A computer-driven system for assessing speech quality and intelligibility

Christian Cavé, Bernard Teston, Alain Ghio, Philippe Di Cristo

ACUSTICA · acta acustica
Vol. 84 (1998) 157 – 161

Technical Notes

A Computer-Driven System for Assessing Speech Quality and Intelligibility

Christian Cavé, Bernard Teston, Alain Ghio, Philippe Di Cristo
Institut de Phonétique d'Aix-en-Provence, Laboratoire "Parole et Langage" ESA 6057, CNRS, 29 Avenue Robert-Schuman, 13621 Aix-en-Provence, France. E-mail: cave@romarin.univ-aix.fr

PACS no. 43.71.Gv, 43.58.Ta

Introduction

Assessing the quality or intelligibility of speech requires the use of psychophysical methods. Indeed, none of the current physical methods is adequate for assessing the wide variety of speech types and speech transmitting and listening conditions. Even the most objective methods must first be validated by comparison with subjective techniques [1, 2], so the administration of perception tests to panels of listeners is still indispensable.

This type of testing is very time-consuming, however, not only because several speakers and listeners must be used to compensate for individual variability, but also because measurements must be repeated many times to allow for statistical validation of the results. Hence there is a need to automate as many operations as possible such as stimulus generation, procedure control, and data analysis.

The computer-driven system we are presenting here meets these objectives. It is not possible in this technical report to present the theoretical foundations of the procedures implemented in the system, nor the different methods used during development. Readers interested in these issues may wish to refer to Calliope [3, 455-488], Santi [4, 88-135] and Pisoni *et al.* [5].

1. Description of the system

1.1. General description

The system consists of a PC-type microcomputer connected to the necessary hardware and software for testing eight subjects at a time. The system performs the following operations:

- Generation and presentation of auditory stimuli to subjects;
- Message display on individual screens facing subjects;
- Recording of subjects' responses;
- Automatic control of all testing procedures;
- Storage, preliminary processing, and presentation of results.

These operations are driven by programs developed in the Microsoft Windows environment [6]. The system takes full advantage of the environment's multimedia and audio capabilities and its powerful spreadsheet and database management tools.

1.2. Basic operation

Each subject is in front of a monochrome alphanumeric screen where the test instructions and response choices are displayed, a response box with five buttons, and a set of earphones. The testing procedure consists of sending an auditory stimulus to the subjects, while a list of possible responses is displayed on the screens. The system supports different types of tests. At the present time, three tests are permanently installed (see 3.4). Subjects respond by pressing one of the five buttons on the response box (see 3.4). Response time is recorded.

Received 15 November 1995,
accepted 23 April 1997.

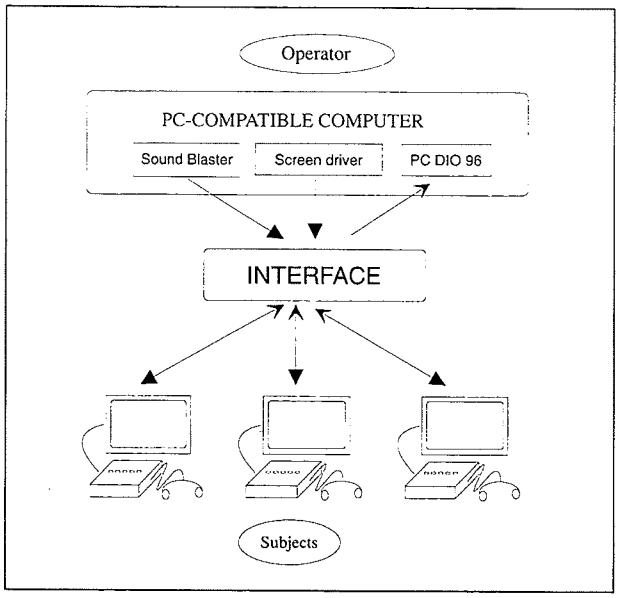

Figure 1. System hardware configuration.

In its current version, the system tests up to eight subjects at a time, although the number of response stations can be extended to sixteen. Testing is collective, which saves a considerable amount of time and ensures more uniform experimental conditions than a setup where subjects are tested one by one. Responses are recorded automatically, and subsequent processing requires only minor data manipulation.

