MASTER

Introductory research on the effects of operational sounds on the quality perception of consumer products

Louwers, M.T.

Award date:
1993

Introductory research on the effects of operational sounds on the quality perception of consumer products

M.T. louwers
Thesis
Eindhoven University of Technology

Abstract

This report is a result of an introductory research on the effects of operational sounds on the quality perception of consumer products.
In addition to traditional efforts to reduce the noise level of operational sounds in consumer products, which can be measured objectively, the goal more and more becomes to find ideal patterns of sound, which can only be measured subjectively. Therefore a listening test is defined and performed, in which operational sounds which can be associated with the loading action of a DCC-player, were judged by a number of subjects. The listening tests reveal that the model defined in this manner, provides results which account for opinion of the subjects. Also can be concluded that the average consumer tends to emphasize the role of the patterns of sound rather than the absolute sound level while judging the quality of a product by it's operational sounds.

Preface

This report is part of the final project I fulfilled in order to obtain a Masters degree in Physics at the Eindhoven University of Technology. The project was performed at the Philips Research Laboratories in Eindhoven and initiated by Ir. M.G.M. de Wit of the Noise abatement group and Prof. Dr. A.J.M. Houtsma of the Institute for Perception Research (IPO).

Michael Louwers, June 1993

' Man is the measure of all things,... '

(Protagoras of Abdera, 480-410 B.C.)

Contents

Chapter 1

Introduction

Quality control concerning consumer products is an important research issue within the Philips-concern. Primarily, however, this research is concerned with quality control of the primary functions of the consumer products.
In order to remain distinguishable from competitors, the question of how well the products meet the secondary requirements will become more and more important.

These secondary requirements are primarily related to the user-interface between product and user.
Requirements related to the quality of operational sounds are an example of these secondary requirements for consumer products.
In addition to traditional efforts for reducing operational sounds, some producers of consumer products have begun to emphasize the quality of operational sounds. More and more the goal becomes to find ideal patterns of sounds, which does not necessarily mean reduction of the sound level.

This report is a result of an introductory research on the effect of operational sounds on the quality perception of consumer products.
To measure this effect a listening test is performed, in which some operational sounds which can be associated with the sound caused by the loading action of a Digital Compact Cassette-player(DCC), were judged by a number of subjects.
This listening test is an imitation of the first contact between a potential user and a DCC-player, according to the 'Paired comparisons'-principle.
The objective of this introductory research is to answer the question if further research on this issue is useful.

In chapter 2 a theoretical background of the Paired comparisons experiment and Thurstone's judgment scaling method will be presented.
In chapter 3 an outline will be given of the technical realization of the listening test. Attention will be paid to the movement of the loader tray, the reproduction of the operational sounds and the synchronization of both visual and auditive stimuli.
Chapter 4 will outline the design of the listening test and give a description of the stimuli used in the tests.
In chapter 5 the results of the listening tests will be discussed.
Chapter 6 outlines the main conclusions of this research and gives some recommendations.

Chapter 2

Theoretical background of Paired comparisons and Scaling models

2.1 Introduction

Louis Leon Thurstone (1887-1955) pointed out that 'psychophysical' scaling methods could be used for accurate measurement of psychological attributes of stimuli which have no obvious measurable physical correlate. Thurstone developed a law of comparative judgment and showed that it was possible to obtain internally consistent measurements for various psychological attributes.

2.2 Scaling model according to Thurstone

The basic model underlying the Thurstone scaling methods is essentially as follows. Take as given a series of stimuli to which the subject can respond differentially with respect to some given attribute. Our task is to locate these stimuli on a *psychological continuum* in such way that one can account for the responses given by the subject. The psychological continuum can be considered to be a continuum of subjective or psychological magnitudes. Each psychological magnitude is mediated by a *discriminal process*. Each discriminal process thus has a value on the psychological continuum. Each stimulus when presented to a subject gives rise to a discriminal process. Because of momentary fluctuations in the organism, a given stimulus does not always excite the same discriminal process, but may excite one with a higher or lower value on the psychological continuum. If one presents a stimulus to the subject a large number of times, one can think of a frequency distribution on the psychological continuum of discriminal processes associated with that stimulus.

The postulate is made that the frequencies with which discriminal processes are associated with any given stimulus form a *normal distribution* on the psychological continuum. The scale value of the stimulus on the psychological continuum is taken as the value of its modal discriminal process. Since in normal distribution the mode, median and mean value coincide, the scale value of the stimulus can also be considered as the value of the mean discriminal process associated with it. The standard deviation of the distribution associated with a given stimulus is called the *discriminal dispersion* of that stimulus. The discriminal dispersions, as well as the scale values, may be different for different stimuli.

The observer cannot report directly the value of the discriminal process on the psychological continuum. Hence, one cannot obtain directly from the observer the frequency distribution associated with a stimulus. One can, however, deduce equations relating judgments of *relations among stimuli* to the scale values and dispersions of the stimuli on the psychological continuum. One of these sets of equations is known as *the law of comparative judgment.*

This law is concerned with paired comparisons judgments, that is, with judgments of the form 'stimulus A is X'er (e.g 'better') than stimulus B'.

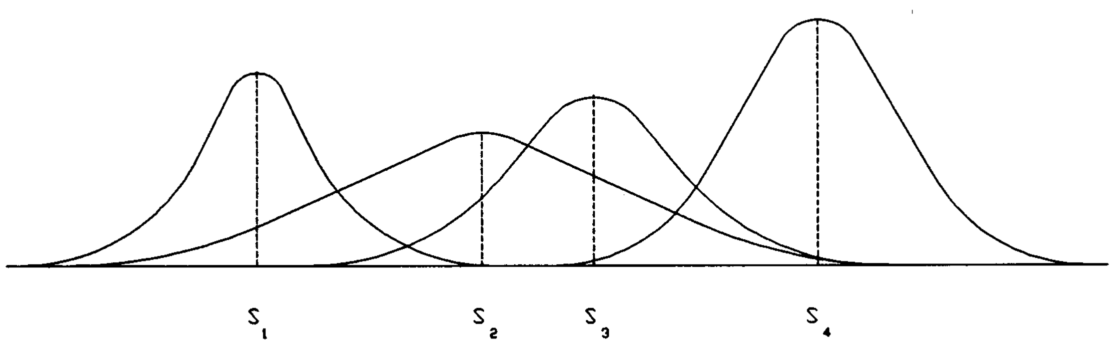


Fig. 2.1 Distributions on the psychological continuum of discriminal processes associated with four stimuli.

2.3 The law of comparative judgment

The law of comparative judgment is a set of equations relating the proportion of times any given stimulus k is judged greater on a given attribute than any other stimulus j to the scale values and discriminal dispersions of the two stimuli on the psychological continuum. The set of equations is derived from the postulate presented in section 2.2. Consider the theoretical distributions of discriminal processes for any two stimuli j and k as shown in Figure 2.2.

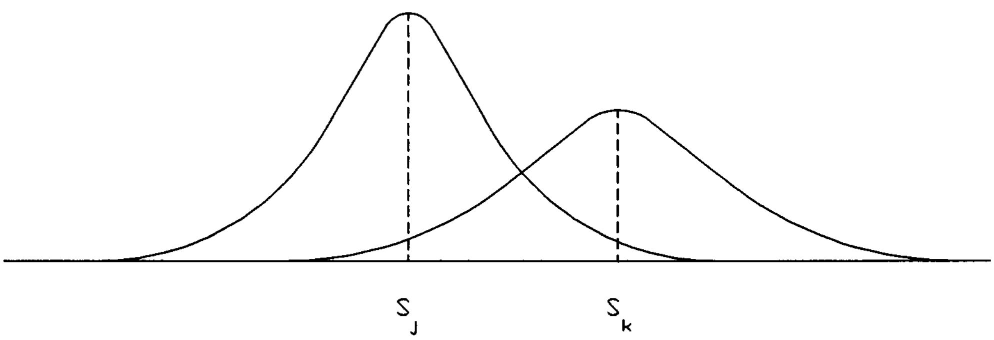


Fig. 2.2 The distribution of discriminal processes associated with stimuli j and k on the psychological continuum

Let s_j and s_k correspond to the scale values of the two stimuli and σ_j and σ_k to their discriminal dispersions. If the two stimuli were presented together to the subject, each would excite a discriminal process: $d(j)$ and $d(k)$. The difference in discriminal processes $d(k)$ - $d(j)$ for any single presentation of the pair of stimuli is called a *discriminal difference*. If the stimuli were presented together a large number of times, the discriminal differences themselves would form a normal distribution on the psychological continuum. The mean of this distribution is equal to the difference in scale values of the two stimuli, since the difference between means is equal to the mean of differences. From the formula of the standard deviation of differences, we know that

$$\sigma_{d(k)-d(j)} = \sqrt{\sigma_j^2 + \sigma_k^2 - 2r_{jk}\sigma_j\sigma_k} \tag{2.1}$$

where r_{jk} is the correlation between momentary values of discriminal processes associated with stimuli j and k. Each time the two stimuli are presented to the

observer, he is required to judge which is higher on the psychological continuum. It is assumed that the judgment 'stimulus k is greater than stimulus j' occurs whenever the discriminal difference $d(k)$ - $d(j)$ is positive. When the two distributions overlap, as in Figure 2.2, it is possible for the discriminal difference to be negative for any particular trial even though the scale value s_k is greater than s_j. From a large number of judgments, the proportion of times stimulus k is judged greater than stimulus j can be determined.
The distribution of discriminal differences on the psychological continuum is illustrated in Figure 2.3

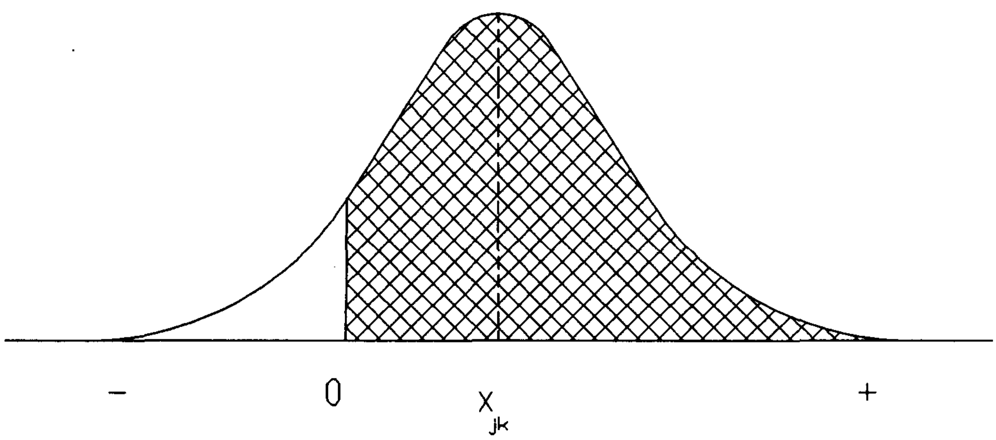


Fig. 2.3 The distribution of discriminal differences on the psychological continuum

The shaded portion to the right of the zero point corresponds to the proportion of times $d(k)$ - $d(j)$ is positive, and hence, the proportion of times stimulus k is judged greater than stimulus j. The mean of the distribution is equal to the difference in scale values of the two stimuli s_k - s_j. From the theoretical proportion of times stimulus k is judged greater than stimulus j one can determine the difference (s_k - s_j) from a table of areas under the unit normal curve. This difference is called x_{jk} and is measured in $\sigma_{d(k)-d(j)}$ units. We can thus write the equation,

$$s_k - s_j = x_{jk}\,\sigma_{d(k)-d(j)} \qquad (2.2)$$

Combining equation 2.1 and equation 2.2 gives,

$$s_k - s_j = x_{jk}\sqrt{\sigma_j^2 + \sigma_k^2 - 2r_{jk}\sigma_j\sigma_k} \tag{2.3}$$

where,

- *s_j and s_k denote the scale values of stimuli j and k.*
- *σ_j and σ_k denote the discriminal dispersions of stimuli j and k.*
- *r_{jk} is the correlation between the pairs of discriminal processes d(j) and d(k).*
- *x_{jk} is the normal deviate corresponding to the theoretical proportion of times stimulus k is judged greater than stimulus j.*

Equation 2.3 is the complete form of the law of comparative judgment.

2.4 Simplifying hypotheses

The law of comparative judgment is not solvable in it's complete form, since, regardless of the number of stimuli, there are always more unknowns than observation equations.

In order to arrive at a workable set of equations, i.e., one for which a large enough set of stimuli will give more observational equations than unknowns, it is necessary to specify additional restrictions. If we assume zero correlations and equal discriminal dispersions ($\sigma_j = \sigma_k = \sigma$ and $r_{jk} = 0$), then equation 2.3 would reduce to

$$s_k - s_j = x_{jk}\sigma\sqrt{2} \tag{2.4}$$

which is the equation of Thurstone's case V [11].

2.5 The method of Paired comparisons

All forms of the law of comparative judgment assume that each stimulus has been compared with each other stimulus a large number of times.

The law requires that data of the form 'the proportion of times any stimulus *k* is judged greater than any other stimulus *j*' are available. The direct method for obtaining empirical estimates of these proportions is known as the *method of paired comparisons.*

The method of paired comparisons is essentially a generalization of the two-category case of the method of constant stimuli. In this method, each stimulus is compared with a single standard. The method of paired comparisons is used primarily in cases when the stimuli to be compared can be judged only subjectively; that is to say, when it is impossible or impracticable to make relevant measurements in order to decide which of the two stimuli is preferable. In paired comparisons, each stimulus serves in turn as the standard. With n stimuli, there are thus $n(n-1)/2$ pairs. Each pair is presented to the subject, whose task is to indicate which member of the pair appears greater with respect to the attribute to be scaled. The statement that no ties are allowed is consistent with the derivation of the law, wherein the probability of a zero discriminal difference is small. In order to obtain data from which the proportions may be estimated, it is necessary that a large number of comparisons are made of each pair of stimuli. The necessary replication might be obtained by

- Having a single subject judge each pair a large number of times (class I models)
- Many subjects each judge each pair once (class II models)
- Several subjects each judge each pair several times (class III models)

The choice of class will depend upon

- The purpose of the experiment
- The extent of individual differences
- The nature of the stimuli

If individual differences are known or can be assumed to be negligible, any of the three alternatives may be used, whichever is the most convenient. If one is interested in the 'average' scale for a population, then the second alternative should be used, where the subjects are an appropriate sample from that population.

No explicit provision is made for *time or space errors* in the law of comparative judgment. Nor is there provision for changes in performance due to fatigue or practice effects, or for judgments based in part on factors other than the relative magnitudes of the discriminal process. Consequently it is necessary to control experimentally the conditions that might introduce these biasing effects. Most of these factors can be controlled in the assignment of the relative positions (spatial or temporal) of the members of each stimulus pair and the order of presentation of the pairs themselves.

