.h>

#define ICR 0x20
```

```
#define EOI 0x20

unsigned long ticks;
unsigned long begin_time;
unsigned long end_time;
unsigned long far *clock = (unsigned long far *)0x0040006CUL;
unsigned long far *oflow = (unsigned long far *)0x00400070UL;
unsigned long save_clock;
unsigned long save_oflow;

void (interrupt far *oldint8) ();

void interrupt far newint8 ()
{
     static unsigned int count = 55;

     ++ticks;
     if (--count == 0 ) {
          (*oldint8)();
          count = 55;
     }
     else outp( ICR, EOI );
}

static void hires_clock (void)
{
     _disable();
     save_clock = *clock;
     save_oflow = *oflow;
     ticks = 0;
     oldint8 = _dos_getvect (8);
     _dos_setvect(8, newint8);
     outp( 0x43, 0x36);
     outp( 0x40, 0xa9);
     outp( 0x40, 0x04);
     _enable();
}

static void lores_clock( void )
{
     _disable();
     outp( 0x43, 0x36 );
     outp( 0x40, 0 );
     outp( 0x40, 0 );
     _dos_setvect( 8, oldint8 );
     *oflow = save_oflow;
     *clock = save_clock + (end_time -begin_time ) / 55UL;
     _enable();
}

void start_hires(void)
{
```

```
     hires_clock();
     begin_time = ticks;
}

void stop_hires(void)
{
     end_time = ticks;
     lores_clock();
}

double hires_duration( void )
{
     return ( ( (double) ( end_time - begin_time ) / 1000.0)
);
}


/* Pauses for a specified number of microseconds. Uses
newtime.h */
void sleep( unsigned long  wait )
{
    unsigned long goal;

    goal = wait + ticks;
    while( goal > ticks)
        ;
}

void beep( int frequency, unsigned long duration )
{
    int control;

    /* If frequency is 0, Beep doesn't try to make a sound. It
     * just sleeps for the duration.
     */
    if( frequency )
    {
        /* 75 is about the shortest reliable duration of a
sound. */
        if( duration < 75 )
            duration = 75;

        /* Prepare timer by sending 10111100 to port 43. */
        _outp( 0x43, 0xb6 );

        /* Divide input frequency by timer ticks per second
and
         * write (byte by byte) to timer.
         */
        frequency = (unsigned)(1193180L / frequency);
        _outp( 0x42, (char)frequency );
        _outp( 0x42, (char)(frequency >> 8) );
```

```
        /* Save speaker control byte. */
        control = inp( 0x61 );

        /* Turn on the speaker (with bits 0 and 1). */
        _outp( 0x61, control | 0x3 );
    }

    sleep( duration );

    /* Turn speaker back on if necessary. */
    if( frequency )
        _outp( 0x61, control );}
```