OPTICOM

OPERA

YOUR DIGITAL EAR!

User Manual

Version 3.5

Document Version as of 20 Dez. 02

Contents

SOFTWARE LICENSE AGREEMENT AND LIMITATIONS

General

Conditions deviating from these General Contract Conditions shall not be deemed valid unless we have confirmed them expressly in writing. Verbal agreements are not valid unless the obligation to confirm such agreements in writing has been renounced by mutual agreement and in writing.

We shall carry out delivery and installation of the goods and machines as well as instruction of the operating personnel at the expense of the buyer. Consulting on application and usage shall be given to the best of our knowledge, based on our experience. The goods and machines delivered are subject to change. Changes in design and/or shape shall be accepted by the buyer, unless these changes are not deemed fundamental modifications substantially limiting the purpose of the purchased goods as agreed.

The buyer shall take responsibility for the lawful usage of our machines as stipulated in the laws, rules and stipulations applicable.

Complaints Concerning Defects

Any complaints concerning deficiencies in quality and performance or the delivery of the correct number and types of goods agreed, that can be determined by reasonable efforts shall be filed promptly with the buyer in writing within fourteen days after delivery. Hidden or latent faults shall be notified to the buyer promptly after discovery. In case of any complaint the buyer shall on our request undertake to promptly send back the goods concerned in their original packages.

Software License

The accompanying software to this OPTICOM product is licensed, not sold. OPTICOM hereby grants the user of the OPERA Software (herewith referred to as 'Licensee') with respect to the Licensed Patents, Licensed Trade Marks and the OPERA Software a non-exclusive, non-transferable, non-assignable, non-sublicensable limited right to Use the licensed number of copies of the OPERA Software, solely to facilitate the objective quality evaluation of audio signals in accordance with the respective Standard; provided that:

Licensee shall not disable any copy protection mechanism of the OPERA Software provided by OPTICOM; and Licensee shall prohibit any additional copying of the OPERA Software in whole or in part, other than the number of licensed copies and other than it is essential for the proper operation of the OPERA software or for normal security back-up purposes;

Licensee shall not modify, translate, reverse-engineer or de-compile the OPERA Software except to the extent permitted by law;

Licensee shall not resell, sublicense or otherwise redistribute the OPERA Software;

Licensee shall maintain the OPERA Software in confidence and ensure that it is protected from unauthorized copying or disclosure by measures that are no less stringent than those it uses to protect its own valuable information and that are, in any case, no less than reasonable in the circumstances;

Except as expressly granted, Licensee shall have no other rights in the OPERA Software. For the avoidance of doubt the rights granted shall not include a license to modify, have modified, create and/or have created derivative works of the OPERA Software or any Algorithm in Source Code form, and to make, or have made copies of an Executable version other than the licensed number of copies of the OPERA Software. Under no circumstances will anything in this Agreement be constructed as granting, by implication or otherwise, a license to any other technology owned and/or licensed by OPTICOM other than the licensed OPERA Software.

In such a case as either party vests any patent rights in any enhancements and new features in the OPERA Software, the relevant party, upon request of the other party shall offer to the other party a license with a scope similar to the license obtained by Licensee against fair, reasonable and non-discriminatory terms and conditions.

Limitations, Warranty[1], Liability

1. OPTICOM shall retain all right, title and interest in and to the OPERA Software, subject to the license granted. Licensee shall be entitled to establish all proprietary rights for itself in the intellectual property represented by enhancements and new features, created by Licensee, whether in the nature of trade secrets, copyrights or patent rights or other rights. OPTICOM shall be entitled to establish all proprietary rights for itself in the intellectual property represented by enhancements and new features, created by OPTICOM, whether in the nature of patent rights or other rights.

Nothing herein shall constitute or be construed as:

a requirement that OPTICOM shall file any patent application, secure any patent or maintain any patent in force, or

an obligation on the part of OPTICOM to furnish any technical information, technical support, software of any kind or any information concerning pending patent applications of OPTICOM.

2. OPTICOM warrants that at the purchase date it has full power and authority to grant Licensee the rights granted herein and that it has no knowledge of any pending legal procedures regarding the Licensed Patents. OPTICOM does not warrant and shall not be liable for the existence of such disposal subsequent to the coming into force of this Agreement.

[1] shall mean "Maengelhaftung" according to the German Civil Code of 01. January 2002

3. OPTICOM warrants that the OPERA Software along with the Accompanying Hardware works according to this documentation ("Users Manual") which is part of the delivery and that the OPERA Software properly implements the relevant measurement Algorithm in accordance with respective recommendations.

4. Although all software has been designed and controlled with due care, it has to be assumed that it steadily undergoes a development process. Should any programming errors be discovered, and reported to OPTICOM in writing, then OPTICOM, within twelve months from the delivery date, shall be obliged to correct the deficiencies as far as prescribed by the warranty. OPTICOM shall remedy defects of the Licensed OPERA Software along with the Accompanying Hardware that may occur. The obligation to correct errors in compliance with the warranty granted, is limited to the correction of errors. Any such defects of the OPERA Software along with the Accompanying Hardware shall be repaired by replacing the software by a new version, or by replacing the system by a new hardware component, which shall be delivered by OPTICOM without undue delay.

5. If within a reasonable period of time the defective OPERA Software along with the Accompanying Hardware will have undergone a replacement twice without success, Licensee has the option of demanding a reduction of the price to be paid or the return/termination of the delivered items. Further warranties are expressly excluded, e.g. the Licensee is not entitled to claims based on warranty,

a) if the deficiency has been caused by improper usage of the OPERA Software along with the Accompanying Hardware, faulty installation, usage of unsuitable accessories or improper operation, or faulty or incomplete programming by licensee, or in case of any modification to the OPERA Software along with the Accompanying Hardware carried out by Licensee or a third party;

b) if the OPERA Software along with the Accompanying Hardware will not have been maintained or serviced in accordance with our recommendations and this has caused the deficiency;

c) if the deficiency has been caused by improper modification of the OPERA Software along with the Accompanying Hardware;

d) if the damage has been caused by an Act of God, e.g. damage by lightning;

e) if the deficiency results from normal wear and tear, especially as far as working parts are concerned.

6. Such claims according to 6.2 – 6.5 shall become statute-barred one year from the date on which such claims arose, or upon expiry of the legal prescription period, whichever period is shorter.

7. OPTICOM's liability and the liability of it's legal representatives and those employed in fulfillment of the Agreement, arising from breaches of contract or tort is limited to cases of intent, gross negligence or recklessness, lack of warranted quality (characteristics) and violation of a material responsibility, which would jeopardize the contractual objectives.