2. System hardware configuration

2.1. Basic architecture

The system has the following components (Figure 1):

- A PC-type 486 or Pentium computer, which drives the entire system. It is equipped with an SVGA graphic screen, 8 Mb or 16 Mb of read/write memory, and a 500 Mb hard disk (or larger) for storing speech samples.
- Three specialized boards:
  - A Hercules monochrome screen driver,
  - a National Instruments PC-DIO-96 numeric input/output board,
  - a multimedia Sound Blaster 16 SP audio board.
- A custom-designed interface, which serves as a link and distribution point between the monochrome screens, response boxes, and earphones.
- Eight response boxes.
- Eight monochrome screens.
- Eight headsets.

2.1.1. Sound files

The sounds originate in a Sound Blaster 16 SP audio board that supports 8- and 16-bit mono- or stereo-encoding, and sampling frequencies of 11, 22, and 44 kHz. The stimuli presented to the subjects are saved in sample file format in mass storage. At testing time, the stimuli are sent to the audio board where they are converted into analog signals for transmission to the headsets after volume amplification.

ACUSTICA · acta acustica
Vol. 84 (1998)

The sample files can be acquired directly with the Sound Blaster board either through the microphone input or by hooking up a tape recorder or D.A.T. to the line input. The stimuli can also be read off other types of storage devices (CD ROM, magnetic tape, disk, etc.), provided the imported files are converted into .WAV format. This format complies with RIFF standards (Resource Interchange File Format) developed by Microsoft [7, 8]. Files in .WAV format contain the samples proper, and a header describing the characteristics of the signals (sampling frequency, number of bytes per sample, mono/stereo mode, etc.).

2.1.2. Display source

The Hercules monochrome screen driver generates the alphanumeric messages presented to the subjects, which include the test instructions and response choices. To use the screen driver, a protected memory address must be stored in the Windows environment. A resident program developed by our laboratory ensures indirect access to the necessary memory addresses [9]. This technique makes it possible to simultaneously control the subjects' monochrome screens and the operator's standard SVGA color monitor.

2.1.3. Response recording

Eighty channels are needed to simultaneously manage up to sixteen subject stations and five response choices. The channels are controlled [10] by a National Instruments PC-DIO-96 board with 96 parallel digital I/O's equipped with four AMD8255A programmable peripheral interfaces (PPI). Each input on the board is connected to a push button on a response box. Regular scanning of the PC-DIO-96 board inputs detects which buttons have been pressed on the response boxes.

2.2. Control-station / response-station interface

All computer-driven experiments require an interface between the physical world to be captured and the numerical environment of the computer. For the present application, we developed an interface that is connected to the control station and to each of the response stations (Figure 1). It performs the following functions:

- Audio processing.
  Amplification and control of the Sound Blaster line output sound level is achieved by a power amplifier capable of supplying 16 headsets. The volume is kept constant at ± 0.2 dB regardless of the number of headsets connected to it. It is equipped with a calibrated attenuator for adjusting the listening volume.
- Video processing.
  In standard configurations, the video connection between a Hercules alphanumeric board and a monochrome monitor cannot exceed a few meters. Furthermore, the output only supports one screen. To solve these problems, the interface in our system includes 16 video line amplifiers to enable direct control of each output line to the display screens with no specific limit on the length of the connecting cords.
- Recording subjects' responses off the push button response boxes. This operation is straightforward. The only necessary precautions consist of processing potential rebounds generated by the long connecting cords and eliminating any electromagnetic interference that might trigger false detections.

2.3. Response stations

Each response station includes a response box, a headset, and a monochrome screen. Two-directional cords link each station to the interface described above. Travelling through these cords are (a) the audio and video signals going from the interface to the stations, and (b) the subjects' responses moving in the opposite direction. Electronic processing by the interface allows for cords several tens of meters long, so the response stations can be set up in a large room.

Beyer type BT 100 electrodynamic earphones are used. They were chosen for their durability, high sensitivity and response curves, and low impedance dispersion which eliminates the need for individual calibration.