Perhaps the best procedure is to counterbalance each pair of stimuli: e.g. with stimulus pair *(j,k)*, present *j* first half of the time, *k* first half of the time.
Ross [12] gives a general method for calculating "optimal" orders for stimulus pairs.

Table 2.1 Matrix of optimal order of presentation in case of odd n according to Ross.

0,1	n-1,2	n-2,3	n-3,4	...	½(n+3),½(n-1)
0,½(n+1)	1,2	n-1,3	n-2,4	...	½(n+3),½(n+1)
0,2	1,3	n-1,4	n-2,5	...	½(n+5),½(n-3)
0,½(n+3)	2,3	1,4	n-1,5	...	½(n+5),½(n+3)
...	...	...	...	...	...
0,½(n-1)	½(n-3),½(n+1)	½(n-5),½(n+3)	½(n-7),½(n+5)	...	1,n-2
0,n-1	½(n-1),½(n+1)	½(n-3),½(n+3)	½(n-5),½(n+5)	...	1,n-1

The orders are optimal in the sense that,

- Each stimulus appears first in half of the pairs of which it is a member.
- Pairs having one stimulus in common are maximally separated in the order of presentation.
- There is no detectable pattern in the order of presentation.

If in a stimulus pair *(j,k)* (which means that stimulus *j* is presented first (left) and stimulus *k* is presented second (right)), *j* is preferred most of the time and in the counterbalanced stimulus pair *(k,j)*, *k* is preferred most, one can conclude that the experiment is afflicted with either time errors or space errors.

2.6 The basic matrices

After each of the *n(n-1)/2* pairs of stimuli have been presented a large number of times, we have as raw data the number of times each stimulus was judged greater than each other stimulus. These observed frequencies may be arranged in the *n x n* square matrix **F**, the Raw Frequency Matrix (Table 2.2). The general element f_{jk}' which appears at the intersection of the *j*th row and *k*th column, denotes the observed number of times stimulus *k* was preferred to stimulus *j*. The diagonal cells will be left vacant. Since symmetric cells sum to the total of judgments of the pair made, the matrix contains $n(n-1)/2$ independent cells.

		Stimulus k					
		1	2	....	k	....	n
	1	-	f'_{12}		f'_{1k}		f'_{1n}
	2	f'_{21}	-	....	f'_{2k}	....	f'_{2n}
Stimulus j	.	.	.		.		.
	.	.	.		.		.
	j	f'_{j1}	f'_{j2}	....	f'_{jk}	....	f'_{jn}
	.	.	.		.		.
	n	f'_{n1}	f'_{n2}	....	f'_{nk}	....	-

Table 2.2 Matrix F of Raw Frequencies

In the matrix **P**, the Preference Proportion Matrix, which is constructed from the matrix **F**, the element p_{jk}' is the observed proportion of times stimulus k was preferred to stimulus j. Here symmetric cells sum to unity.

Again, matrix **X**, the Basic Transformation Matrix, is constructed from matrix **P**. The element x_{jk}' is the *unit normal deviate* (equation 2.5) corresponding to the element p_{jk}'. and may be obtained by referring to a table of areas under the unit normal curve [1],[5],[7].

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x_p} e^{-(z^2/2)} dz = P \qquad (2.5)$$

This X matrix is skew-symmetric: that is, the symmetric elements sum to zero, since $x_{jk}' = -x_{kj}'$. Matrix **X** contains the sample estimates x_{jk}' of the theoretical values found in the equation of the law of comparative judgment. The element x_{jk}' is an estimate of the difference s_k-s_j between the scale values of the two stimuli measured in units of the standard deviation of the distribution of discriminal differences $\sigma_{d(k)-d(j)}$. Each independent element of the matrix **X** is an estimate of a value for one equation of the law. However, since the elements are observed quantities, each will be somewhat in error. Analytical procedures have been designed which tend to allow the errors to cancel one another and thus give reasonably good estimates of the unknowns. One of these procedures is known as *The least squares solution*.

2.7 The least squares solution

Mosteller [13] has shown that the usual procedure for obtaining estimates of scale values from a matrix **X** which contains no vacant cells is a least squares solution. The set of equations is given by,

$$s_k - s_j = c x_{jk} \qquad (j,k = 1,2,\dots,n) \tag{2.6}$$

where,

- *s_j and s_k denote the scale values of stimuli j and k, respectively.*
- *x_{jk} is the unit normal deviate corresponding to the theoretical proportion of times stimulus k is preferred to stimulus j.*
- *c denotes the constant standard deviation of the distribution of discriminal differences.*

With fallible data (observed proportions in stead of theoretical proportions), the x_{jk}' become estimates of the true x_{jk}, and equation (2.6) no longer holds for all pairs of stimuli. The observed values x_{jk}' will be used to solve for a set of estimates s_j', s_k' the true scale values s_j , s_k.

Let the unit of measurement be such that the constant c in equation (2.6) is equal to unity.
then (2.6) simplifies to

$$s_k - s_j = x_{jk} \qquad (j,k = 1,2,\dots,n) \tag{2.7}$$

Let x_{jk}'' denote the difference between pairs of estimates of scale values, then

$$x_{jk}'' \equiv s_k' - s_j' \qquad (j,k = 1,2,\dots,n) \tag{2.8}$$

With errorless data, the derived x_{jk}'' will equal the corresponding observed x_{jk}'. With fallible data, they will be somewhat different. The task is to solve the set of estimates of the scale values of the stimuli for which the sum of squares of these discrepancies is a minimum, i.e., which minimizes the quantity Q, with

$$Q = \sum_{j=1}^{n} \sum_{k=1}^{n} (x_{jk}' - x_{jk}'')^2 \tag{2.9}$$

Since $x_{jk}'' = s_k' - s_j'$, one can write

$$Q = \sum_{j=1}^{n} \sum_{k=1}^{n} (x'_{jk} - s'_k + s'_j)^2 \qquad (2.10)$$

To minimize Q, one takes the partial derivative of Q with respect to each s'. Note that a particular value of s' (say s_2') appears only in the second row and second column of the matrix of squared errors.
Further, since $x_{jk}' = -x_{jk}'$ and $(s_k' - s_j') = -(s_j' - s_k')$, one needs only to be concerned with the columns. Differentiating the elements of each column with respect to s_k', we get

$$\frac{\partial Q}{\partial s'_k} = -2 \sum_{j=1}^{n} (x'_{jk} - s'_k + s'_j) \qquad (k = 1,2,...,n) \qquad (2.11)$$

Setting the partial derivative equal to zero and rearranging terms gives

$$\sum_{j=1}^{n} s'_k = \sum_{j=1}^{n} x'_{jk} + \sum_{j=1}^{n} s'_j \qquad (k = 1,2,...,n) \qquad (2.12)$$

Dividing by n gives

$$s'_k = \frac{1}{n} \sum_{j=1}^{n} x'_{jk} + \frac{1}{n} \sum_{j=1}^{n} s'_j \qquad (k = 1,2...,n) \qquad (2.13)$$

It will be convenient to set the origin at the mean of the estimated scale values, i.e., so that

$$\frac{1}{n} \sum_{j=1}^{n} s'_j = 0 \qquad (2.14)$$

In this case equation (2.13) reduces to

$$s'_k = \frac{1}{n} \sum_{j=1}^{n} x'_{jk} \qquad (k = 1,2...n) \qquad (2.15)$$

Thus, a least squares estimate of the scale values can be obtained simply by averaging the columns of matrix Q.

note:
The least squares solution requires that all elements in the matrix ***X*** *be present. However, whenever an observed proportion p_{jk}' is '1' or '0', the transformation to the corresponding x_{jk}' cannot be made.*

2.8 Solutions in case of an incomplete matrix X

Several procedures are available for obtaining estimates of the scale values when the Basic Transformation Matrix **X** contains unfilled cells, due to cells in the preference proportion matrix **P** valued 0 or 1.

The most direct procedure is to approximate a '0', in matrix **P**, with '0,01' (or '0.001' depending on the accuracy desired) and a '1' with '0,99' (or '0,999'). The matrix **P** obtained in this way can be treated further according to the procedure outlined in sections 2.6 and 2.7.

Another way to treat incomplete matrices is called the 'Traditional procedure for incomplete matrices' [1]. According to section 2.7 the differences between the theoretical equations for stimulus k and $k+a$ can be written as follows,

$$s_{k+a} - s_k = x_{j,k+a} - x_{jk} \qquad (j = 1,2,\ldots,n) \qquad (2.16)$$

In like manner, the corresponding differences between observed x-values ($x'_{j,k+a} - x'_{jk}$) is an estimate of the differences in estimates of scale values. For any two stimuli (k and $k+a$), there will be as many estimates as there are filled pairs of cells in the kth and $(k+a)$th columns of matrix **X**. The average of the estimates is taken as the estimate of the difference $d_{k,k+a}$.

$$d'_{k,k+a} = s'_{k+a} - s'_k = \frac{1}{n_k}\sum_{j=1}^{n_k} (x'_{j,k+a} - x'_{jk}) \qquad (2.17)$$

In practice, differences are obtained only for stimuli that are adjacent on the attribute being scaled. The usual procedure when constructing matrix **X** is to arrange it's columns in rank order with respect to the attribute (according to the sum of the columns of matrix **P**). Given the matrix **X** with columns arranged in rank order, the differences $d'_{k,k+1}$ are obtained using equation 2.17 with a=1. The zero point of the scale is located arbitrarily and the scale values are obtained by accumulating the successive differences.

The decision which of the procedures is best in estimating the theoretical scale values s_{jk} depends on the values of the obtained p'_{jk}. The criterion for deciding on the procedure to be used is called 'Goodness of fit' or applicability of the model to the data. An outline of this criterion is given in section 2.9.

2.9 Applicability of Thurstone's model

In the preceding sections, we have seen that, through the use of the *law of comparative judgment*, scales values of stimuli may be estimated from observed proportions p_{jk}'. Given these scale values, the procedure may be reversed; that is, derived proportions p_{jk}'' can obtained from the estimates of the scale values.
The difference between any two scale values (s_k' - s_j') is equal to the fitted normal deviate x_{jk}''. By performing the subtraction for all the values of j and k, a matrix of fitted normal deviates can be constructed. Since $s_k' - s_j' = -(s_j' - s_k')$, only those x_{jk}'' below the principal diagonal need to be computed. These fitted deviates may then be used to obtain a matrix of fitted proportions p_{jk}''.
In order to test the degree of applicability of the model to the data, one computes how well the fitted proportions p_{jk}'' correspond to the observed proportions p_{jk}'.
A common procedure is simply to obtain the average absolute deviation,

$$\overline{p_{jk}'' - p_{jk}'} = \frac{1}{n(n-1)} \sum_{j>k} |p_{jk}'' - p_{jk}'| \tag{2.18}$$

If the average discrepancy is 'small', it is concluded that the model fits adequately.

2.10 Statistical significance in a paired comparisons experiment

The sensitivity d' is defined as [9],

$$d' = \frac{(S_k - S_j)}{\sigma_{noise}} \tag{2.19}$$

where,

- *σ_{noise} denotes the discriminal dispersion of a single stimulus*

A sensitivity d'=1 is commonly used as the discrimination threshold. According to equation 2.4 in section 2.4, d'=1 in a paired comparison experiment corresponds with a theoretical normal deviate x_{jk} of $1/\sqrt{2}$. This x_{jk} corresponds with theoretical preference proportion p_{jk} of 0.76.

The experimental preference proportion p_{jk}' is binomially distributed [6], but will be

approximated by a normal distribution $N(\mu,\sigma_{exp})$.
The standard deviation of p_{jk}' depends on the value of p_{jk}' as well as on the number of times the stimulus pair is judged (equation 2.20).

$$\sigma_{exp} = \sqrt{\frac{p_{jk}'(1-p_{jk}')}{t}} \qquad (2.20)$$

where,

- *p_{jk}' denotes the measured preference proportion of stimulus pair (j,k)*
- *t denotes the number of times the stimulus pair is judged.*
- *σ_{exp} denotes the standard deviation of the experiment*

The reliability of such a experimental preference proportion is measured in units of σ_{exp}.
One can say, for instance if $p_{jk}' > 0.5$, with a probability of 95% that (when a pair j,k is judged an infinite number of times) the 'real' proportion p_{jk} for this subject is at least $(p_{jk}')_{min}$ with,

$$(p_{jk}')_{min} = p_{jk}' - 1.645\sigma_{exp} \qquad (2.21)$$

where,

- *1.645 denotes the value of the unit normal deviate corresponding with a significance level of 5% one-sided.*

Chapter 3

Experimental set up for Paired comparisons experiment

3.1 Introduction

In this chapter the most important technical features of the experimental set-up for the listening test will be outlined.

During the design of the experiment the most important criterion was that as little as possible appeal should be made to the imagination of the subject. The listening test should be an imitation of the first contact between a potential user and two DCC-players with the same appearance. As a result the experiment is designed in such way that,

- The experimental labour for the subject is reduced as much as possible.
- The auditory stimuli (the operational sounds to be tested) are combined with a visual stimulus (the movement of the tray during a loading action).
- Within each trial the operational sounds to be tested are reproduced by two different loader units.
- The beginning of each trial is triggered by the subject itself.
- Successive trials are independent of one another.

Figure 3.1 illustrates the total experimental set up.