8. For each individual case of damage, such liability shall be restricted to the foreseeable typical damage OPTICOM had to expect when contracting the Delivery in consideration of the circumstances known. The liability for slight negligence shall not exceed an equivalent of US$ 300.000,-.

9. Licensee shall not be entitled to claim any damages against OPTICOM, including damages for indirect loss, e.g. missed profit, impossibility of performance, positive violation of a contractual duty or failure to perform. In case of a loss of data, OPTICOM shall only be liable for the expenses required to reconstruct the lost data using backup files duly created in regular intervals.

10. Any claims arising from Product Liability Law shall not be affected by the aforegoing terms of this Article.

In the event that any afore-mentioned terms or conditions are found to be invalid, unlawful or unenforceable to any extent, this shall not effect any other terms and conditions agreed herein. The parties shall endeavor to agree to such amendments which shall in as far as possible effect the economic intentions expressed therein. In the case of a gap of these terms this shall apply accordingly.

Applicable law and Place of Jurisdiction

If the customer is either a merchant entered into the commercial register, or a legal person under public law, or a Special Fund under public law, Erlangen shall be agreed as the place of jurisdiction. In all other cases, the legal place of jurisdiction shall apply. This agreement shall be construed under and governed by the laws of the Federal Republic of Germany.

PREFACE

A Brief How-to Guide to this Manual and How to Get More Information and Support.

We are delighted to welcome you as our new customer. As you might know from our company vision - **quality is our business**. Especially when it comes to signal quality of speech and wide band audio signals. Our new **OPERA™ system**, which is an abbreviation for **O**bjective **Per**ceptual **A**nalyzer, reflects a new generation approach, integrating both the latest experience and standards in the research and development of perceptual based, objective methods for the determination of signal quality.

We will introduce the basic operation of OPERA™ and give some guidelines on the most common applications in this manual. This manual, however, is not meant to substitute research reports, papers and standards documentation. Where appropriate, we will refer to the corresponding literature indicated by a number in squared brackets.

How this Manual is Organized

This manual is organized in seven chapters.

Chapter 1, "Introduction", will briefly explain the OPERA™ measurement. You will learn about the ideas behind this measurement tool and get an overview of the OPERA™ product family.

Chapter 2, titled **"Test Methodology"**, provides you with the necessary basic knowledge about perceptual measurements. This chapter is meant to guide you to the correct measurement for all the applications you might want to use your OPERA™ system with.

Chapter 3, "Installation and Setup", will guide you through the installation procedure and inform you of the hardware requirements if you are using either the OPERA™ Software Suite or the OPERA™ Toolkit. It is important to read this chapter carefully when you unpack and install your OPERA™ system for the first time.

Chapter 4, "Getting to Know The OPERA™ Framework", will outline the basic concept and explain the operation of the framework program. The framework is a summary of the functionalities, which you will always need, regardless of specific measurement algorithms.

The next two Chapters, **Chapter 5, "Wide Band Audio Quality Testing"**, and **Chapter 6, "Telephony Band Voice Quality Testing"**, deal with the specific measurement setups and methods for both principle applications. Depending on the options installed in your OPERA™ system, you may need to refer to one or the other, or both chapters.

Chapter 7, Automation and Programming explains how to use OPERA from the commandline and outlines other possibilities of automating tasks in OPERA.

These chapters are followed by the **"Technical Specifications"**, the **"References"**, a **"Glossary"** and finally the **"Index"**.

In the **Appendix** you will find some background information, such as a collection of papers and articles along with the relevant standards documentation.

Conventions Used in This Manual

In this manual we will use some conventions in order to ease the understanding of the operation. For instance,

- all **menu options** that can be selected will be in **bold style**,
- all command line parameters that you might need to type will be printed in the **`courier`** `type style`,
- basic command line keywords will also be in **`courier bold`** `type style`.

You will also find a lot of figures displaying screen shots. Please note that due to the ongoing development and software update process, the screen shots of your system might differ slightly from the examples in this manual. In the case of extreme differences, please do not hesitate to inform us.

SYMBOL-LEGEND
Information
Important Hint
Menu Options
Enter parameters
see References

Quite often you will experience gray shaded symbols next to the text paragraphs. The legend on the left explains the meaning of the most commonly used symbols. They will point out passages in the manual containing information, important hints, basic menu options and command line parameters. The little "book symbol" indicates that more detailed information is available in the papers, books or articles cited in the references.

Further Information and Support

For all questions arising from the use of OPERA™ or that might be related to the interpretation of measurement data, please make sure to refer to this user manual and the relevant standards documents. In case the information given will not be sufficient to answer your question, you can visit our on-line support section, available from our website:

http://www.opticom.de

We specifically recommend the support section of our website for a report of known bugs and problems. This section will be available soon, and should help you to easily check if you encounter an unknown problem.

In case you would encounter a bug or a problem, which is not yet listed on the support section of our web page, please make sure to contact the OPTICOM support with a detailed bug report.

Note:

In case of hardware problems with your PC-workstation please refer to the hardware documentation of the OEM manufacturer first. All OPERA™ products will be based on well supported standard hardware PCs that will be supported world-wide through the original manufacturer. The OEM manufacturer will be able to help you in the case of hardware problems related to your PC, for instance if the system would not boot anymore, in the case of a hard disk crash, or when encountering problems with the power supply unit. The same applies to the monitor. OPERA portable systems are directly supported by OPTICOM.

To find out your nearest support contact for the **OEM PC hardware**, please see the support offices section of the accompanying hardware documentation.

For all other problems, please contact OPTICOM, and ask for your local support representative.

Chapter 1

1 INTRODUCTION

An Introduction to the OPERA™ Measurement System and the OPERA™ Product Family.

After reading this chapter you will be familiar with the basic ideas and the concept behind this tool. The OPERA™ product family will be outlined at the end of this chapter.

1.1 What is OPERA™?

Compression has become state-of–the-art technology in modern communications – thus allowing for a great number, diverse and inexpensive new components of the information age, such as: mobile phones, VoIP, MP3 internet audio, radio and TV satellite networks, DAB, DVD, and many more. On the other hand the economic benefit of lowering data rates to a minimum is contradictory to clear sound. In spite of "all digital technology“, sound quality and the intelligibility of speech have become issues again, and are of much more impact than in those "good old analog days".

Our new generation of quality testers, called OPERA™ – short for "Objective Perceptual Analyzer“ – represent the latest development in objectively evaluation and assure the quality of compressed speech and wide-band audio signals by modelling the human ear: OPERA™ is your digital ear. OPERA™ is not only suitable to assess a single processing device, with OPERA™ you can achieve a comprehensive analysis of the end-to-end quality, from the studio source to the receiver, or from the caller to the callee. And because OPERA™ works quite similar to the human ear, it is able to distinguish between imperceptible, and more or less annoying transmission errors.