Video screen displays are used to transmit alphanumeric data to subjects. Philips BM7 monochrome monitors were chosen to achieve the best tradeoff between the high resolution needed for displaying messages [11] and the cost of the equipment.

The response boxes are metal cases with five push buttons. The boxes also serves as relay points for connecting the subject's headset and the monitor. The push buttons are short stroke, sharp contact, fast action microswitches especially suited to measuring reaction times.

3. System software

The computer software was developed in the Microsoft Windows environment [12, 13], chosen for it user-friendliness, widespread use, and compatibility with other software packages. Our laboratory's extensive experience in measuring speech intelligibility, and recent large-scale measurement projects [14, 15], have enabled us to design a system that is both highly flexible and very efficient due to the many automated operations.

3.1. Hardware checks

The system requires the use of a wide variety of hardware including screens, headsets, response boxes, cords, the interface, specialized boards, the computer, and the power supply. Despite the reliability of the equipment chosen, there is always a risk that some component will break down. Likewise, the various devices may be improperly connected. To detect such problems, the system offers a number of utility programs for checking the hardware and connections before a run. Utilities are also available for verifying the auditory installation, the entire video distribution system, and each of the push buttons on the subject response boxes. These diagnostic aids have proven very useful.

3.2. Parametrization of testing conditions

During software development, we attempted to render the system as flexible as possible by allowing user entry of function parameters. A pre-experiment phase is used to define the system configuration (number of response stations, which ones will be occupied, push button initialization, testing conditions, subject data, etc.). It is also possible to specify the directories where the speech sample files are stored, and to indicate where the subjects' responses are to be recorded. Utility programs allow the operator to choose a standard or personalized configuration, and to save the selected configuration. For instance, the operator might want to initialize the system for several groups of subjects, or to validate response buttons 1 and 5 only for a test with binary responses.

3.3. Script files

Tests are controlled by scripts, a technique already used in the S.O.A.P. system [16]. In fact, the system's core software resembles that of a lexical-syntactic analyzer which reads and interprets a script file sent to it. The execution sequence is modified in accordance with the commands received. This type of operation provides a great deal of flexibility and supports a wide variety of tests.

Each script line (Figure 2) contains a keyword followed by an argument specifying what operation to perform. Here again, parametrization offers maximum flexibility in test execution:

- An instruction file can be chosen for multilingual applications.
- The file names where subject responses will be automatically recorded can be entered (see 3.7).
- The test type can be specified as either N-uplets, preferences, or scales (see 3.4).
- The time limit for responding can be set.
- The time interval between each stimulus can be defined.
- The place where the system should start measuring response time (beginning or end of stimulus) can be specified.
- Random presentation of stimuli and response choices can be selected.

```
CONDITIONS        helium.180m.tourelle.repetition1   => testing conditions
INSTRUCTIONS      C:\perceval\bin\consigne.txt       => instruction file
TYPE_TEST         1                                  => test code[1]
RESULTS           C:\perceval\reponses\he180T1       => response files[2]
WAITING TIME      2000                               => authorized waiting time after listening (in ms)
PAUSE             1000                               => pause between two presentations (in ms)
START                                                => start scan
#—                Scan
STIMULUS          C:\PERCEVAL\STIMULI\pile.wav       => sound file to present to subjects
DISPLAY           1) bile 2) pile                    => character string on screens
RESPONSE          5[3] 2[4] pile.bile[5]             => response scoring
CORRECT           2                                  => number of the correct response
SCORING           +Vois -Vois                        => error scoring (voiced/voiceless)
CONTEXT           _i                                 => non-variable (context /i/)
—
STIMULUS          C:\PERCEVAL\STIMULI\vingt.wav
DISPLAY           1) vingt 2) bain
RESPONSE          1 2 vingt.bain
CORRECT           1
SCORING           -Inter +Inter                      => error scoring (continuous/interrupted)
CONTEXT           _~E                                => non-variable (context /ɛ̃/)
—
...
—
END                                                  => end of test
```

Figure 2. Example of a script file. [1] type 1 = N-uplet test. [2] Each subject has a response file. File he180T1.REZ is the response file of a subject whose initials are EZ, he180T1.RCA is the file of a subject whose initials are CA, etc. [3] stimulus number. [4] number of response choices (2 = pair, 3 = triplet, 4 = quadruplet, etc.). [5] list of response choices.