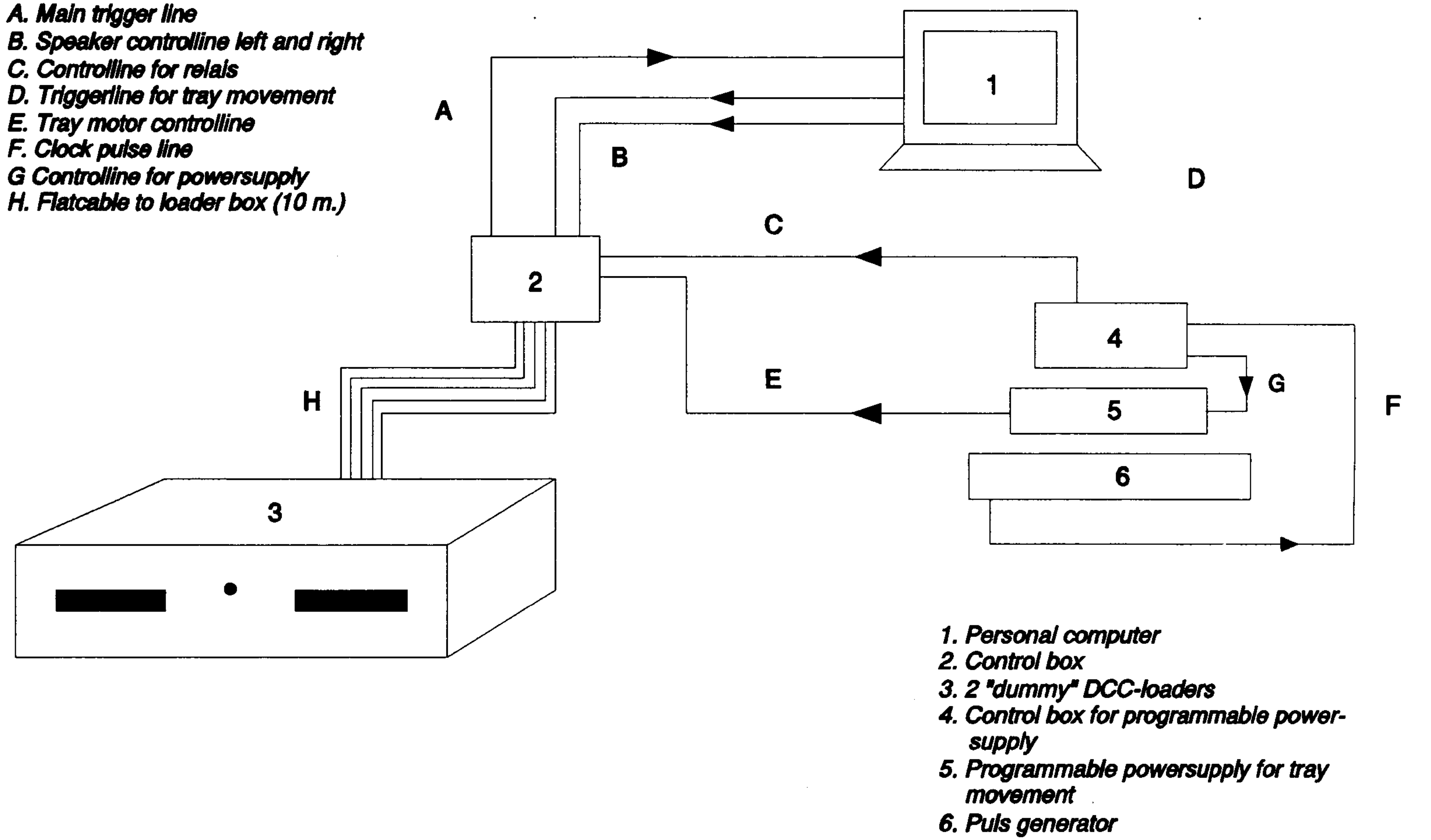


Fig. 3.1 Outline of experimental set up for listening-test

3.2 The auditory stimuli

By auditory stimuli we mean the different operating sounds used in the listening test are meant. These sounds are recorded with a *Sound Blaster Pro Audio Card* with a sample frequency of 44 kHz. (mono, 8 bits per sample) and edited in the Vedit2-Sound Editor. The different sound files are stored on the harddisk of a personal computer-system.

In section 3.1 the reason for combining the auditory stimulus with a visual stimulus was explained. Therefore a box with two 'dummy' DCC-loaders had to be designed (fig. 3.2). The DCC-loaders used for this purpose were original loaders taken from a DCC 600 player.
Because the listening test concerns operating sounds replacing the original sounds of a loading action, the dummy loaders had to be made as silent as possible during the loading action. Therefore few alterations were performed. The most important alteration was the removal of the cassette locking device. This alteration was possible because the locking action is not part of the visual stimulus in the original DCC-player.
Another important alteration was the placement of rubber stops at the ends of the tray movement.

The most important criterion for the electronically reproduced operating sounds was that they should sound as natural as possible.
Therefore the sound files, generated by the personal computer, were amplified and adjusted by an equalizer. In the test box each loader had its own compartment. For each loader the sound was reproduced by its own active speaker system.

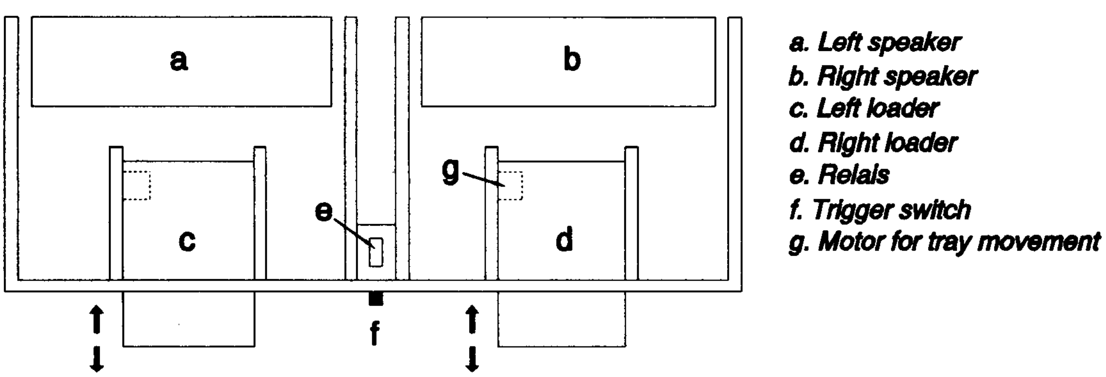


Fig. 3.2 Topview of testbox with dummy DCC-loaders and separate speaker systems

3.3 The visual stimulus

The visual stimulus belonging to a loading action consists of three parts,

- The outward movement of the tray
- A one second pause in the outmost position
- The inward movement of the tray

The cassette locking- and unlocking actions generate no visual stimulus because these actions occur inside the apparatus. In order to synchronize the auditive and visual stimuli a delay of 0.5 sec. had to be build into the control signal for the tray movement. The total control signal for the tray movement is illustrated in figure 3.3.

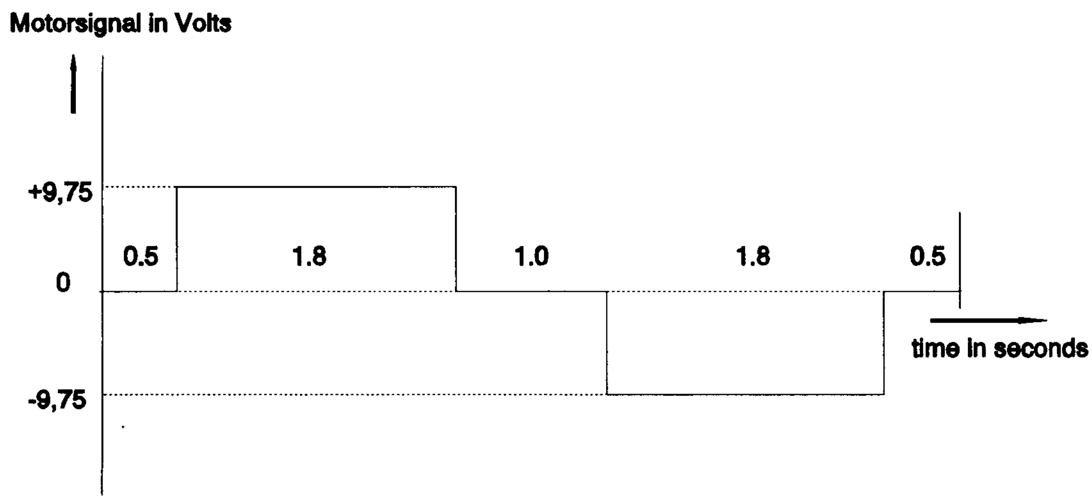


Fig. 3.3 Control signal for the tray movement in Volts as a function of time

The control signal is generated by a programmable power supply, which is able to execute a number of pre-defined steps. In this case 56 steps of 0.1 second are programmed, which is equal to one cycle of the tray movement. At the end of each cycle a relais is switched in order to change the tray to be controlled. Within each trial each tray is moved once. The clock pulses which control both power supply and relais are generated by a EPM5032DC-15 programmable IC (APPENDIX I).
Because the power supply could only generate positive voltages, a circuit had to be designed, which enabled the tray motor to be controlled between -9.75 Volts and +9.75 Volts (APPENDIX II).

3.4 Synchronization of the auditory and visual stimulus

One of the most important criteria mentioned in section 3.1 was that as little as possible appeal should be made to the imagination of the subject. Therefore it is very important for the auditory and visual stimulus to be synchronized as accurately as possible.
Another reason for the accurate synchronization is to prevent the *Ventriloquism effect* [10] from occurring. This phenomenon occurs whenever spatial information in light and sound is made conflicting. The information in one of the modalities is partly or wholly discarded in favour of that in the other. This mechanism is called *Stimulus dominance*.. Usually vision is found to be completely dominant over audition.
Therefore a computer program is written in which the trigger signals to both kinds of stimuli are controlled. This program is written in C++ and the listing is presented in APPENDIX III.

Chapter 4

Design of the Paired comparisons listening tests and description of the stimuli

4.1 Introduction

In designing a listening test the most important parameters are,

- The number of stimuli to be tested
- The nature of the stimuli to be tested
- The number of replications within one subject
- The number of subjects to participate

The ultimate choice of design will always be a trade off between the time available for the test and the type and amount of data required for a proper evaluation. In this chapter the design of the listening tests will be discussed.

4.2 Categorization of the total sound image

Because of the relatively complex nature of the operational sounds to be tested, the decision is made to divide the total sound image, associated with the loading action of a DCC-player, into 6 different parts according to their function within this loading action.
These 6 parts (figure 4.1) can be grouped into three categories because of pair wise similarities between functions. Part 1 can be paired with part 6, part 2 with part 5 and part 3 with part 4 (figure 4.2).

Stimuli representing one of these three categories are tested in separate Paired comparisons listening tests. Each category consists of five different stimuli.
Each representative of a particular category is combined with the original representative (the operating sound generated by the loading action of a DCC 600 player) of the other two categories in order meet with the requirement of genuineness as much as possible.
An example of a stimulus of category II, combined in that way, is given in figure 4.3.

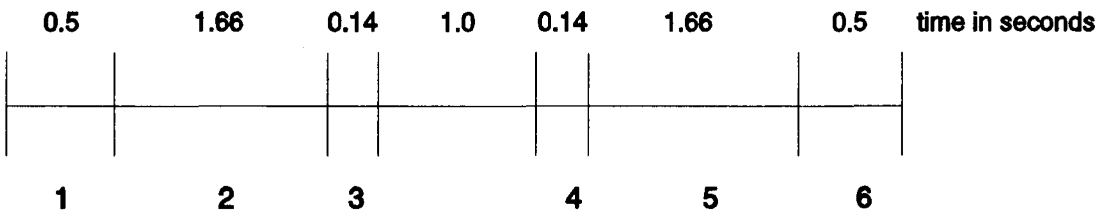


Fig. 4.1 Division of the total sound image into 6 parts according to functionality within the loading action.

1. Unlocking of cassette
2. Outward movement of tray
3. End of tray movement
4. Beginning of inward movement of tray
5. Inward movement of tray
6. Locking of cassette

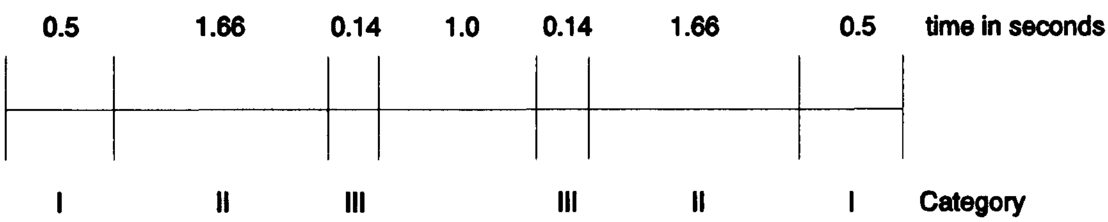


Fig. 4.2 Grouping into 3 categories because of pair wise similarities of functionality.

Category I. Locking/unlocking of cassette
Category II. Movement of tray
Category III. End/beginning of movement of tray

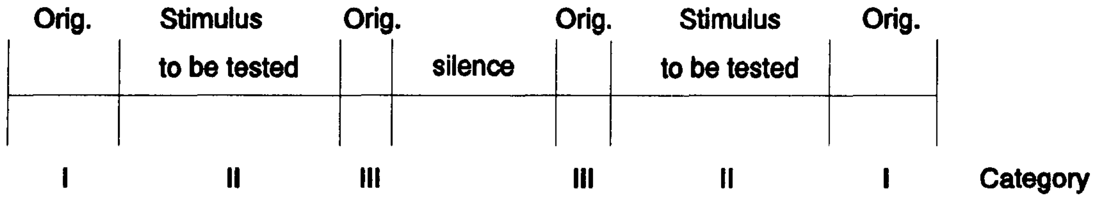


Fig. 4.3 Example of the construction of a stimulus of Category II.

4.3 Order of presentation of the stimuli

The number of pairs in each category is given by equation 4.1,

$$Number\ of\ pairs = \frac{n(n-1)}{2} \tag{4.1}$$

where
- *n denotes the number of stimuli in each category*

With 5 different stimuli within a category, designated as A,B,C,D and E, there will be 10 pairs to be tested in each category.

The stimuli can be ordered according to the theory discussed in section 2.5. Each stimulus appears first in half of the pairs of which it is a member, through time and space-errors will be avoided. Pairs having one stimulus in common are separated maximally and no detectable pattern in the order of presentation.
Table 4.1 illustrates an example of a series of stimulus pairs of category I, in which all stimulus pairs are presented 4 times, so that the series consists of 40 pairs.

Table 4.1 Optimal orders of 40 pairs of category I

A,B	C,E	D,A	B,C	E,D	A,C	D,B	E,A	C,D	B,E
A,D	C,B	D,E	C,A	B,D	A,E	D,C	E,B	B,A	E,C
D,B	E,A	C,D	B,E	A,B	C,E	D,A	B,C	E,D	A,C
E,B	B,A	E,C	A,D	C,B	D,E	C,A	B,D	A,E	D,C

note: The optimal orders for the categories II and III are obtained by replacing B by F or J, C by G or K, D by H or L and E by I or M

4.4 Description and recording of the stimuli

Of the 13 stimuli presented to the various subjects, 10 were recorded from existing mechanisms. Figure 4.4 illustrates the recording set up. The recording is made as directly as possible. For that reason the microphone is placed as closely to the object as possible and the recordings were made in a room with a relatively short reverberation time (± 1.5 sec.). In order to make the stimuli sound as naturally as possible, the same part of the sound field had to be recorded as is typically heard by the user. For that reason the angle between the recording microphone and the object was taken ± 45° (in front of the object). Once in the computer, the recordings were edited by the *Sound Blaster Pro Audio Card* as explained in section 4.2.
The remaining stimuli were reproduced by a *Sine random generator*, and presented directly to the *Sound Blaster Pro Audio Card*, where they were further edited.