Other than traditional measurement methods (like S/N, THD+N), the new OPERA™ system is able to simulate the subjective evaluation of human subjects. The analysis is based on the most recent perceptual techniques, such as PEAQ, PESQ and PSQM. As a major advantage, OPERA™ employs the same kind of natural stimulus for a measurement as in practical operation: human speech or music program material. Moreover, this makes it possible to monitor the quality during network operation. Cultural and language differences may be taken into account by the evaluation as well. As a consequence of the novel approach to measure the perceived audio quality instead of signal characteristics, it is possible for the first time to gain an objective quality metrics which truly characterizes the quality of service („QoS“) of a network.

OPERA's flexible scalability may range from a single stand alone tester up to powerful network-wide setups with distributed systems sharing information over TCP/IP. OPERA™ may be used interactively as an analyzer, or runs fully automated according to a predefined schedule.

The open framework concept of OPERA™ allows the addition of advanced measurement algorithms as plug-ins in the future as soon as they will become available. In addition, user defined measurement algorithms may be integrated upon request.

Basically, there are two different versions of OPERA™, a Telecom Version and a Broadcast Version. Some of the features of the **Telecom Version** are at the time:

- ITU-T P.862/PESQ
- ITU-T P.861/PSQM
- PSQM+ (PSQM improved for GSM)
- Echo measurement with real speech
- Delay measurement
- Interfaces to file (*.wav), loop start (a/b), E1/T1, [VoIP, and wireless to follow]
- ...

Some of the features of the **Broadcast Version** are at the time:

- ITU-R BS.1387/PEAQ
- Delay measurement
- Real time measurement
- Interfaces to file (*.wav), analog XLR balanced (20 bit) and digital AES/EBU
- ...

All standard measurement algorithms are based on the reference code, which was used for the standardization, and all algorithms are tested and verified to be fully conforming to the standards.

OPERA™ is available as a software version, a completely pre-installed portable system and a completely pre-installed rackmount system. The Workstation version is not available as a standard product anymore. In addition, we offer custom tailored and OEM solutions.

1.2 Getting to Know the OPERA™ Product Family

1.2.1 OPERA™ Measurement System

The OPERA™ Measurement System comprises both hardware requirements and software. It comes completely pre-configured. It is available as a portable and a rackmounted version. The rackmounted version is fully compatible with the portable version and not described separately. In addition there exists a worstation type version which is not available anymore, but still supported and described in this manual.

Workstation Version
The basic configuration of the Workstation Version of OPERA™ includes a completely equipped high performance PC system with selected components and the OPERA™ framework software. If assessment of speech quality is required, the Telecom configuration is provided. Audio quality can be measured using the broadcast configuration. Combinations of telecom and broadcast configurations of the OPERA™ Measurement System are also available.

For a detailed description of what is included in the delivery, please refer to Section 3.2.1.

Figure 1.1 shows a photograph of the OPERA™ Workstation.

Figure 1.1: The Workstation version of OPERA™

Portable PC Version
The basic configuration of the Portable PC Version of OPERA™ includes a completely equipped portable high performance Dual Processor PC system with selected components and the OPERA™ framework software.

As with the Workstation Version, there are different configurations available, the Telecom version for speech quality assessment, and the Broadcast configuration for measuring audio quality. Both configurations come pre-configured for live measurements, the corresponding required interfaces and cables will be provided. Of course assessing prerecorded files is possible too. For a detailed description of the delivery , please refer to Section 3.2.2. Again, combinations of Telecom and Broadcast Configurations of the OPERA™ Measurement System are available.

Figure 1.2 shows a photograph of the OPERA™ Portable.

Figure 1.2: The Portable PC version of OPERA™

1.2.2 OPERA™ Software Suite

OPERA™ is also available as a Software Suite, which is delivered without any hardware. A description of the corresponding hardware requirements can be found in the "Installation and Setup" chapter in Section 3.1.2 and in the "Technical Specifications" chapter.

In contrast to the Workstation or the Portable PC Version of OPERA™, the Software Suite supports file based measurements only. Live measurements are not supported. Measurement functionality and performance will be the same as for the hardware based OPERA™ system in the file based mode. However, processing time may vary with the available processing power of the PC.

Note:

Even if there is an audio board installed in your system, no live measurements are possible with the Software Suite.

Chapter
2

2 TEST METHODOLOGY

Essential Knowledge about Perceptual Measurements and a Guidepost to the Correct Measurement Method.

This chapter describes listening test methods which are modeled by the OPERA™ system, and a description of the underlying concept of the proposed algorithms for perceptual measurement and the international standardization of perceptual audio measurement techniques. In addition, this chapter will provide a guide to the correct measurement method for your applications.

2.1 Assessing Quality

Until recently, the only widely accepted assessment procedures for audio or speech codecs were listening tests, due to the lack of international standards for measuring the perceived audio quality.

ITU-T P.800

Historically related to the assessment of telephone connections, useful methods for testing telephone band speech signals were first standardized within the ITU-T[1]. Recommendation P.800 [ITUT800] defines the absolute category rating test method (ACR) which has been used for the assessment of speech codecs since 1993. Within the ACR test method, the ITU five grade impairment scale is applied (see **Table 2.1**). In the telecommunication environment, testing is done without a comparison to an undistorted reference. This copes with a typical situation of a phone call, where the listener has no access to a comparison with a reference, e.g. the original voice of the other party. However, it should be noted that the listening test according P.800 could be regarded as a comparison between a test signal and a reference "in the mind" of the listener. The reason for this is that the listener is very familiar with the natural sound of a human voice.

For comparison reasons, and in order to be able to merge the results of different individuals, it is necessary to adjust the listeners' opinions to an absolute scale. For this purpose, predefined examples with well defined noise insertions of fixed modulated noise reference units (MNRU, [ITUT810]) are presented at the beginning of a test. Each sample represents an example distortion corresponding to the ITU-T version of the five grade impairment scale.

[1] International Telecommunication Union, Geneva, (former CCITT), see also http://www.itu.org

Impairment	Grade
Excellent	5
Good	4
Fair	3
Poor	2
Bad	1

Table 2.1: The ITU-T five-grade impairment scale

Based on these test conditions a population of typically 20 to 50 test subjects will be presented with an identical series of speech fragments. Every test subject will be asked to score each sample by applying the impairment scale. After statistical processing of the individual results, a Mean Opinion Score (MOS) can be calculated. With thorough setups, such test results can be reproduced quite well, even at different locations. It goes without saying that the effort needed in terms of subjects and time is tremendous. It is clear that such test methods can not be applied within a practical or field environment in the daily life.