3.4. Test types

Given that testing is collective, adaptative procedures cannot be used. In addition, since there are five buttons on the response boxes, the tests must be designed for at most five responses. Aside from these constraints, there are no limitations on the type of test that can be administered, nor on the way a given test is run. A specific script file can always be defined for controlling execution in the desired fashion. Three types of tests are currently available.

- N-uplet tests.
  The subject hears a stimulus and then several response choices are proposed on the screen. For example, the word "*baffe*" is heard and "*basse*", "*baffe*", and "*bache*" are proposed as responses. The correct response is "*baffe*". Among the various possible N-uplet tests, the diagnostic rhyme test [17, 18] and the quadruplets test [14] are permanently on-line.
- Preference tests.
  The subject hears several stimuli, for example, three versions (synthesis A, B, or C) of a sentence like "*It'll be a nice day tomorrow.*" The subject must indicate which of the three he/she prefers by pressing on the corresponding button.
- Scale tests.
  The subject hears a stimulus. Several ratings are proposed on the screen. For example, the subject must rate the stimulus on a scale from 1 to 5, or state whether it is good, average, or poor.

3.5. Error scoring

In the case of an N-uplet test where there is only one correct response, the errors made by the subjects can be scored. To facilitate data analysis, errors can be classified by type, context, etc. For classification purposes, the operator must provide the system with a complete list containing not only the various competing choices but how they should be coded. The coding method is as follows:

```
word1     word2
code1     code2
context.
```

Example for a diagnostic rhyme test.

```
pile      bile
-vois     +vois     ⇒ Voiced/voiceless
_i                  ⇒ context /i/
vingt     bain
-inter    +inter    ⇒ continuous/interrupted
_~E                 ⇒ context /ɛ̃/
```

3.6. Measuring response time

Response time can be a crucial parameter for assessing task difficulty and interpreting results [19]. The system measures response time to the nearest millisecond.

The 'Multimedia' capability of Windows [7, 8] ensures response time measurement accuracy. The 'mmsystem' library can be used to set the access time on the computer's internal clock. It is thus possible to obtain resolutions below 50 ms, which is the baseline period. The microprocessor acts on the clock by means of an interrupt whose role is to interrupt the current action in order to execute a priority operation (scan the response boxes in our case). The use of a system interrupt guarantees that the operation one wishes to execute will occur at regular fixed intervals. This ensures accurate and reliable response time measures.

3.7. Results

For each subject, a response file is generated containing the necessary information for processing the data. For a quadruplet test, for example, the following information is stored: script name, testing condition codes, subject's name, number of quadruplets, number of response choices, response choices, correct response, response given by subject, response scoring system (no response = 0, correct response = 1, incorrect response = −1), response time, and optionally, the error code and context (Figure 3).

ACUSTICA · acta acustica
Vol. 84 (1998)

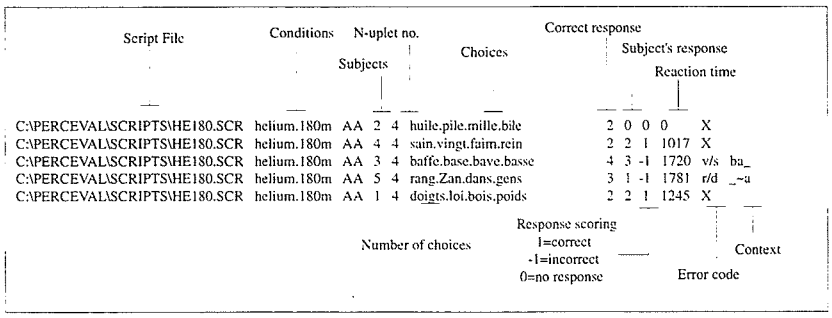

Figure 3. Example of a response file.