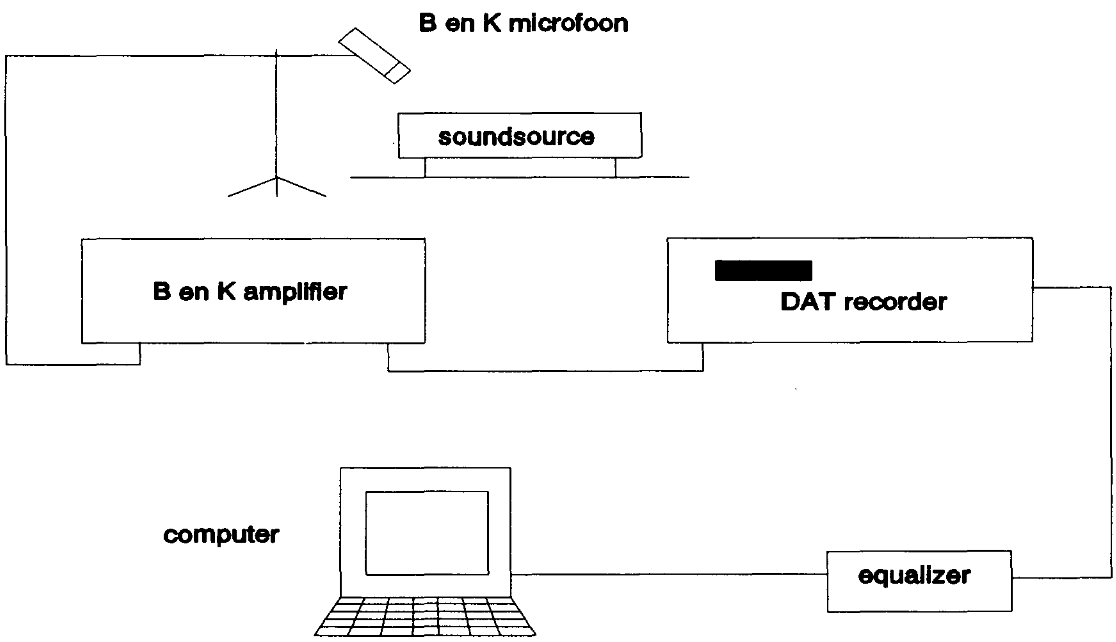


Fig. 4.4 Set up for recording the different stimuli

Category	Stimulus	Description of the stimulus	Stimuluslength
I,II,III	A	Original Philips DCC-600 player	2 x 2.3 sec.
I	B	Hydraulic sound	0.50 sec.
I	C	White noise imitating hydraulic sound	0.50 sec.
I	D	Original Philips CD 850 player	0.50 sec.
I	E	Original Philips DCC-600 player made rattling	0.50 sec.
II	F	Philips CD 850 with volume fade in and fade out	1.66 sec.
II	G	Original Philips DCC-600 player made rattling	1.66 sec.
II	H	Original Philips DCC-600 player made sliding	1.66 sec.
II	I	Original CD 850 with constant volume	1.66 sec.
III	J	Original Philips DCC-600 player made rattling	0.14 sec.
III	K	Informative sinus 'beep' 2.5 kHz faded in and out	0.14 sec.
III	L	Original Philips CD 850 player	0.14 sec.
III	M	Informative sinus 'beep' 2.5 kHz rectang. envelope	0.14 sec.

Table 4.2 Description of the stimuli according to categorization of section 4.2

In appendix V, plots of the time signals of stimulus A up to and until M are given.

4.5 Consistency test

According to the theory outlined in section 2.5 the kind of replication used in class I-models can be replaced by the kind of replication used in class II-models or in class III-models, if individual differences are known or can be assumed to be negligible. Another important condition for replacing the kind of replication is the measure of consistency for each individual.

In order to get an indication of the consistency of an individual while judging a certain stimulus pair, a listening test must be performed according to a class I-model. This listening test will provide a measure for the minimum number of replications needed in the following consumer test (t_{min}). In this consistency test each category will be tested separately.

According to section 2.10, where the d'=1 criterion was defined as the commonly used criterion for discrimination, an estimate will be made for the smallest number of replications needed to satisfy equation 2.21:

$$t_{\min} = \frac{p'_{jk}(1-p'_{jk})\ 2,706}{p'^{2}_{jk} - 2p'_{jk}\ p_{jk} + p^{2}_{jk}} \qquad (4.2)$$

where
- *p_{jk} denotes the theoretical preference proportion corresponding to d'=1.*

The problem is that the value of t_{min} depends on the value of p'_{jk}. The normal procedure is to evaluate the value of p'_{jk} after a number of replications and repeat this until the number of replications exceeds the value of t_{min} according to equation 4.2.

For each category a consistency test was performed with 3 subjects. Because within a category all pairs have to be judged the same number of times, the p'_{jk}'s were averaged over all pairs within one category as well as over the 3 subjects.

note: No account was taken for the 'direction' of the preference proportions;
$(1 - p'_{jk}) = p'_{kj}$

The resulting values for p'_{jk} and t_{min} are:

Category	p'_{jk}	t_{min}
I	0.96	3
II	0.92	8
III	0.89	16

In each category the pairs were judged 20 times per subject. Therefore the resulting theoretical proportions p_{jk} can be assumed to exceed the discrimination threshold d'=1 with a probability of at least 95% (see section 2.10). According to equation 4.3 the average consistency over the 30 stimulus pairs (10 pairs per category) is 0.925.

$$\overline{p'_{jk}} = \frac{\sum_{i=1}^{n} (p'_{jk})_i}{n} \tag{4.3}$$

where,

n denotes the number of categories

Therefore the number of replications within each subject in the following consumer test (Class III-model) can be taken relatively small. Decided is to take t = 4 in the consumer test.

Chapter 5

Results and discussion of Paired comparisons listening test

5.1 Introduction

The stimuli mentioned in section 4.4, were judged by 22 subjects. To obtain a representative sample of the population, 7 age groups wre made, each covering 5 years and all but one containing 3 subjects. The stimuli were tested in 3 separate listening tests each containing one category. According to equation 4.1, each category contains 10 different stimulus pairs, which are presented 4 times to each subject according to table 4.1. Therefore each separate listening test contains a series of 40 stimulus pairs. In spite of the fact that the subject was able to define it's own speed in judging the pairs, the average time interval for one judgment was approximately 16 seconds, resulting in a total time for the judgment of one category of 11 minutes.
To avoid judgement errors made by the subjects due to lack of concentration, a short pause was built in after each series, and the order of presentation of the series was taken random.
In the final results the preference data of 2 subjects were left out, because the responses proved to be too inconsistent.

5.2 Results

The following scale values are obtained by applying Thurstone's scaling model (chapter 2) in combination with Mosteller's least squares solution (section 2.7) to the averaged preference proportions (Table 5.1), where the entry (j,k) corresponds to the proportion of times stimulus *k* is preferred to stimulus *j*. In case of an incomplete matrix **X**, the first of the two procedures mentioned in section 2.8 is used (the value '1' is approximated by a '0.999', and the value '0' is approximated by '0.001'. This procedure results in a value for average absolute deviation that is 20% smaller than when using the traditional procedure for incomplete matrices.

Stimulus pair	Pref. Prop.	Stimulus pair	Pref. Prop.	Stimulus pair	Pref. Prop.
B,A	0.613	F,A	0.138	J,A	0.825
C,A	0.425	G,A	0.988	K,A	0.888
D,A	0.638	H,A	0.925	L,A	0.313
E,A	0.963	I,A	0.213	M,A	0.938
C,B	0.275	G,F	0.999	K,J	0.675
D,B	0.413	H,F	0.999	L,J	0.250
E,B	0.999	I,F	0.775	M,J	0.775
D,C	0.550	H,G	0.038	L,K	0.100
E,C	0.988	I,G	0.001	M,K	0.838
E,D	0.999	I,H	0.001	M,L	0.888

Table 5.1 Preference proportions averaged over 20 subjects

Stimulus pair	U.N.D.	Stimulus pair	U.N.D.	Stimulus pair	U.N.D.
B,A	0.287	F,A	-1.090	J,A	0.938
C,A	-0.190	G,A	2.260	K,A	1.220
D,A	0.355	H,A	1.440	L,A	-0.489
E,A	1.790	I,A	-0.797	M,A	1.540
C,B	-0.600	G,F	3.090	K,J	0.455
D,B	-0.220	H,F	3.090	L,J	-0.675
E,B	3.090	I,F	0.758	M,J	0.758
D,C	0.128	H,G	1.780	L,K	-1.285

E,C	2.260	I,G	-3.090	M,K	0.990
E,D	3.090	I,H	-3.090	M,L	1.220

Table 5.2 Unit normal deviates corresponding to preference proportions in Table 5.1

Stimulus	Scale value	Stimulus	Scale value	Stimulus	Scale value
A	0.630	A	0.507	A	0.903
B	0.562	F	2.271	J	-0.113
C	0.892	G	-2.882	K	-0.556
D	0.802	H	-1.652	L	1.035
E	-2.886	I	1.760	M	-1.269

Table 5.3 Scale values according to the least squares solution (section 2.8)

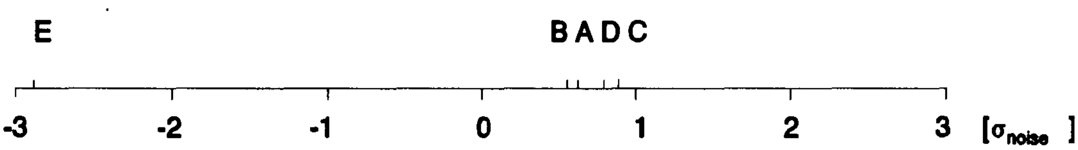


Fig.5.1 Scale values of stimuli belonging to category I

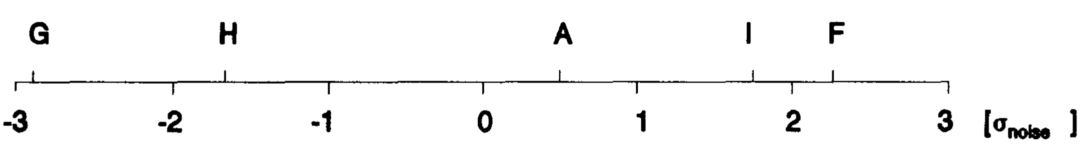


Fig 5.2 Scale values of stimuli belonging to category II

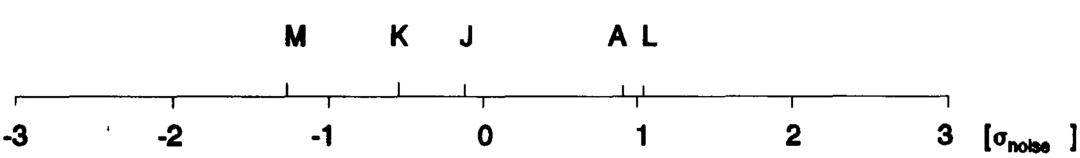


Fig. 5.3 Scale values of stimuli belonging to category III

5.3 Goodness of fit of the model to the data

According to the theory outlined in section 2.9 a measure of the applicability of the model to the data can be obtained by computing the fitted proportions p''_{jk} from the fitted deviates x''_{jk}. According to equation 2.8 x''_{jk} can be obtained by computing the difference between pairs of estimates of scale values $(s'_k - s'_j)$.
The tables 5.4 and 5.5 will illustrate respectively the values for x''_{jk} and p''_{jk}.

Stimulus pair	x''_{jk}	Stimulus pair	x''_{jk}	Stimulus pair	x''_{jk}
B,A	0.051	F,A	-1.243	J,A	0.722
C,A	-0.188	G,A	2.407	K,A	1.036
D,A	-0.117	H,A	1.531	L,A	-0.092
E,A	2.490	I,A	-0.881	M,A	1.562
C,B	-0.239	G,F	3.650	K,J	0.314
D,B	-0.168	H,F	2.774	L,J	-0.814
E,B	2.443	I,F	0.362	M,J	0.822
D,C	0.071	H,G	-0.876	L,K	-1.128
E,C	2.682	I,G	-3.228	M,K	0.508
E,D	2.611	I,H	-2.412	M,L	1.636

Table 5.4 Fitted deviates corresponding to the estimates of scale values s'_j illustrated in Table 5.3

Stimulus pair	p''_{jk}	Stimulus pair	p''_{jk}	Stimulus pair	p''_{jk}
B,A	0.520	F,A	0.107	J,A	0.765
C,A	0.425	G,A	0.992	K,A	0.849
D,A	0.455	H,A	0.934	L,A	0.464
E,A	0.994	I,A	0.189	M,A	0.941
C,B	0.406	G,F	0.999	K,J	0.623
D,B	0.434	H,F	0.997	L,J	0.208
E,B	0.992	I,F	0.641	M,J	0.794
D,C	0.528	H,G	0.191	L,K	0.129
E,C	0.996	I,G	0.001	M,K	0.694
E,D	0.995	I,H	0.008	M,L	0.949

Table 5.5 Fitted Preference proportions corresponding to the fitted deviates of Table 5.4.

According to equation 2.18 the average absolute deviation between p'_{jk} and p''_{jk} per category can be obtained (Table 5.6).

Category	Average absolute deviation
I	0.025
II	0.018
III	0.030

Table 5.6 Average absolute deviations per category according to equation 2.18

According to table 5.6 one can say that the *least squares solution* in combination with the first of the procedures mentioned in section 2.8 gives a correlation coefficient between the obtained proportions and the fitted proportions of 0.97 in the categories I and III and of 0.98 in category II.

5.4 Discussion

Category I

Figure 5.1 reveals that on the average stimulus *C* is preferred most, although the 4 most preferred stimuli are well within one σ_{noise}. This means that there is not much agreement between the various subjects about the ranking of the stimuli in category I, despite of the fact that the subjects individually were very consistent in their preference. The differences between A,B,C and D is perceptually not significant. Subjects who preferred stimulus *C* often also preferred stimulus *B*, because they both were hydraulic-like. So grouping of stimuli *B* and *C* and stimuli *A* and *D* often occurred. Stimulus *E* was disliked by all subjects.

Category II

Figure 5.2 reveals that stimulus *F* is preferred most, although the difference in scale value with stimulus *I* is not very large (less than one scale unit). They both differ from stimulus *A* more than one scale unit.
The difference between stimulus *A* and the stimuli *G* and *H* is so large that one can say with almost absolute certainty that the subjects are very sensitive to sliding and rattling noises during the tray movement. The overall difference in scale values is largest in this category, because of the great length of the part of the total sound image to be tested in this category.