ITU-R BS.1116

The ITU has also recommended a test procedure to assess wide band audio codecs on the basis of subjective tests. Subjective assessments of low bit rate audio codecs in the past always targeted an almost transparent quality. For this reason, the test method focuses on the comparison of the coded/decoded signal to the unprocessed original reference. The relevant recommendation is known as BS.1116, titled "Methods for the Subjective Assessment of small Impairments in Audio Systems including Multichannel Sound Systems" [ITUR1116] which was issued by the ITU-R[2] in 1994 and was updated in 1997.

The test method, which is recommended by BS.1116, is referred to as "double-blind triple-stimulus with hidden reference". It is extremely sensitive and allows for the accurate detection of small impairments. The grading scale used should be treated as continuous with "anchors" derived from the ITU-R five-grade impairment scale according to ITU-R BS.562 [ITUR562]. It is depicted in **Table 2.2**.

Impairment	Grade	SDG
Imperceptible	5.0	0.0
Perceptible, but not annoying	4.0	-1.0
Slightly annoying	3.0	-2.0
Annoying	2.0	-3.0
Very annoying	1.0	-4.0

Table 2.2: The ITU-R five-grade impairment scale

The analysis of the results from a subjective listening test is generally based on the Subjective Difference Grade (SDG) and is defined as:

$$SDG = Grade_{\text{Signal Under Test}} - Grade_{\text{Reference Signal}}$$

Provided that the listener correctly assigns the hidden reference signal, the SDG values will range from 0 to –4, where 0 corresponds to an imperceptible impairment and –4 to an impairment judged as very annoying. The assignment of the SDG scale is shown in the last column in **Table 2.2**.

[2] Radiocommunication Sector of the ITU (former CCIR)

In contrast to the listening test according to ITU-T P.800, an explicit comparison between the test signal and a reference signal is needed in the case of BS.1116, since the listener never knows how the original signal sounds.

This method was applied in a variety of international verification tests in the past. However, keep in mind that because of the scope of BS.1116 it can be applied to small impairments only, which means a practical limitation to almost "transparent" studio quality. Another issue which has been discussed among experts, is the recommendation to use the scale at a resolution of one decimal place, resulting in 41 (!) discrete steps. There are indications that for some subjects this is too great a choice, and furthermore the meaning of the impairment anchors is interpreted differently [KARJ85].

Because of the restrictions to small impairments, there is consensus among experts that other methods are needed for very low bit rate tests (i.e. of large impairments). Various approaches have been introduced and work is currently in progress in several task groups, e.g. the MPEG standardisation work [GILC96]. The methods according to ITU-T P.800 were adopted for some assessments to overcome the problem of a "gap" for a useful recommendation on testing significantly impaired wide band audio signals. While in principle they seem to be better suited for impaired music signals when compared to the BS.1116 method, it's exploitation for very low bit rate audio coding applications still remains questionable, as there are no clearly defined example distortions in such a case. The scale was derived from telephone speech quality, and is not well defined when translated to music coding. The achieved results may therefore significantly depend on the subjective interpretation of the impairment levels.

MUSHRA

At the time of drafting this manual, an advanced listening test procedure has been advised by an EBU expert group, known as "MUSHRA". MUSHRA stands for "Multiple Stimulus With Hidden Reference Anchors". The new method targets testing significantly impaired audio signals, such as those derived at very low bit rates. MUSHRA is expected to become adopted as an international recommendation by the ITU working party 10-11Q. As soon as more experience has been gained, OPTICOM will provide its customers with more information at **www.peaq.org**.

2.2 Advanced Audio Measurements Employing "Perceptual Modeling"

Assessing the quality was a pending issue during the years of the development of compression schemes. Consequently, the idea of substituting the subjective tests by objective, computer based methods has been an ongoing focus of research and development. Early work motivated through the development in speech coding was reported in [KARJ85]. Since then several methods were introduced.

The underlying concepts of the proposed algorithms for perceptual measurement techniques are all quite similar. The common structure of these algorithms is depicted in **Figure 2.1**. The process of human perception is modelled by employing a difference –measurement-technique which compares both, a reference signal (i.e. the "input" signal to a codec) and a test signal (i.e.

the "output" signal of the codec). First, the algorithms process an ear model for the reference and the test signal, in order to calculate an estimate for the audible signal components. The result can be imagined as the "internal representation" inside the human auditory system. Comparing the internal representations of the reference, and the test signal leads to an estimate of the *audible difference*. To derive an overall quality figure, this information, which is a function of time, must be processed accordingly, like the human brain of a subject would do in a listening test. The respective part of processing within an algorithm is referred to as *cognitive modelling*. A total quality figure will be derived as the final result, which can be compared to a MOS ("Mean Opinion Score") resulting from a listening test.

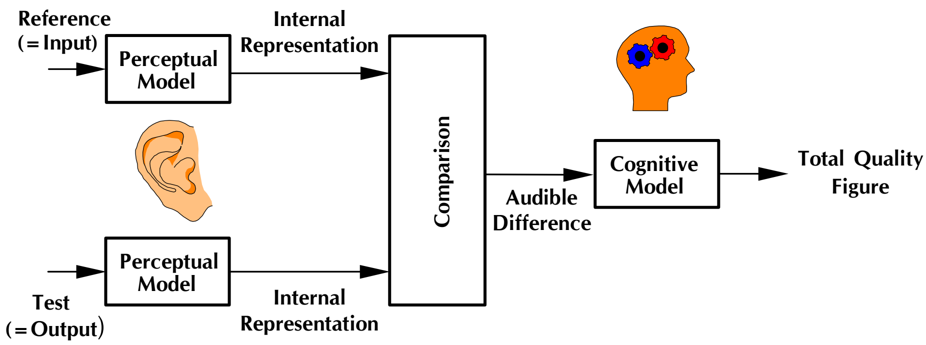


Figure 2.1: The underlying concept for perceptual measurement

The evaluation of the internal representation is often related to an estimate of the masked threshold. This estimate is based on data found in a number of psychoacoustic experiments, such as those conducted by Zwicker [ZWIC67, ZWIC82]. Most of these experiments model certain isolated effects of the human auditory system. One way to design a perceptual measurement algorithm is to generalize these model data and apply them to complex audio signals. This was for example the approach outlined in the NMR Algorithm in 1987 [BRAN87, BRAN89, BRAN92, GILC96, HERR92a, HERR92b, KEYH93, KEYH96, KEYH98, SEIT89]. Similar approaches were used for PAQM and PSQM [BEER95, BEER92, BEER94].