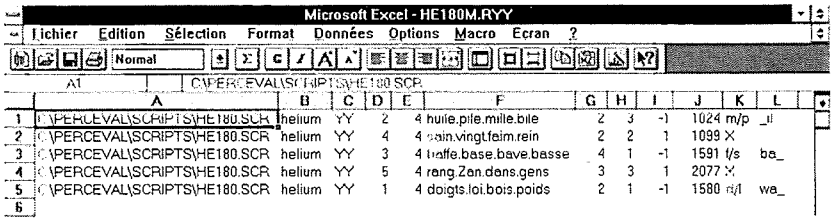

Figure 4. Excerpt from a response file imported into Excel.

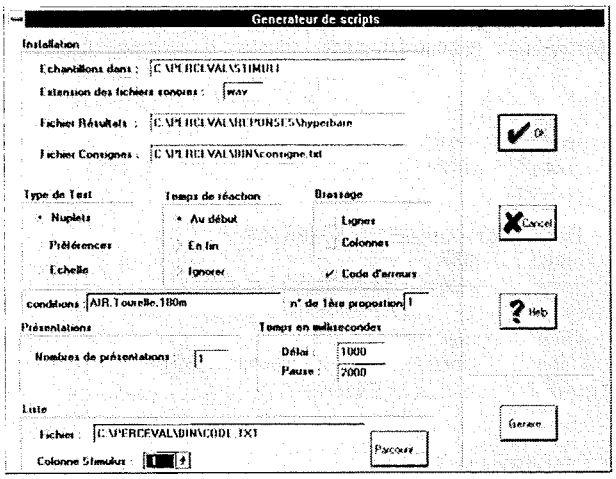

Figure 5. Script generation window.

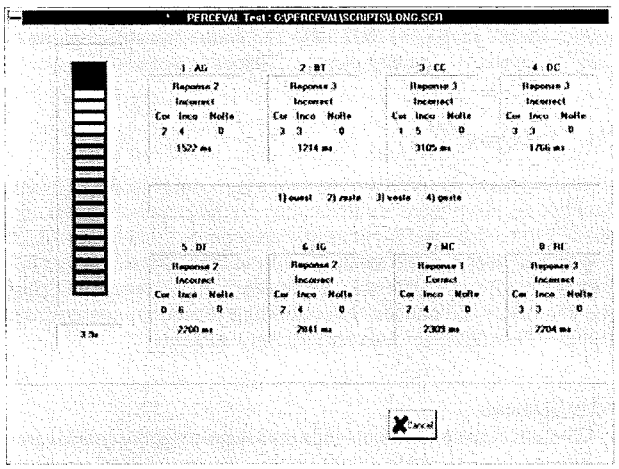

Figure 6. Test control window.

The response files are filled up automatically during testing. They are written in table format to facilitate importation into spreadsheets like Quattro Pro or Excel (Figure 4) or statistical analysis packages like Statview or Systat. All standard types of processing are possible, in addition to more detailed analyses. Some examples are counting the errors made on item 10 for all subjects pooled in such and such a testing condition, or using the error scoring capability to determine the number of errors the voicing feature in the [i] context under a given testing condition.

4. System operation

4.1. Running a test

The system software operates like any other Windows application. There is a run icon, a main window, information windows, pull-down menus, and dialog boxes. The main window supplies various utilities for ensuring proper execution: hardware checks, testing sessions configuration, script file design (Figure 5), run procedures, presentation of results, etc.

During execution, the operator has a control screen (Figure 6) with a bar graph indicating the time elapsed, a central area reproducing the information displayed on the subject screens, and eight boxes (one per station), each showing the corresponding subjects' name, response given, total number of correct, incorrect, and non-responses, and the response time. This information is displayed in real time and helps the operator detect potential failures in the equipment or inattentiveness on the part of the subjects. In the 16-station configuration, the screen can be switched back and forth from stations 1–8 and stations 9–16.

At the end of a testing session, the operator can request some raw results such as the overall percentages of correct, incorrect and no-response items. Further analyses may then be conducted using the data processing programs (see 3.7).

Note that it is possible to generate stimuli and then save them in mass storage off the system. Removable disks or backup tapes are recommended for transferring the data from the system to other storage devices. Similarly, for certain system operations, some of the equipment is not necessary (interface, specialized boards, response stations). Script generation and data processing can also be done elsewhere.