Category III

Figure 5.3 illustrates that stimuli *A* and *L* are obviously preferred to the stimuli *M*,*K* and *J*, because the distance is more than one scale unit. Stimuli *M* and *K* (informative 'beeps') are disliked even more than stimulus *J* (rattling noise). Stimulus *K* ('soft beep') is preferred to stimulus *M* ('rectangular beep').

Combining the most and less preferred representatives

Figure 14 and 15 in APPENDIX V illustrate the time signals of the total sound image. They arise when combining respectively the best and worst representatives of each category.

Chapter 6

Conclusions and recommendations

As already mentioned, this research on the association of operational sounds with the quality perception of consumer products had a introductory character. A procedure for future testing is defined. The theoretical part of this procedure contains a scaling model according to Thurstone in combination with Mosteller's least squares solution. The listening test itself was designed according to Ross' theory concerning optimal orders of presentation of the stimuli.
The criterion for the number of replications within one subject is based on a one-sided significance level of 5% at the discrimination treshold (d'=1). A consistency test (section 4.5) was performed in order to find this minimum number of replications needed in the consumer listening test.
Because of the introductory character of this research, the operational sounds to be tested had to be chosen from a large collection of possible sounds. The total sound image is divided into parts according to functionality (section 4.2) in order to simplify the task for the subjects and to ease analysis for possible mechanical implementation. This categorization reduces the number of trials and the subjects were more able to distinguish between the various stimuli, because in one listening test only one part varied while the other two parts remained constant.
While choosing the stimuli an attempt was made to keep the original stimulus (sound belonging to loading action of DCC 600 player) in the middle of the ranking on the preference scale. Figure 5.1, 5.2 and 5.3 reveal that the strongest preference occurs in category II. This is probably the result of the fact that in this category the part to be tested (the movement of the loader tray) is relatively long, through the stimuli become more distinct.

As an overall result the tests reveal that changing sounds within steady state portions are seldom preferred. They will often be associated with a mechanical malfunction or defect. Clicking sounds can have a positive effect on the quality perception, provided that they can be associated with an obvious function within the total action, e.g. an indication of the end of a certain action (like slamming a door). Clicking sounds must also sound 'confident'; short, relatively much energy in the low frequency area and only one or two peaks in the time signal.
Although the informative 'beeps' tested in category III were disliked on the average, many of the participating subjects indicated that 'beeps' can be informative if used at the right moment.

The most important result of this introductory research is that in judging the quality of a product the average consumer tends to emphasize the role of the *patterns* of sound belonging to a certain operating action rather than the absolute sound level caused by the operating action.

References

[1] Torgerson, W.S. *Theory and methods of scaling*. John Wiley and Sons, 1958.

[2] David, H.A. *The Method of Paired Comparisons*. Charles Griffin and Company Ltd., 1988.

[3] Buxton, W., Gaver, W., Bly, S. *The use of Non-speech Audio at the Interface; A draft manuscript*. Cambridge University Press, 1991.

[4] Houtsma, A.J.M., Tholen, H.J.G.M. *A carillon of major third bells II: A perceptual evalution*. Music perception Vol. 4, 255-266, 1987.

[5] Owen, D.B. *Handbook of Statistical Tables*. Addison and Wesley publishing company, 1962.

[6] Hoel, P.G. *Introduction to Mathematical Statistics*. John Wiley and Sons, 1947

[7] Dunn-Rankin, P. *Scaling methods*. Lawrence Erlbaum Associates, Inc,1983.

[8] Kruskal, J.P. *Multidimensional scaling*. Bell laboratories, Murray Hill, New York, 1978.

[9] Green, D.M., Swets, J.A. *Signal detection theory and psychophysics*. New York, John Wiley and Sons, 1966.

[10] Howard, I.P., Templeton W.B. *Human spatial orientation*. University of Durham, England. John Wiley and Sons, 1966.

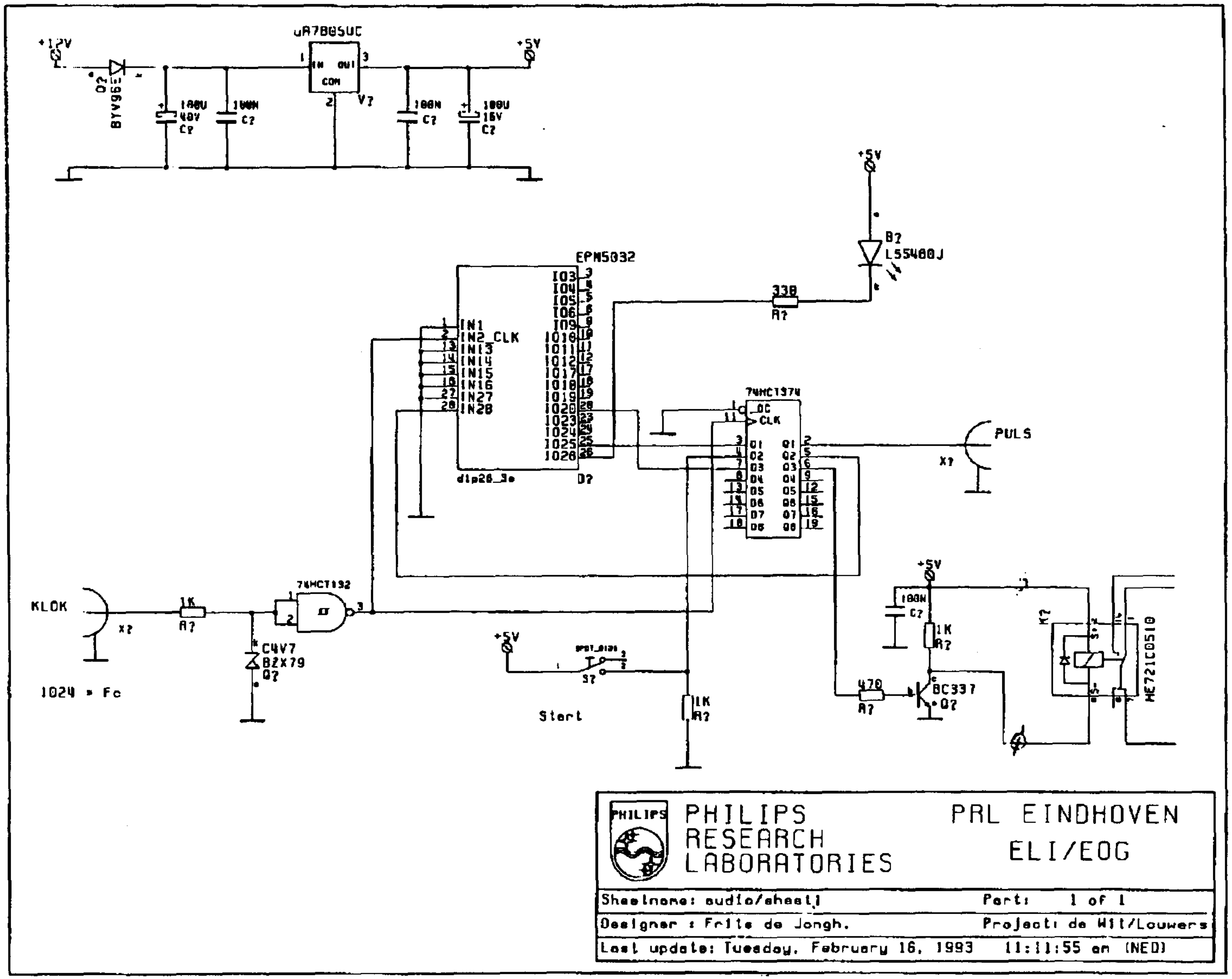
[11] Thurstone, L.L. A law of comparative judgment. *Psychol. Rev.*, 34, 273-286.

[12] Ross, R.T. Optimum orders for the presentation of pairs in the method of paired comparisons. *J. Educ. Psychol.*, 25, 375-382.

[13] Mosteller, F. Remarks on the method of paired comparisons. I. The least squares solutions assuming equal standard deviations and equal correlations. *Psychometrika*, 16, 1951 , 3-11.

Appendix I

Circuit design and listing of EPM5032

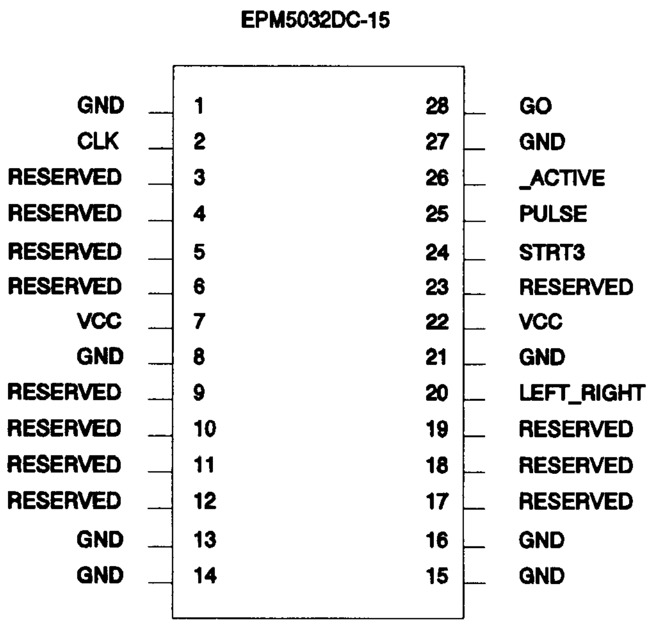


+12V
BYV96E
D?
100U
40V
C?
100N
C?
uA7805UC
IN
OUT
COM
V?
100N
C?
100U
16V
C?
+5V
+5V
B?
LS5460J
330
R?
EPM5032
IN1
IN2_CLK
IN13
IN14
IN15
IN16
IN27
IN28
IO3
IO4
IO5
IO6
IO9
IO10
IO11
IO12
IO17
IO18
IO19
IO20
IO23
IO24
IO25
IO26
dip28_3e
D?
74HCT374
OC
CLK
PULS
X?
KLOK
X?
1K
R?
74HCT132
C4V7
BZX79
D?
1024 * Fc
+5V
S?
Start
1K
R?
+5V
100N
C?
1K
R?
470
R?
BC337
Q?
K?
HE721C0510
PHILIPS
PHILIPS RESEARCH LABORATORIES
PRL EINDHOVEN
ELI/EOG
Sheetname: audio/sheet1
Part: 1 of 1
Designer : Frits de Jongh.
Project: de Wit/Louwers
Last update: Tuesday, February 16, 1993 11:11:55 am (NED)

audio i:\louwers\epld\sxty_cnt\audio.rpt

***** Logic for device 'audio' compiled without errors

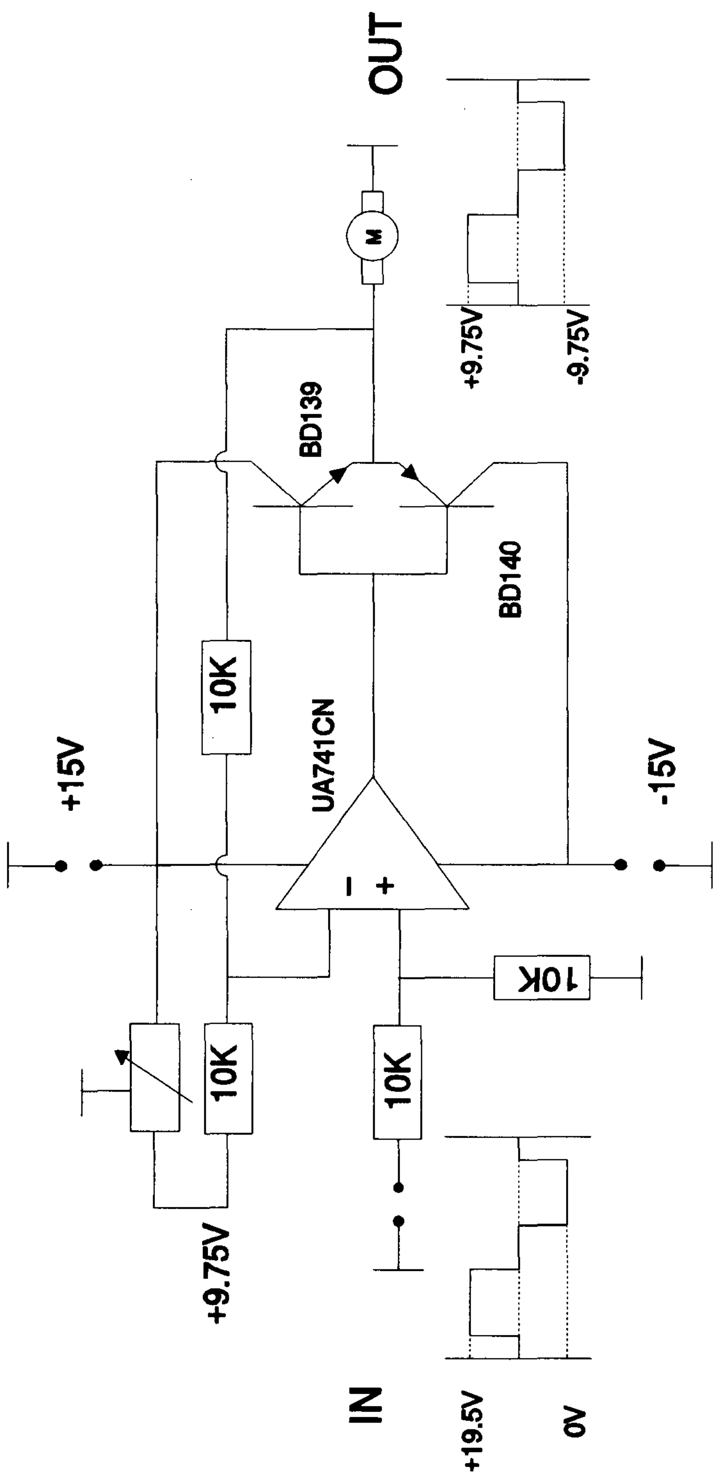
Device: EPM5032DC-15
Security: OFF



```
%***************************************
*                                     *
* sixty counter to make 60 pulses     *
* within 120 clockpulses              *
***************************************%

TITLE "counter for 60 pulses"; Author F. de Jongh, Philips Natuurkundig Laboratorium

DESIGN IS "SIXTY"
DEVICE IS "AUTO";

SUBDESIGN SIXTY

(
  STRT                      : INPUT;
  CLK                       : INPUT;
  PULSE                     : OUTPUT;
  ACTIVE                    : OUTPUT;
  STRT3                     : OUTPUT;
  LEFT_RIGHT                : OUTPUT;
)

VARIABLE
  ST1                     : DFF;
  ST2                     : DFF;
  ST3                     : DFF;
  Q[15..0]                : DFF;
  ACT                     : DFF;
  SWAP                    : DFF;
BEGIN
  % all register clocks set to clk %
   STI.CLK                = CLK;
   ST2.CLK                = CLK;
   ST3.CLK                = CLK;
   Q[].CLK                = CLK;
   ACT.CLK                = CLK;
   SWAP.CLK               = CLK;
   STI.D                  = STRT;
   ST2.D                  = STI.Q;
   ST3.D                  = STI.Q AND (NOT ST2.Q);

IF Q[] = = O THEN
  CASE ST3.Q IS
     WHEN VCC = > Q[] = 1;
            SWAP.D =SWAP.Q;
     WHEN GND = > Q[] = O;
            SWAP.D =SWAP.Q;
  END CASE;
ELSE
 IF Q[] = = 61440 THEN
   Q[] = O;
   SWAP.D = !SWAP.Q;
 ELSE
   Q[] = Q[] + 1;
   SWAP.D =SWAP.Q;
 END IF;
END IF;
% Drive output ports with data %
ACT          = Q[] = = O;
_ACTIVE       = ACT;
STRT3        = ST3.Q;
PULSE        = Q[9];
LEFT_RIGHT = SWAP;

END;
```

Appendix II

Circuit design for control of tray motor.