2.3 International Standardization

International standardization of perceptual audio measurement techniques was mainly driven by two expert groups within the International Telecommunications Union (ITU).

ITU-T P.861

Within the telecommunication sector of the ITU, in 1996 study group 12 finalized recommendation P.861 [ITUT861] for the objective analysis of speech codecs. After a wide-ranging comparison of proposed methods, the group opted for the PSQM algorithm. PSQM correlated up to 98 percent with the scores of subjective listening tests.

ITU-T P.862

Since P.861 was mainly developed for application to isolated speech codecs in mobile networks, a new measure was required to cope with real networks as well as packet based transmission. Driven by this demand for a verified test procedure for VoIP, an expert group within ITU-T SG12 has been working on an improved speech quality model. After a competitive phase, the new model "PESQ" has been devised. PESQ stands for "Perceptual Evaluation of Speech Quality". It combines a further refinement of PSQM and PAMS. Extensive tests showed PESQ's superior performance especially for VoIP applications. In February 2001, PESQ was accepted as the ITU-T Rec. P.862. More information on PESQ can be found at **www.pesq.org**.

ITU-R BS.1387

Within the study period 1994 – 1998, the ITU-R had established task group 10/4 with the scope to recommend an objective, perception based model to evaluate the quality of wide band audio codecs. After collecting a set of proposals, including the most popular ideas such as NMR, PAQM, PERCEVAL, POM and others, the group of model proponents opted for a joint collaboration to derive an improved model. In 1998, two versions of this new model were presented: A "Basic" version, featuring a low complexity approach, and an "Advanced" version for higher accuracy at the trade-off of higher complexity. After thorough verification, the model was recommended as a measure for the perceived audio quality ("PEAQ") under recommendation BS.1387 in late 1998.

Both standards, ITU-T P.861, and ITU-R BS.1387, currently represent the state-of-the-art technique for the objective evaluation of the perceived audio quality of audio codecs. Both techniques were derived from modelling the corresponding subjective experiment by an algorithm based approach. Thus it is essential to understand the scope of the modelled subjective experiment when trying to interpret the calculated results.

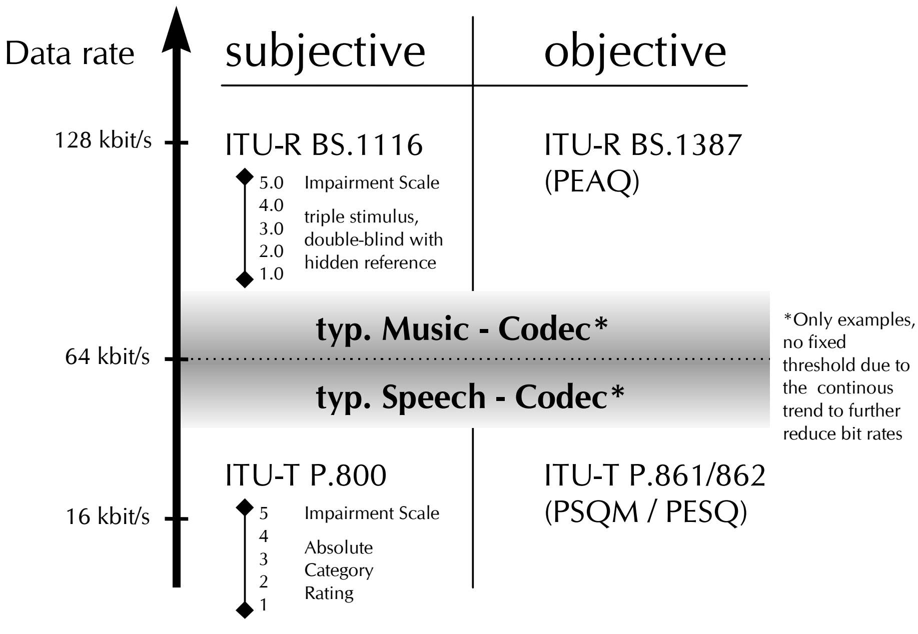


Figure 2.2: Overview on subjective and objective recommendations

Figure 2.2 summarizes the subjective test procedures and their corresponding objective counterpart in the context of typical data rate limits. As mentioned earlier, the threshold between both worlds - broadcasting and telecommunication - is floating due to the steadily attempt to further reduce the bit rates by more efficient coding schemes. Consequently, the overall data rate scale depicted in the figure should be taken as a course indicator only.

2.4 Which "Measurement" for Which "Application"?

A summary of the principle assessment scenarios and the corresponding measurement algorithms applications with OPERA™ follows. This section can be used as a reference when uncertain which measurement algorithm to apply.

First, remember the recommended perceptual measurement techniques always try to model the underlying subjective experiment of the corresponding listening test.

To decide which one is the proper experiment, remember to ask yourself the following questions:

- Is this an assessment of **wide band audio** signals (music, **or** bandwidth > 16kHz)?
- Would the subject be able to **compare** the test signal with the original **reference signal**?

If the answer to both questions is YES, you should apply the PEAQ algorithm.

If the answer to both questions is NO, you should apply PESQ, (or PSQM if required).

If none of the above seems to apply, an experimental situation outside the scope of both measurements should probably be considered. In this case, always consider how subjects would behave in a listening test. In some cases, however, you may also find that even subjects would not be able to properly score the sound quality.

In **Figure 2.3** the principle setup for a BS.1116 listening test is shown. **Figure 2.4** shows the corresponding situation for a P.800 compliant test setup.