4.2. Successful projects

The system has already been implemented in several speech quality and intelligibility assessment projects. In a recent campaign on the intelligibility of coded speech, the system successfully collected nearly 750,000 responses, an undertaking that would not have been feasible otherwise. The system has also been used regularly to evaluate speech intelligibility in underwater and hyperbaric environments [14], and to evaluate speech unscramblers [15].

5. Conclusion

The computer-driven speech assessment system we developed can be used to evaluate speech quality and intelligibility. It was designed to allow for collective testing and to automate the various operations involved in setting up experiments, presenting stimuli, and recording subject responses. Powerful stimulus and response preprocessing programs simplify the analysis of the results obtained. The system has proven to be effective in several large-scale measurement projects (see 4.2). The experience acquired during these projects has been applied to enhancing the performance and reliability of the current version of the system (release 2.0). Because of its ability to rapidly collect massive amounts of data, the system can be used in studies requiring either large corpora (e.g. synthetic speech evaluation systems, TTS) or many subjects (e.g. studies on inter-individual variability in speech decoding).

References

[1] H. J. M. Steeneken: Comparisons among three subjective and one objective intelligibility test. Report in IZF 1987-88, pp 33. TNO, Institute of Perception, Soenteeberg, 1987.
[2] L. Morellini: Etalonnage d'une mesure objective d'intelligibilité par des tests subjectifs. Actes du 2ème Congrès Français d'Acoustique, Suppl. Journal de Physique III **2** (1994) C1, 339–342.
[3] Calliope: La parole et son traitement automatique. Masson et CNET/ENST, Paris, 1989.
[4] S. Santi: Synthèse vocale des sons du français: Modélisation acoustique et évaluation perceptive. Dissertation. Université de Provence, 1992.
[5] D. B. Pisoni, H. C. Nusbaum, B. G. Greene: Perception of synthetic speech generated by rule. Proceedings of the I.E.E.E. **73** (1985) 1665–1676.
[6] B. Levingston, M. Levingston: Windows 3.1 shareware and freeware. IDG Book, San Mateo, 1993.
[7] Microsoft: Multimedia programmer's workbook, chap. 3-4. Microsoft Press, Redmond, 1991.
[8] Microsoft: Multimedia programmer's references, chap. 8. Microsoft Press, Redmond, 1991.
[9] R. Brown, J. Kyle: Pc interrupts. Addison Wesley, Reading, 1991.
[10] A. Gupta, O. M. D. Toong: Insights into personal computers. Elsevier, New York, 1985.
[11] W. O. Galitz: Handbook of screen format design. North-Holland, Amsterdam, 1987.
[12] C. Petzolt: Programming windows. Microsoft Press, Redmond, 1990.
[13] P. Norton, P. Yao: Borland c++ programming windows. Barton Books, Los Angeles, 1992.
[14] C. Cavé, A. Marchal, C. Meunier: Constitution d'un test d'intelligibilité pour la parole hyperbare. Final report of DRET contract No. 90-1201, 1992.
[15] C. Cavé, P. Di Cristo, A. Ghio, A. Marchal, C. Meunier: Intelligibilité de la parole subaquatique et hyperbare. Final Report. Contracts FSH/CEP&M 4737/90 and M4737/91, 1994.
[16] Rapport SAM: Speech output assessment package (SOAP), V. 4.0. Esprit Project No. 2589, 1992.
[17] J. P. Peckels, M. Rossi: Le test de diagnostic par paires minimales. Adaptation au français du 'Diagnostic Rhyme test' de Voiers. Revue d'Acoustique **27** (1973) 245–262.
[18] M. Cartier, M. Rossi: Le test de diagnostic par paires minimales: mise en oeuvre et résultats. Actes du Symposium Intelligibilité de la Parole, GALF, Liège, 1973. 191–208.
[19] M. Talbot: Reaction time as a metric for the intelligibility of synthetic speech. – In: Speech and language based interactions with machines. J. A. Waterworth, M. Talbot (eds.). Ellis Horwood Ltd., 1987, 66–88.