A computer-based Speech Tracking procedure

Gnosspeius, J. and Spens, K-E.

journal: STL-QPSR
volume: 33
number: 1
year: 1992
pages: 131-137

http://www.speech.kth.se/qpsr
A COMPUTER-BASED SPEECH TRACKING PROCEDURE

John Gnosspelius & Karl-Erik Spens

Abstract

A computer program was written to facilitate the administration of the Speech Tracking procedure described by De Filippo & Scott (1978) and to make different speech tracking tests more comparable. A record was kept of the total number of blockages, the number of blockages resolved outside the system, the average time spent on blockages, the average time spent on non-blocked words, and the average tracking speed. Experience with the system indicates that special care should be taken in the choice of text as well as in the partitioning of the text into segments.

INTRODUCTION

Continuous Discourse Tracking (CDT), a method for training and evaluation of technical aids for deaf persons, was introduced by de Filippo & Scott (1978). The sender reads the text from a novel, phrase by phrase. The receiver’s task is to repeat the phrases verbatim. If the receiver can do this, the reader goes on to the next phrase but if s/he cannot repeat the phrase, a blockage has occurred (Owens & Telleen, 1981). A blockage can be resolved by several different means such as repeating the blocked word or by presenting a synonym to the blocked word (De Filippo & Scott, 1978). There are also more severe kinds of blockages. That is, those which cannot be resolved within the tested communication system and have to be transmitted in another way, such as sign language or writing.

During comparison of several different speech tracking tests obtained with relatively similar tactile aids by different research teams, it was found that the results of these tests varied considerably (Spens & al., 1992). One reason for this was found to be the difference in the amount of time allowed for the correction of blockages. This is one uncontrolled factor in CDT. Other factors that also can influence the test results are the difficulty of the text and the variation in lip-reading ability.

The many positive elements in CDT: good face validity, unlimited amount of test material, ease of administration, ease of keeping the motivation to participate high, have resulted in a theoretical analysis of the tracking procedure with the aim of finding methods to better standardise the procedure (Spens & al., 1992). They show that the tracking score can be described by the formula below.

\[ L = \frac{L_c}{(1 + W_{pb} \cdot (k - 1))}, \quad k = t_b \cdot L_c \]

The parameters in the formula are the maximum tracking rate (the ceiling rate) within the system, ceiling rate (Lc), the proportion of words in blocked phrases (Wpb), and the average time spent on a blocked word (tb). The formula shows that if a long time is allowed to resolve blockages and the number of blockages is low, the average tracking speed will be very sensitive to small changes in the number of blocked words.

In Fig. 1, the impact that a difference in these parameters has on the average tracking speed is shown. The upper line is the one with the lower k-value. If there are few blocked words, that is, an easy text is used, the slope will be steep and the
average tracking speed will vary a great deal with a small variation in the number of blocked words. The average tracking speed will vary even more if a large amount of time is spent on correcting blockages. This effect will cause great differences in the relative measure (the improvement aided/unaided).

\[
L = L_c/(1 + W_{pb}(k-1)) \text{ words/min.}
\]

from above, \(k=2,3,5,8\) and 12

![Graph](image)

Fig. 1. The average tracking speed, \(L = f(L_c, W_{pb}, k)\). Upper curves have lower \(k\)-values. Dotted curves have \(L_c=50\) words/minute.

**A COMPUTER-ASSISTED TRACKING PROCEDURE**

It is very difficult to compare tracking scores if the parameters in the formula are unknown. To facilitate comparisons, it was decided to write a computer program that controlled each of the parameters. As one difficulty in comparing results lay in the difference in the time spent on correcting blockages, the aim was to minimize this time and also to make it uniform throughout the test. The method used is based on the tracking procedure described by Fenn & Smith (1987).

An important factor that should be controlled in speech tracking is the length of a text segment presented to the receiver. If it is too long, it will be hard to remember and, thus, the number of blockages will be high. On the other hand, if the segments are too short, the maximum tracking speed (\(L_c\)) will be low because of a high rate of turn taking between the receiver and the sender. In the system we have developed, the text is divided into preselected phrases and stored in the computer. The speaker reads a phrase from the computer screen. The length of a phrase is determined when it is stored. Thus, the influence of variations in phrase length caused by the speaker will be eliminated.

Only repetitions are used if a word is not correctly identified by the subject. Each time a blockage that cannot be resolved inside the system occurs, blocked word should be presented to the subject with a minimal time loss. Fenn & Smith do this by stopping the timing, writing down the blocked word and then resuming the test session. In our system, instead of stopping the watch when showing the blocked word, it is assumed that the transfer of a word to the receiver is short, and that every transfer requires the same amount of time. This means that the timing is never stopped during a test. The number of allowed repetitions before the word is to be
transmitted to the subject has to be determined before starting the test. In this way, the time spent on correcting a blockage, which cannot be resolved within the number of repetitions determined in advance, can be estimated and will be uniform throughout the test. It is assumed that a blockage is composed of a single word.

The program can be used on both simulated deaf subjects (noise-masked normal-hearing subjects) and real deaf patients. The same small time loss to resolve blockages that cannot be transmitted with the tested communication system will be obtained, since the transmission of a blocked word is done visually.

The words per minute (wpm) score used is the number of read words, divided by the elapsed time, in minutes (De Filippo & Scott, 1978).

\[
\text{average tracking speed} = \frac{\text{total number of words}}{\text{total test time}}
\]

The tracking speed will depend on the number of blockages as well as on the time allowed for resolving them (Spens & al., 1992). Therefore, it is very important that the determined maximum number of repetitions per blocked word does not vary.