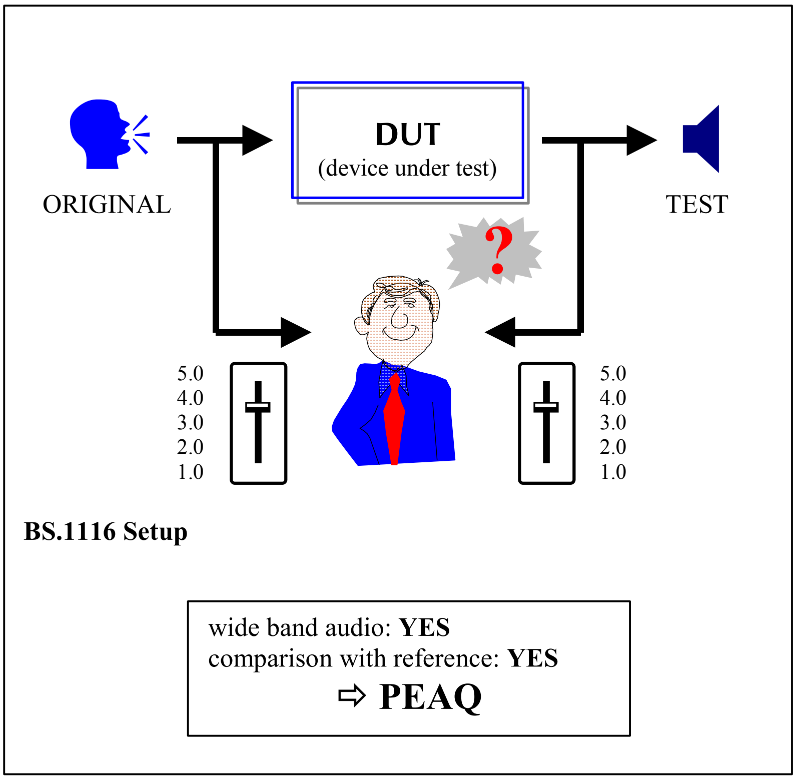


Figure 2.3: Illustration of the principle of BS.1116

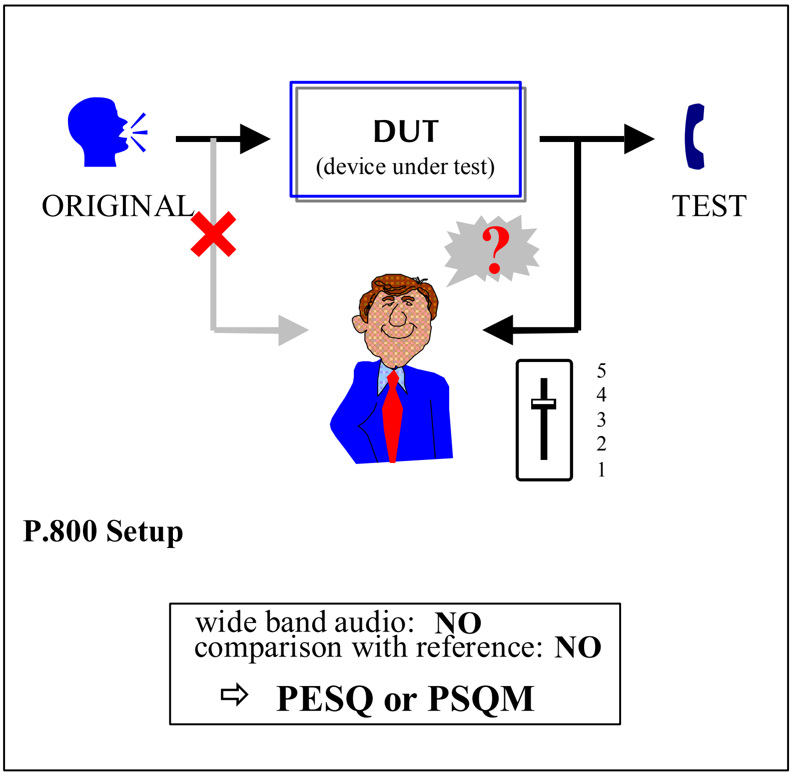


Figure 2.4: Illustration of the principle of P.800

2.5 Selection of the Reference File

As a rule of thumb, the reference file should be a signal that comes as close as possible to the kind of signal which shall be applied to the device under test in real life. E.g., if you design a special headset for female call center agents, you should use a test stimulus that contains mostly female speech. If the device should be used by male and female users as well as children, you should perform separate tests with typical stimuli for each of these cases. For the assessment of MPEG audio codecs that are used for the transmission of high quality music between broadcast studios, real music should be used. Especially with wide band music codecs a variety of at least six to ten different test samples should be selected, since the performance of audio codecs differs widely depending on the test material.

The duration of the test sequence should be within the range of approximately four to eight seconds. Longer tests will lead to averaging effects (short distortions may be averaged down by a long but almost perfect transmission) and shorter sequences may not be long enough to contain representative parts of the signal. If for any reason very long reference files are desired, OPERAs feature of measuring just a short sequence out of the entire input signals could be selected. Details regarding this feature under the Trigger option menu are explained in chapter 4.

The sample rate of the reference file is frequently already defined by the algorithm that shall be used for the evaluation of the recorded data. PEAQ according to ITU-R BS.1387 for example requires 48kHz sample rate, although the implementation in OPERA will deliver reliable results at 44.1kHz, too. Most speech quality measures are defined for 8 and 16kHz sample rate only. For more details, refer to the description of the individual algorithms or the standard documents that apply.

The selection of the sample format should mainly be driven by considering the capabilities of the underlying hardware. While using the audio interfaces provided by OPERA, it makes sense to select the 16bit linear. Since currently all measures use 16bit linear internally, any higher resolution, although supported by the hardware, will not result in more accurate measurements. When performing test calls with the voice board, the sample format should be 8bit mu-law or 8bit A-law (G.711). Otherwise the measurement will include at least one more step of encoding, since the DSP on the voice board will convert all input data back to G.711.

A set of typical wide band audio examples is mentioned in the ITU-R rec. BS.1387. Speech samples are also provided by the ITU-T, in the Series P Supplement 23.

2.6 How to Assess "Signal Enhancers"

Signal enhancers are pieces of equipment that try to make the processed signal sound better than the original signal, like e.g. noise reduction systems etc. When the input signal of the enhancer is taken as the reference and the output signal as the test signal of any perceptual measure, the result will usually be the opposite of what is expected. In general, the enhanced signal will be graded down the more, the better your enhancer works. This is because perception based measurement algorithms assume that any audible difference between the two input signals is a distortion, and by definition the "enhanced" signal will sound different than the unprocessed signal.

To get around this, a clean signal as the reference file (R) is recommended. This shall be distorted artificially, which results in signal D, the distorted reference signal. Signal D may now be used as the input to the enhancer. The output of the enhancer will be E, the enhanced signal.

When assessing speech quality at this time, the clean reference R and the enhanced signal E as the input signals of OPERA should be chosen. The grade calculated by the measurement algorithm now indicates how similar the enhanced signal sounds to the clean reference. This also implies that based on the measurement no statement can be made weather the enhanced signal sounds better than the original signal or not.. If the measurement result shall be compared to the result obtained from a listening test, it is important to remember that the question to the subjects must be "How much does the enhanced signal differ from the reference signal?" and not "Does the enhanced signal sound better or worse than the reference signal?". There is no standardized measure available today which answers the second question, which is frequently a matter of personal taste.