Appendix III

Listing of controlprogram "DCC"

```
//                  Author: M. de Wit
//                                          Building WAK-p-01
//                                   Philips Research Laboratories
//                   tel:(int)-31-40-742323

#include <dir.h>          //findfirst(),ffblk
#include <stdlib.h>       //getenv(),system(),malloc()
#include <io.h>           //_close(),_open(),filelength()
#include <fcntl.h>        //O_BINARY,O_RDONLY
#include <dos.h>          //freemem(),SREGS,REGS,_dos_open etc.
#include <string.h>       //strcat(),strcpy()
#include <fstream.h>      //ifstream ofstream
#include <iostream.h>     //cout,cin
#include "sbcpp.h"

#define MAXPATH 80

struct  baserec {
char         code[6];
char         name[32];
char         remarks[32];
};

void         sbsetup(void);

class  voicefile {
unsigned char     button;
char         vfname[80];
public:
                  voicefile(char * initvfname);
void         play(void);   //output voice from disk.
};

class  doublevoice {
voicefile *       DCC1;
voicefile *       DCC2;
unsigned char     button;
unsigned          channel;
unsigned          numofelem,numofpairs;
baserec  *        testbase;
ifstream          setfile,basefile;
ofstream          outfile;

public:
                  doublevoice(void);
void         FastPlay(void);
void         OpenFiles(void);
void         Test(void);
};
#include "sbdcc.h"

#define LEFT  0xff00
#define RIGHT 0x00ff
#define MASTER 0
#define VOICE  1

// prototypes
unsigned          DosReadDrv (int Handle, char far *Buffer, unsigned wLen, unsigned *wByteRead);
char far *        LoadDriver (char *szDrvName);
extern "C" char   jstatdcc(void);

extern char far * near  ctvdsk_drv;
extern char far * near  CTAuxDrv;

//************************************************
//*   General functions for reading the basefile. *
//************************************************

istream& operator >> (istream& s, baserec& m) {
   char ch;
   s.get(m.code,5,'\n');
   m.code[5]='\o';
   s.get(m.name,31,'\n');
   m.name[31]='\o';
   s.get(m.remarks,31,'\n');
   m.remarks[31]= '\o';
   s.get(ch);
```

```
   return s;
};

//*********************************************
//*     General functions for the SBPRO       *
//*********************************************

void sbsetup(void)
{
   if (!GetEnvSetting())
   {
    if (sbc_check_card() & 4)
    {
            if (sbc_test_int())
            {
             if (sbc_test_dma() >= 0)
             {
              if ((ctvdsk_drv = LoadDriver("CTVDSK.DRV")) != 0)
              {
               if ((CTAuxDrv = LoadDriver("AUXDRV.DRV")) != 0) //if (!ctvd_init())
                      return;
               else
                      cout << "Error while loading 2nd driver.\n" << flush;
              }
              else
               cout << "Error while loading driver.\n" << flush;
             }
             else
              cout << "Error on DMA channel.\n";
            }
            else
             cout << "Error on interrupt.\n";
    }
    else
            cout << "Sound Blaster Card not found or wrong I/O settings.\n";
   }
   else
    cout << "BLASTER environment not set, incomplete or invalid.\n";
   exit(0);
   return;
};

// DESCRIPTION:
//     Loads driver into memory with the driver name specified. The
//     driver is always loaded to the offset 0 of a segement.
// ENTRY:
//     szDrvName :- Driver name to be loaded.
// EXIT:
//     Pointer to the loaded driver if successful, else returns NULL
char far * LoadDriver (char *szDrvName) {
   void *          driverp;
   char far *       lpDrvPtr = 0 ;
   char far *       lpPtr ;
   char            szDrvFile[100];
   char *           pPtr ;
   int            Handle = 1, NotDone = 1;
   unsigned         wDrvSize;
   unsigned         wTemp, wDrvSeg ;
   struct ffblk      stFile ;


   /* set the default file mode to binary mode */
   _fmode = O_BINARY;

   /* locate driver through environment parameter */
   if ((pPtr = getenv("SOUND")) != 0)
   {
    _fstrcat(_fstrcpy(szDrvFile,pPtr),"\\DRV\\") ;
    _fstrcat(szDrvFile,szDrvName);

    /* NotDone set to 0, if found */
    NotDone = findfirst(szDrvFile,&stFile,0) ;
   }

   /* locate driver in current directory */
   if (NotDone)
   {
```

```
    _fstrcpy(szDrvFile,szDrvName) ;
              /* NotDone set to 0, if found */
    NotDone = findfirst(szDrvFile,&stFile,0) ;
   }

   if (NotDone)
    cout << "Driver file does not exist.\n" ;
   else //Driver exists.
   {
    if ((Handle=_open(szDrvFile,O_RDONLY)) == -1)
         cout << "Open %s error.\n",szDrvFile ;
    else
    {
         wDrvSize = (unsigned) filelength(Handle) ;

         driverp = malloc( (size_t) (wDrvSize + 16) ); //allocate +16bytes = +1 paragraph
         // to be able to load driver at offset 0 of a new segment
         if (driverp != 0)
         {
          wDrvSeg = FP_SEG(driverp)+1;
          lpDrvPtr = (char far *)((unsigned long)wDrvSeg << 16);
          lpPtr = lpDrvPtr;

          if ( DosReadDrv(Handle,lpPtr,wDrvSize,&wTemp) == 0 )
          {
           freemem(--wDrvSeg) ;
           lpDrvPtr = 0 ;
          }
         }
         else
          cout << "Memory allocation error.\n";

         _close(Handle) ;
    }
   }
   return(lpDrvPtr) ;
};

// DESCRIPTION:
//     Read driver to buffer using DOS interrupt 0x21 function 0x3F.
// ENTRY:
//     Handle :- File handle to read.
//     Buffer :- Buffer to write to.
//     wLen   :- Number of byte to read.
//     lpByteRead :- pointer to number of byte actually read.
// EXIT:
//     Byte read if successful, else returns 0.
unsigned DosReadDrv (int Handle, char far *Buffer, unsigned wLen, unsigned *wByteRead) {
   union REGS       regs;
   struct SREGS     segregs;

   regs.h.ah = 0x3f ;
   regs.x.bx = Handle;
   regs.x.dx = FP_OFF(Buffer);
   regs.x.cx = wLen;
   segregs.ds = FP_SEG(Buffer);

   intdosx(&regs, &regs, &segregs);

   if(regs.x.cflag)   /* error */
    *wByteRead = 0;
   else
    *wByteRead = regs.x.ax ;

   return(*wByteRead);
};

//**********************************************
//*    Member functions of class voicefile.        *
//**********************************************
voicefile::voicefile(char * initvfname) {
   _fstrcpy(vfname,initvfname);
};

#pragma loop_opt(off)
void voicefile::play(void) {
   int Handle;
```

```
   if (!ctvd_init(16)) {
    if (!_dos_open(vfname, O_RDONLY,&Handle))
    {
            ctvd_speaker(1) ;

            do {
              button = jstatdcc();}
            while (!( (button == 0xD0) || (button == 0xE0) ));
            outport(0x378,0x10);
            if (ctvd_output(Handle) == NO_ERROR) {
              while(ct_voice_status);

            } else
              cout << "ctvd_output error in voicefile::play.\n";
            outport(0x378,0x0);
            ctvd_speaker(0) ;
            _dos_close(Handle);
    } else
            cout << "Cannot open voicefile: " << vfname << "\n" << flush;

    ctvd_terminate();
   }
   return;
};
#pragma loop_opt(on)

//****************************************
//*  Member functions of class doublevoice *
//****************************************

doublevoice::doublevoice(void) {
   DCC1 = 0;
   DCC2 = 0;
   channel = LEFT;
};


void doublevoice::FastPlay(void) {

    ctadInit();
    switch (channel) {
    case LEFT:
            ctadSetVolume(MASTER,LEFT);
            ctadSetVolume(VOICE,LEFT);
            DCC1->play();
            break;
    case RIGHT:
            ctadSetVolume(MASTER,RIGHT);
            ctadSetVolume(VOICE,RIGHT);
            DCC2->play();
            break;
    default:
            break;
    } // switch(channel)
    ctadTerminate();
   return;
};

void doublevoice::OpenFiles(void){

   unsigned numofbase;
   char    elementline[MAXPATH], elinesp[MAXPATH],buffer[MAXPATH];
   char    SetFileNm[MAXPATH];
   char    OutFileNm[MAXPATH];
   char    ch, pair[2];
   char    ExpName[10],SubjName[10];
   baserec  testbuf;
   int     i,j,serialno,done;
   struct  ffblk ffblk;
   struct  date  d;

   do {
   cout << "Enter experiment name (min. 3 characters): ";
   cin >> ExpName;
   } while (_fstrlen(ExpName) < 3);
   do {
   cout << "Enter subject name (min. 4 characters): ";
```

```
cin >> SubjName;
} while (_fstrlen(SubjName) < 4);
_fstrcpy(OutFileNm,"");
_fstrncat(OutFileNm,ExpName,3);
_fstrncat(OutFileNm,SubjName,4);

serialno=0;
_fstrcat(OutFileNm,itoa(++serialno,&ch,10));
_fstrcat(OutFileNm,".dcc");
done = findfirst(OutFileNm,&ffblk,0);
while (!done) {
  OutFileNm[7]= *itoa(++serialno,&ch,10); //add check i>=10
  done = findfirst(OutFileNm,&ffblk,0); //add check i>=10
}

cout << "Directory listing of *.set\n";
done = findfirst("*.set",&ffblk,0);
while (!done)
{
  cout << ffblk.ff_name << "\n";
  done = findnext(&ffblk);
}

cout << "Which setfile: ";
cin >> SetFileNm;
setfile.open(SetFileNm,ios::nocreate); //infile must be there
if (!setfile)
{
  cout << "Cannot open" << SetFileNm << "\n";
  exit(-1);
}
_fstrcat(getcwd(buffer,MAXPATH),"\\");
_fstrcat(buffer,_fstrupr(SetFileNm));
_fstrcpy(SetFileNm,buffer);

basefile.open("dcc.bse",ios::nocreate); //infile must be there
if (!basefile)
{
  cout << "Cannot open dcc.bse.\n" << flush;
  exit(-1);
}
// fill testbase[] and read header of setfile
//at the end of this loop all participating 'character' - 'voc filename'
//combinations are put in testbase[]
basefile >> numofbase;
basefile.get(ch);
setfile >> numofelem;
setfile.get(ch);
if ((numofelem != 0) && (numofbase != 0) )
{
  testbase = new baserec [numofelem];
  for (i=0; i < numofelem; i++) {
        setfile >> elementline[i];
        elinesp[4*i] = elementline[i];
        elinesp[4*i+1] = ' ';
        elinesp[4*i+2] = ' ';
        elinesp[4*i+3] = ' ';
  }
  elementline[numofelem] = '\x0';
  elinesp[4*numofelem] = '\x0';
  setfile.get(ch);
  j=0;
  for (i = 0;i < numofbase;i++)
  {
        basefile >> testbuf;
        if ( _fstrpbrk(elementline,testbuf.code) )
          testbase[j++]=testbuf;
  }
}
else
{
  cout << "Empty Basefile or empty Setfile!\n";
  exit(-1);
}
basefile.close();
setfile >> numofpairs;
setfile.get(ch);
```

```
    // Start of test
    getdate(&d);
    outfile.open(OutFileNm); //created or cleared, overwritten.
    outfile << " Paired Comparisons.\n\n";
    outfile << " Gebruikte database      : " << "C:\\SBPRO\\DCC.BSE\n";
    outfile << " Gebruikte setfile       : " << SetFileNm <<"\n";
    outfile << " Element benaming        : " << elinesp <<"\n";
    outfile << " Experiment name         : " << ExpName << "\n";
    outfile << " Subject name            : " << SubjName << "\n";
    outfile << " Serial number           : " << serialno << "\n";
    outfile << " Date                    : " << d.da_year<<" "<<(unsigned)d.da_day<<" "<<(unsigned)d.da_mon<<"\n";
    outfile << "\n";
    outfile << "------------------------------------------------------------------------\n";
    return;
    };

void doublevoice::Test( void ) {

    char    ch, pair[2];
    int     i,ipair;

    for ( ipair = 0; ipair < numofpairs; ipair++ )
    {
     setfile >> pair;  // read numbers e.g. 52
     setfile >> pair;  // read matching characters e.g. EB
     setfile.get(ch);

     i=0;
     while ((!_fstrchr(testbase[i].code,(int) pair[0])) && (i < numofelem))
     { i++;} //search index of testbase that matches pair[0]
     if (i >= numofelem)
     {
           cout << "Element 1 of pair " << i+1 <<" does not exist.\n" << flush;
           exit(-1);
     }
     else
     {
           if (DCC1)
             delete DCC1;
           DCC1 = new voicefile(testbase[i].name);
     }

     i=0;
     while ((!_fstrchr(testbase[i].code,(int) pair[1])) && (i < numofelem))
     { i++;} //search index of testbase that matches pair[1]
     if (i >= numofelem)
     {
           cout << "Element 2 of pair " << i+1 << " does not exist.\n" << flush;
           exit(-1);
     }
     else
     {
           if (DCC2)
             delete DCC2;
           DCC2 = new voicefile(testbase[i].name);
     }

     outportb(0x378,0x0); // leds answer box off
     channel = LEFT;
     FastPlay();
     channel = RIGHT;
     FastPlay();

     outport(0x378,0xC); // leds 2 and 4 on
     do {
     button = jstatdcc();}
     while (!( (button == 0xB0)||(button == 0x70) ));
     switch(button)
     {
           case 0x70:
             outfile << pair << " 2   ";
             outportb(0x378,0x4);
             cout << "Van paar: " << ipair+1 << "  heeft het tweede fragment de voorkeur.\n";
           break;
           case 0xB0:
             outfile << pair << " 1   ";
             outportb(0x378,0x8);
```

```
             cout << "Van paar: " << ipair+1 << "  heeft het eerste fragment de voorkeur.\n";
            break;
            default:
            break;
    }
    // put a \n after each 8 tested pairs in the outputfile
    if ( !((ipair+1)%8))
            outfile << "\n" << flush;

  } //for ( ipair = 0; ipair < numofpairs; ipair++ )

  // End of test
  cout << "Is 'tie goed of is 'tie nie goed?\n" << flush;
  outportb(0x378,0x0); //leds off
  outfile << "END." << flush;
  setfile.close();
  outfile.close();
};

void main(void)
{
    sbsetup();
    doublevoice dccloader;
    dccloader.OpenFiles();
    dccloader.Test();
    return;
}
```

Appendix IV

Instruction for listening test

Instructieblad voor de luistertest

In deze luistertest krijgt u steeds twee verschillende bedieningsgeluiden aangeboden. Het is de bedoeling dat u steeds aangeeft welke van de twee bedieningsgeluiden u het meest aanspreekt.