The following data are stored in a file at the end of each session: the number of words in each phrase, the time spent on each phrase, the number of repetitions of the phrase or a word in the phrase, the phrase with indication of blocked words transmitted outside the tested communication system, the total number of words transmitted outside the system, the total number of words, the test duration, and the average tracking speed. All the figures in the result file are presented in columns. This makes it very convenient to copy the information into a spreadsheet program (in our case Excel) for further treatment.

The amount of information obtained with the system makes it possible to perform a number of calculations. Since each phrase is preceded by the time that was spent on it, the maximum tracking speed for the test system ($L_C$) can be calculated from the phrases that contain no repetitions.

The blocked phrases and the time spent on these phrases are given. From this, the proportion of words in blocked phrases ($W_{pb}$) can be calculated. The factor $k$ is obtained as the average time spent on words in blocked phrases divided by the average time spent on words in non-blocked phrases.

**Some practical details**

It is important to place the screen on which the blocked word transmitted outside the communication system appears, close to the sender so that the receiver won’t have to divert her/his view too much. This is easily achieved if an LED display is used instead of an ordinary computer screen. The display can be positioned directly above the reader’s head and, thus, only very small movements of the subject’s eyes are required. Also, the display and the reader’s lips would be at the same distance from the subject. This means that no refocusing of the subject’s eyes is necessary.

Text files used for the test must be stored without format characters, e.g., written in Microsoft Word for Windows and saved as DOS Text w/line breaks. Each line should contain no more than 73 characters. Otherwise the end of the line will not be visible. Lines that contain comments must begin by a left parenthesis. These lines are shown during the test but omitted in the result file.
Results of a pilot study

To gain experience from the system, a short test was performed with one normally-hearing subject that listened to low-pass filtered speech with cut-off frequencies at 250 Hz, 500 Hz, and 1000 Hz in combination with lip-reading. Ten test sessions of 10 minutes each were conducted. The first session was considered to function as training and was discarded. The following nine were performed in random order. Then three more were performed without any aid. A maximum of two repetitions was allowed before the blocked word was visually presented to the subject.

Preliminary results indicate that only a few test sessions are required for $L_e$ to reach its final value (Fig. 2). It is possible to keep $k$ at a fairly constant level. A high proportion of blocked words ($W_{pb}$) seriously diminishes the average tracking speed ($L$). The maximum speed within the system ($L_e$) also decreases. This is caused by the experimenter who wishes to increase the number of correctly interpreted words by speaking more clearly (Picheny, Durlach, & Braidia, 1986). The proportion blocked words that were presented visually to the subject ($W_{pbo}$) is low except in extremely difficult situations.

![Graph](image)

Fig. 2. Results of twelve test sessions arranged according to the aid that was used. Tests with identical aids are arranged in chronological order.

A short description of the computer program

The program was written in Turbo Pascal 6.0 using the object-oriented features of the language. One main program and four sub modules, containing global declarations, file-handling procedures, the code necessary to send a word to another screen, and the procedures that handles the administration of blockages were written. Another file containing the code used to send a word to a LED display was written. If it is desired to use the LED display, one line in the main program must be changed and the program has to be recompiled. The program uses a text file containing user-defined input parameters. If this file does not exist, the system creates it and names it `person.txt`. A text file that holds the test results is also created.
As the computer program is started, a dialog box appears on the screen (Fig. 3). The questions in it must be answered before anything else can be done. The questions can be answered by using a mouse or the tabs key to move to the desired square. The high-lighted characters in the commands are shortcut keys and can be entered from the keyboard. The subject's name, the aid that will be used, the text to read, and the duration of the test are entered. Also, a question as to what windows are to be displayed on the screen is answered.

Fig. 3. Start dialog box.

When this is done the screen is cleared and several new windows are presented (Fig. 4). One of them displays the text. Depending on the answer, above up to three additional windows appear. These are windows that show the blocked words that cannot be conveyed within the system, the number of such words, and the elapsed time. A start button appears in the middle of the screen.

Fig. 4. Start of the tracking session.
When this button is clicked, the timing is started and the tracking test session begins. The sender reads from the text and the receiver repeats what is said. For each repetition, the experimenter presses the right arrow key and when the allowed number of repetitions is obtained, he clicks with the left mouse button on the blocked word. That word is then displayed in the list window and sent to another screen or a LED display in front of the subject. The active line, that is, the line currently being read, is high-lighted in green. With the right mouse button, the experimenter can move the green bar to the next line and thus activate it. Only words in the active phrase may be clicked upon. To the left of the active bar, the number of repetitions that have been made in the current line is shown. This goes on until the pre-set time has elapsed. Then the test is stopped and the experimenter is asked to click with the right mouse button on the last read word. At this point, a window with the results of the test is displayed and the test session is completed.

---

**Result file.**

If desired, the test can be restarted by pressing the function key F2 or by clicking on Restart. During a session, the test can be halted by pressing F3 or by clicking on Halt. A dialog box where two buttons are available then appears. These buttons are used either to resume the test session or to stop the test and get the results obtained so far. The program is exited by pressing Alt-X or by clicking on Exit.
used either to resume the test session or to stop the test and get the results obtained so far. The program is exited by pressing Alt-X or by clicking on Exit.

The results of a test are presented in a file (Fig. 5) which is named for the receiver. If, for example the subject's name is Johan Gnosspelius, the result file will be named johan_gn.doc. It is divided into two parts. The head contains the date, the time, the patient's name, the aid used during the test, and the name of the text file that was used. The second part contains the actual results of the test session.

Also the last line read is stored. The results of the test session are appended to the results of the previous sessions. The second time the program is executed, the result file is used to start the test at the proper place in the text, that is at the first new sentence following the last line of the previous test.

REFERENCES