Going one step further, calculate the gain achieved by the enhancer when relating the final MOS derived this way, to the MOS achieved by comparing the clean reference (R) to the distorted reference (D). Figure 2.5 shows a sketch of such a setup.

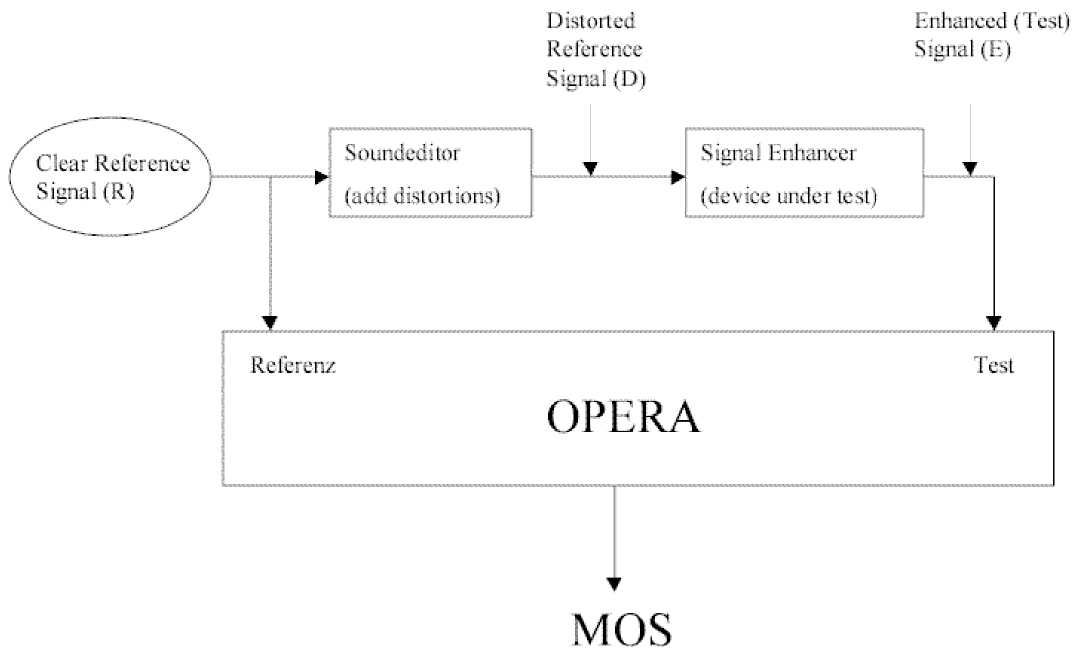


Figure 2.5: Setup for measuring signal enhancers.

Chapter 3

3 INSTALLATION AND SETUP

Using OPERA™ for the first time.

Please review this chapter before you continue. Checking the **complete contents** of your OPERA™ system may help in the future to solve potential problems.

This chapter includes the installation and setup information for the whole OPERA™ product family. Please refer to the section of the product you have purchased.

3.1 OPERA™ Software Suite

3.1.1 Unpacking the Software

After unpacking the Software Suite, please check the delivery for completeness first. The delivery should include the following parts:

- One OPERA™ Software Suite CD.
- One CD with reference wave files and the conformance test set (PEAQ only).
- One dongle (hardware key, a small grey box which is to be attached to the printer port).
- This manual.

3.1.2 System Requirements

Please verify that your computer meets or exceeds either of the following system requirements:

- >=128MB of RAM, 256MB are recommended
- Screen resolution 1024*768, >=64k colors.

- NVIDIA TNT2 compatible graphics adapter (others may work, but are not yet tested).

- Windows NT 4.0 SP4 or SP5, Windows 2000 or Windows XP.

3.1.3 Installation and Setup

To install the OPERA™ Software Suite, follow the steps below, where "X:\" represents the CD-ROM drive containing the OPERA™ setup CD. If you want to update from a previous version, there is no need to uninstall the previous version. The setup program will automatically do this for you

1. Attach the dongle to the parallel interface port of your computer.

2. Start the OPERA™ setup program (X:\ OperaSetupVxxx.exe).

3. Install the required options if any are offered.

4. Reboot the computer.

3.1.4 Verification

If your installation is running too slow, please check if the system is running out of memory. If this is the case, memory will be swapped out to the harddisk and accessing this part of the memory is a million times slower than accessing real memory. If the OPERA Software Suite requires too much memory, adjust the size of OPERA's history buffer through a parameter in the registry. Please ask a specialist for assistance, if unfamiliar with the registry and use of regedit. Modifying the registry may seriously harm your Windows installation and even prevent it from booting.

In order to adjust this parameter, open regedit and look for the key:

HKEY_LOCAL_MACHINE\Software\Opticom\Opera\Memory\MemoryReserved

A good value for this key is 130 000 000 on a machine with 256MB RAM. The parameter defines how much of the physical memory is left free by OPERA.

3.2 OPERA™ Measurement System

3.2.1 Workstation version

Unpacking the System
After unpacking the system check the delivery for completeness first. The delivery should include the following parts:

- PC workstation.

- Monitor (not included in the case of international shipments).

- Keyboard.
- Mouse.
- Two power cords (one power cord included in the case of international shipments).
- One or two audio multi cord connectors with cables (Systems equipped with audio boards only).
- Two phone splitter cables with three RJ-14 connectors each (OPR-101-xxx-W telecommunication version only).
- One dongle / hardware key (may be attached to the printer port already).
- One PEAQ Sample CD (systems with installed PEAQ algorithm only).
- This manual.
- Windows NT CD-ROM and manual.
- Additional documentation and/or software may be included, depending on the version of the system.

Note:

All equipment is designed to work on 220V as well as on 110V AC. Please check for any transportation damages.

Installation and Setup

The OPERA™ system comes completely pre-configured, so no additional setup procedure is required to run the system. Please refer to the Windows NT documentation and consult your local network administrator, when attachment of the system to a network or shared network printer is desired.

Note:

Please do not upgrade the operating system or the service pack without consulting OPTICOM.

To setup the OPERA™ workstation system, place the main unit on a solid and dry surface. Connect the monitor, mouse, keyboard and the power cables to the back of the main unit. Attach the dongle to the parallel interface port of the computer if it is not yet there. Switch on the monitor and the system. For details, please refer to the hardware documentation manual.

IP / LAN Concerns

All OPERA™ systems come with an ethernet port that may be used to connect the system to a data network. As default, the OPERA™ system uses a fixed IP Address. Please consult your network administrator before attaching OPERA to a LAN.

> **Note:**
>
> Do never delete the TCP/IP protocol! It is required for proper operation of the system.