U start elk experiment zelf door op de rode knop voor u te duwen. De linker lade zal opengaan en na 1 seconde ook weer dichtgaan (u hoeft dus niet nogmaals op de rode knop te duwen). Als de linker lade gesloten is, drukt u weer op de rode knop. De rechter lade zal nu open-en dichtgaan. Nadat de rechter lade gesloten is, zullen op het antwoordkastje de twee rode lampjes gaan branden. Nadat u uw keuze hebt gemaakt, door op een van de twee knoppen te duwen, zullen de beide lampjes uitgaan en kunt u een nieuw experiment beginnen door op de rode knop te duwen.

Appendix V

Plots of time signals of the stimuli

(Description in Table 4.2)

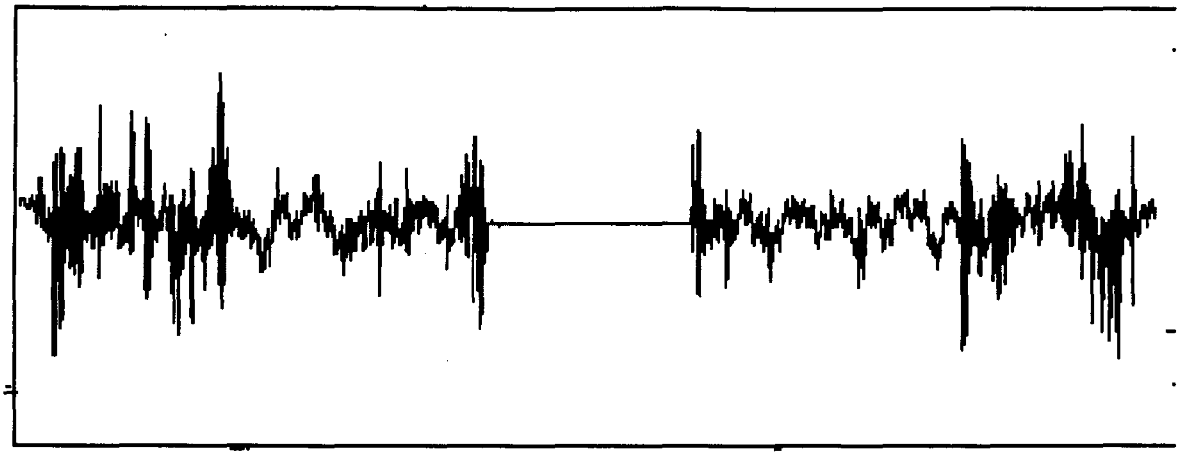


Fig.1 Stimulus A

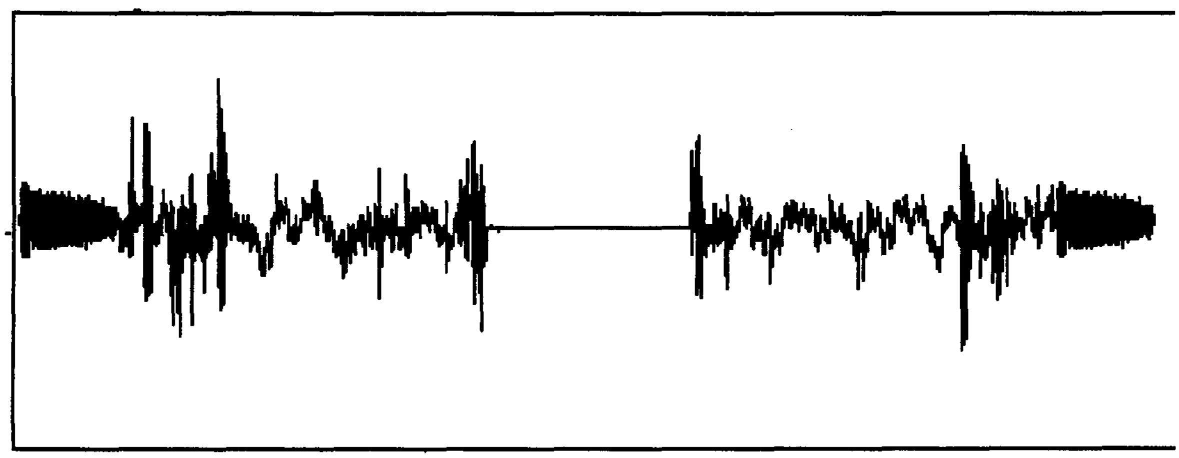


Fig.2 Stimulus B

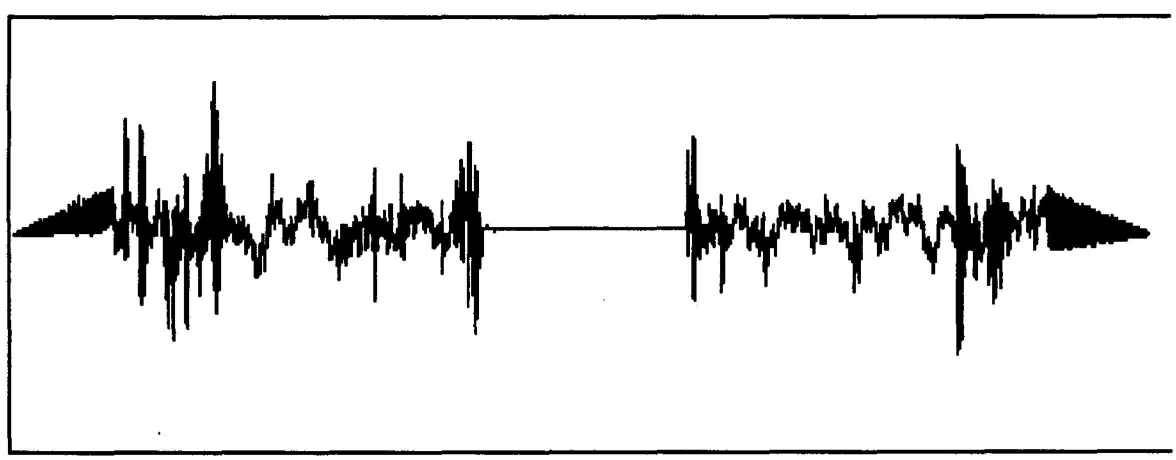


Fig. 3 Stimulus C

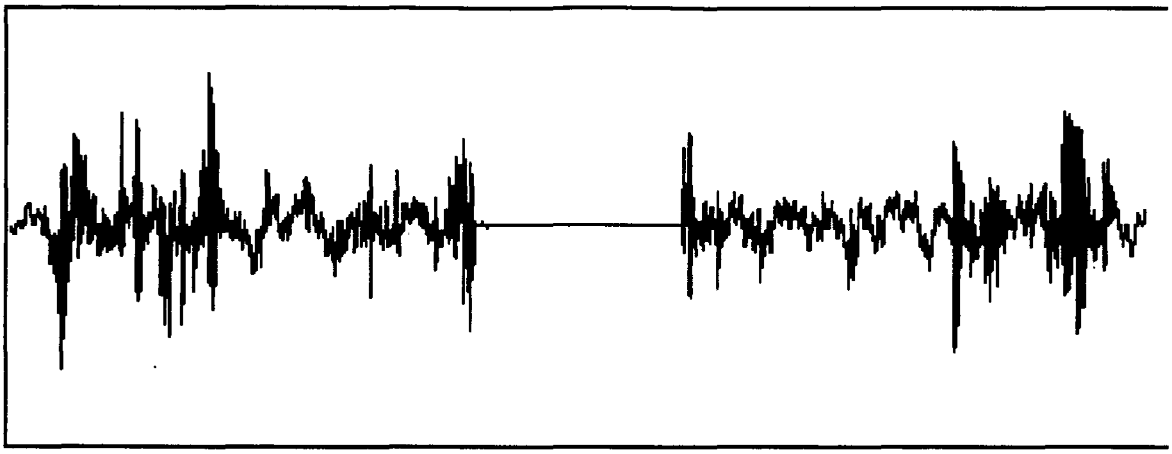


Fig. 4 Stimulus D

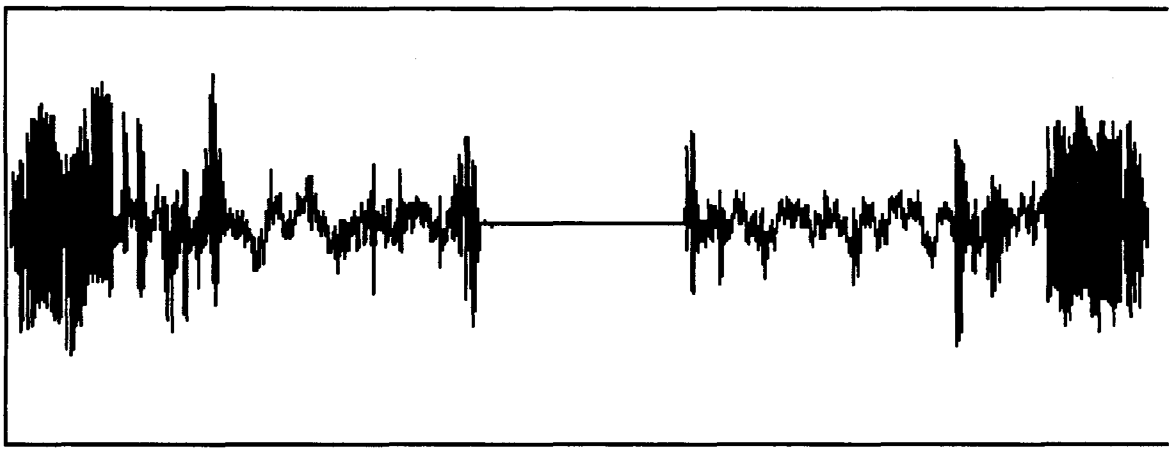


Fig. 5 Stimulus E

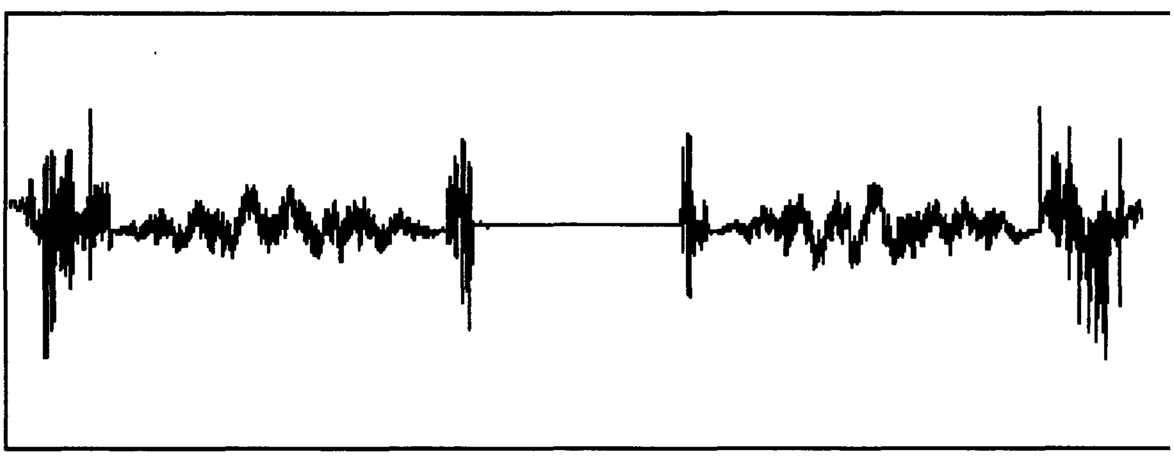


Fig. 6 Stimulus F

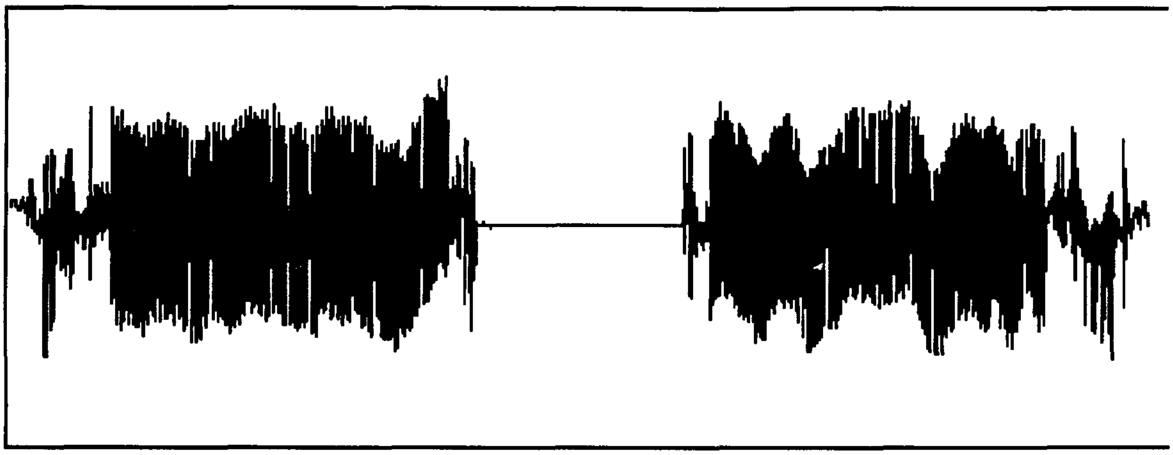


Fig. 7 Stimulus G

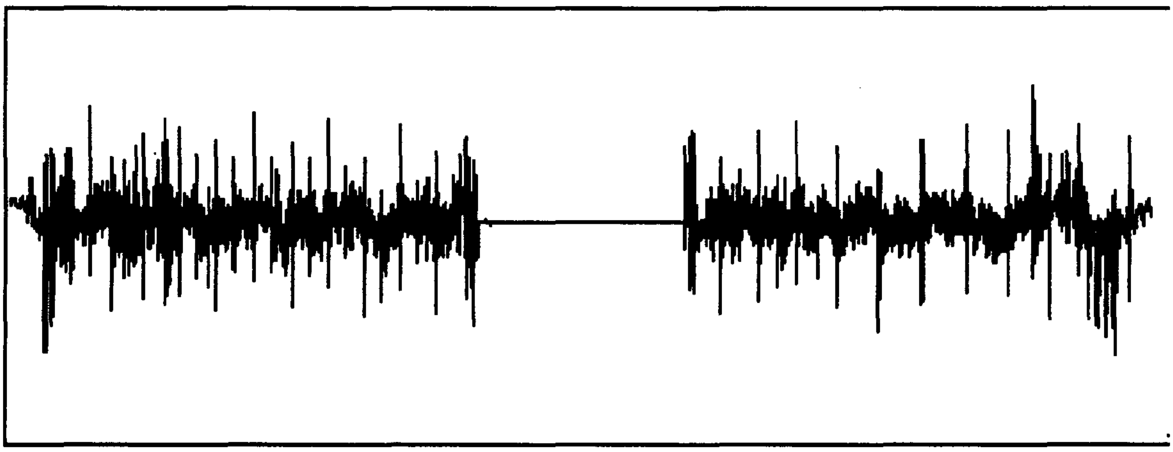


Fig.8 Stimulus H

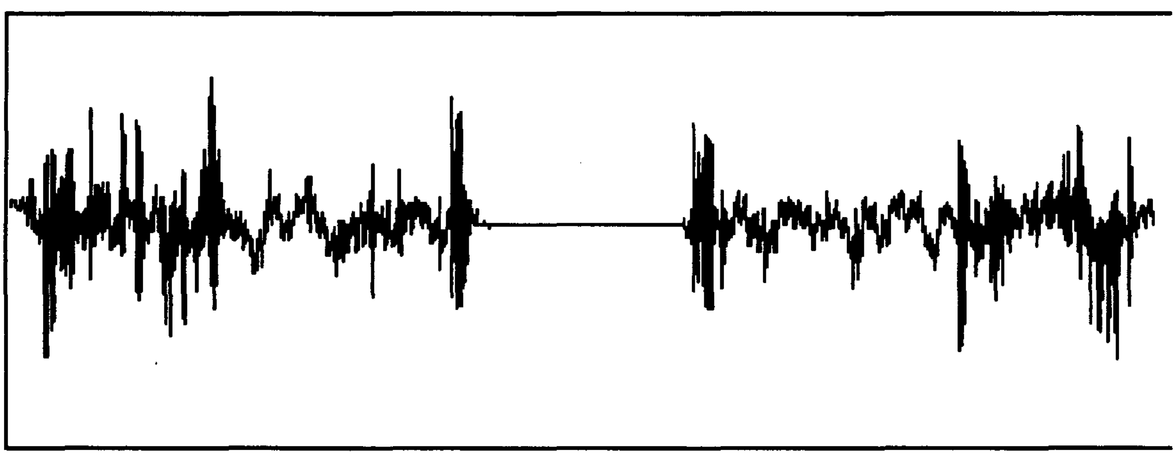


Fig.9 Stimulus I

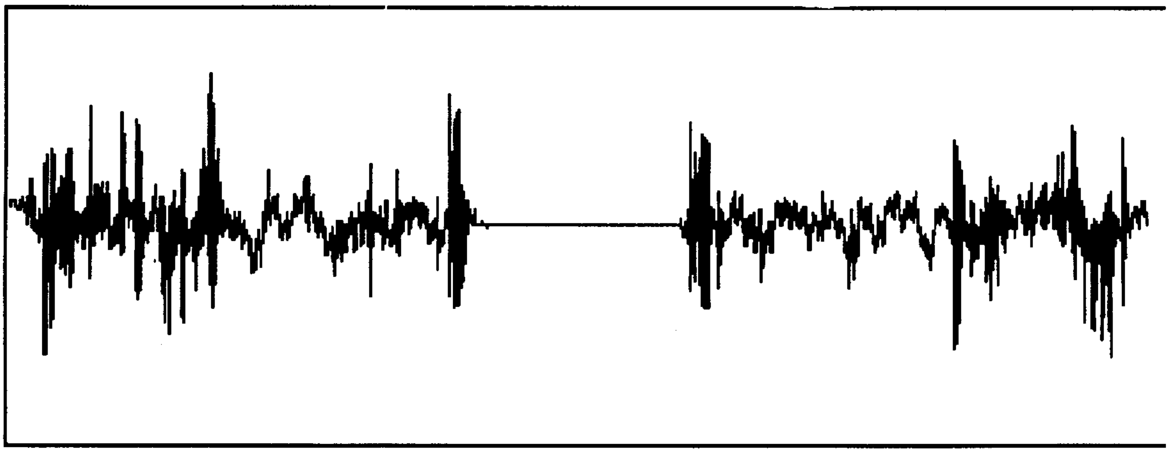


Fig.10 Stimulus J

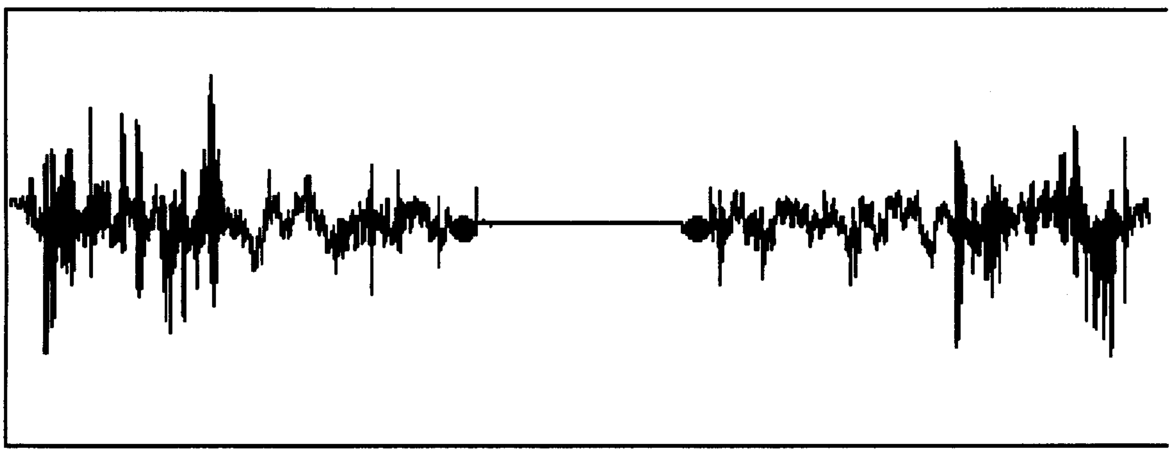


Fig.11 Stimulus K

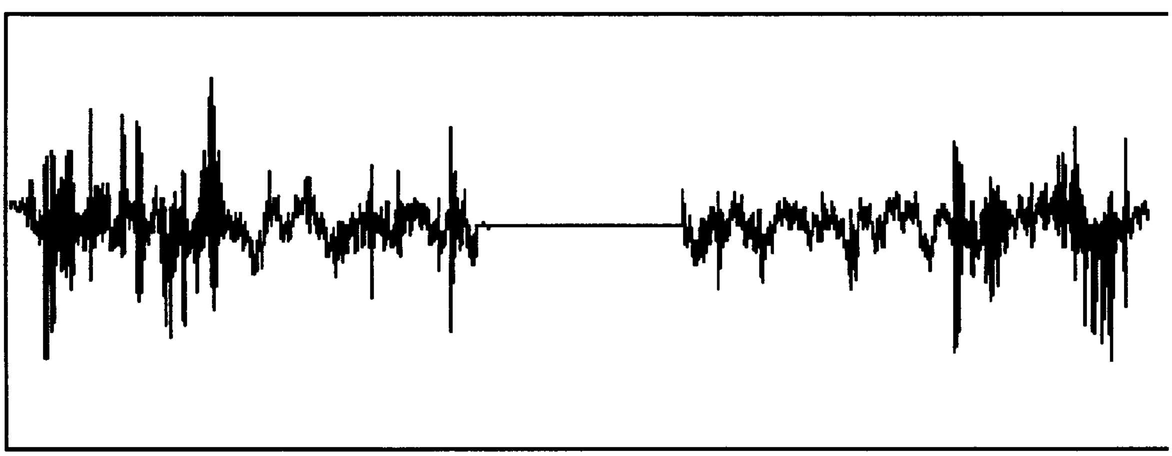


Fig.12 Stimulus L

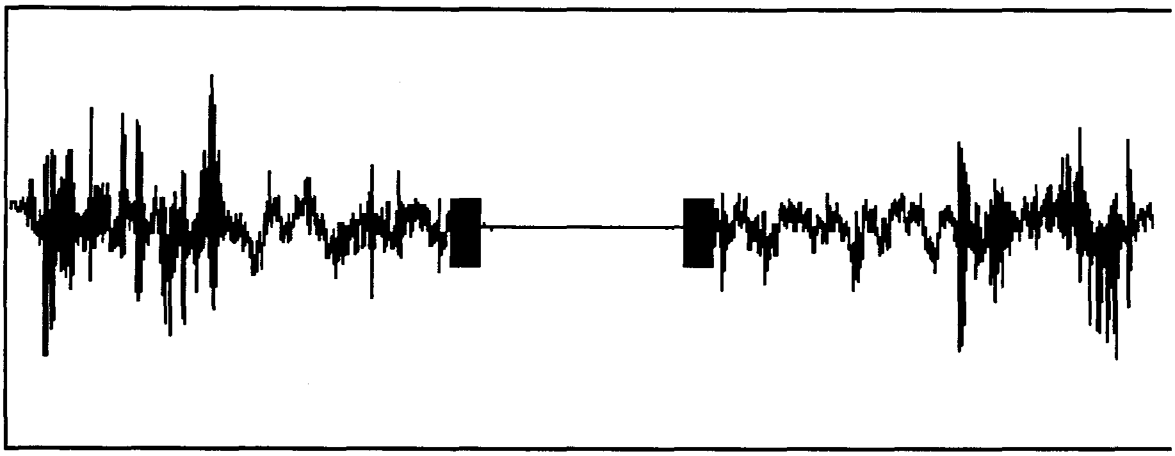


Fig.13 Stimulus M

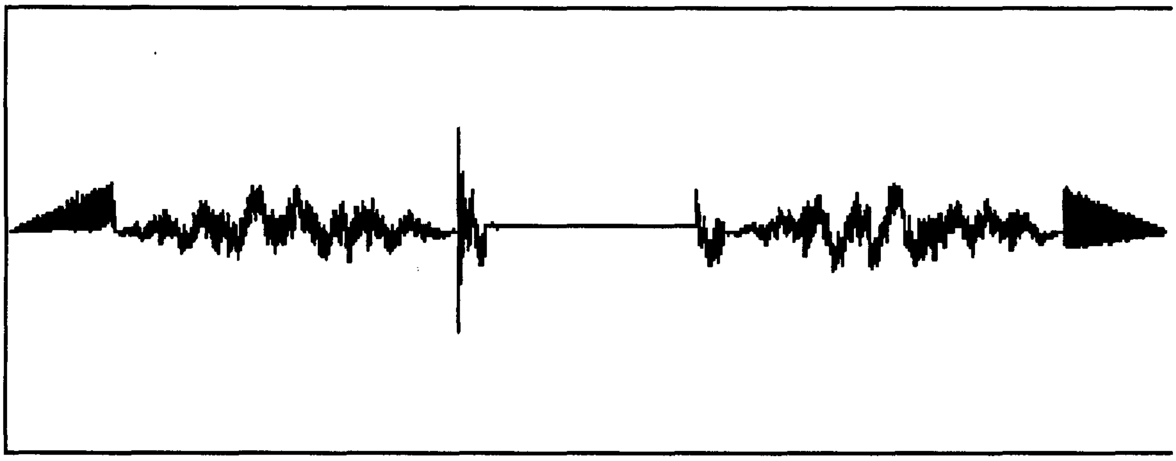


Fig.14 Best representatives of each category combined

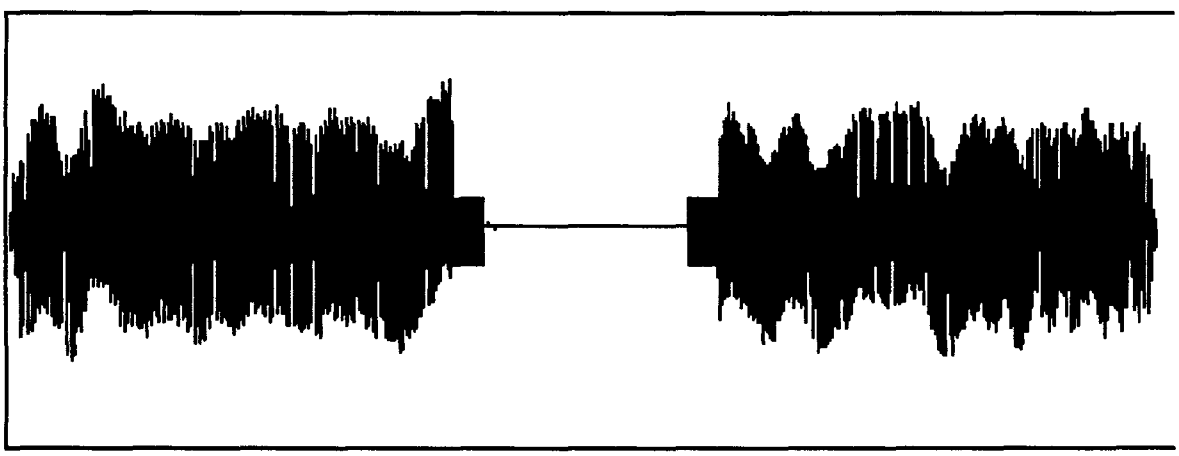


Fig.15 Worst representatives of each category combined