For the setup of the audio interface board or the telephony board, please refer to the corresponding section in Chapter 5 or Chapter 6, respectively.

3.2.2 Portable PC Version

Unpacking the System

After unpacking the system, check the delivery for completeness first . The delivery should include the following parts:

- Portable industry type of computer system with integrated 15" TFT display.
- Keyboard.
- Power cord.
- One or two audio multi cord connectors with cables (Systems equipped with audio boards only).
- Two phone splitter cables with three RJ-14 connectors each (OPR-101-xxx-W telecommunication version only).
- One dongle (may be attached to the printer port already).
- One PEAQ Sample CD (systems with installed PEAQ algorithm only).
- One System Recovery CD (for shipments after after fall 2000)
- This manual.
- Windows NT CD-ROM and manual.
- Additional documentation and/or software may be included, depending on the version of the system.

Note:

All equipment is designed to work on 220V as well as on 110V AC.

Installation and Setup

The OPERA system comes completely pre-configured, so no additional setup procedure is required to run the system. Please refer to the Windows NT documentation and consult your local network administrator, if attachment of the system to a network or shared net work printer is desired.

Note:

Please do not upgrade the operating system or the service pack without consulting OPTICOM.

To setup the OPERA™ portable system, place the main unit on a solid and dry surface. Attach the dongle to the parallel interface port of the computer, if it is not yet there, connect the power cable to the side panel of the main unit and switch on the system. For details, please refer to the hardware documentation manual.

IP Concerns

As default, the OPERA™ system gets his IP address from a DHCP server. If the OPERA™ system is used without a DHCP server or without a connection to a network, a fixed IP address must be entered. For instance, enter the IP address "192.168.0.1". The corresponding netmask in this case is 255.255.0.0. In case of any doubt, please consult your network administrator.

Note:

Please do never delete the TCP/IP protocol! It is required for proper operation of the system

For the setup of the audio interface board or the telephony board, please refer to the corresponding section in Chapter 5 or Chapter 6, respectively.

Chapter 4

4 GETTING TO KNOW THE OPERA™ FRAMEWORK

Description of the Underlying Concept of OPERA™ and the Framework Menu Options and Command Line Parameters.

OPERA™ is based on a modern open architecture which offers a high degree of flexibility as well as room for future technology advances. The system can be visualized consisting of general modules, which will be needed for various kinds of measurements, called the **framework**, as well as some specific modules that implement a detailed functionality, called **plug-ins**.

In this chapter the concept behind OPERA™ will be explained. We will start by defining the data acquisition part of OPERA™. Following this section is a description of the OPERA™ framework as it is used for online measurements and the evaluation of previously acquired data. The performance of measurements from batch files will be explained. Remember, this chapter describes the general operation of the system, i.e. commands that all available algorithm plug-ins have in common. The descriptions of the plug-in specific topics are found in Chapter 5 or Chapter 6, respectively. Users of the Software Suite may skip the explanation of the data acquisition process, since this feature is not available in their version of OPERA™.

4.1 General Concept

Since OPERA™ supports a wide variety of interfaces, it was decided to move the data acquisition part of the measurement into a separate program, called OptiCall™. The evaluation of the recorded data is handled by the OPERA™ Framework. This split enables both parts of the software to focus on their special task and both parts are easier to use than one monolithic, functionality overloaded program. This way it is also possible to first acquire the data by e.g. establishing a test call on a telephone network, and then to evaluate the data off-line using various algorithms,without having to repeat the data acquisition. On VoIP networks, especially it is most valuable PSQM, PESQ and Echo parameters can be evaluated for exactly the same situation on the network (without having to repeat the call for the next algorithm). The standard set-up creates a connection from originator to terminator. Once this connection is established, the test file is sent from the terminating side back to the originating side for analysis. In the expert view (see 4.2.3 The OptiCall™ Program) the set-up where the test file is sent in the reverse direction can be selected.

4.2 Data Acquisition Using OptiCall™

This chapter is divided into a section describing the hardware interfaces and a section that describes the OptiCall™ program which is used for the data acquisition in the telecom version, as well as in the broadcast version.

Note:

OptiCall™ was originally designed for use with telephony interfaces only. Nevertheless it can be used for measurements with the audio interfaces as well. This is especially useful if terminal-type-equipment needs to be assessed including the acoustical path, or for testing wide band audio codecs.

4.2.1 POTS Telephony Interfaces

Analog Loop Start Interfaces

The POTS telephony interfaces of the OPERA™ system (OPR-101-xxx-x) are accessible from the rear of the system or - in the case of the portable version – from the side panel. As shown in **Figure 4.1** the board connects to analog loop start lines using two RJ-14 connectors. These are labelled as "**Lines 0/1**" and "**Line 2/3**" on the chassis of your OPERA™ system. The electrical specification of these interfaces is country dependent and conforms to the standards for analog loop start interfaces.

No special set-up steps are required here. The appropriate line interfaces of the OPERA™ system connect to the telephone network to be assessed.

A 3.5 mm stereo jack for an audio output is located on the POTS telephony board's end bracket. Any Sound Blaster compatible device with line input (e.g. active loud speakers) is compatible with this interface. On the OPERA™ portable testers, this output is used for monitoring the telephone lines.

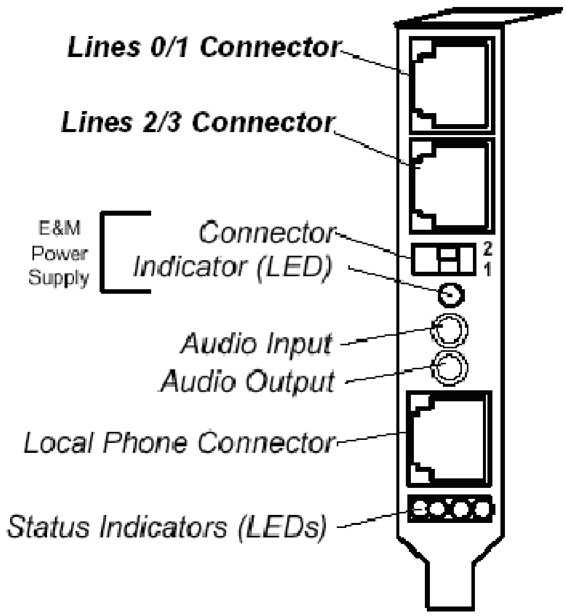


Figure 4.1: POTS Telephony board end bracket

> **Note:**
>
> On the end bracket of the POTS telephony interface board, the following connectors are currently not used: E&M power supply, Audio Input and Local Phone Connector.

> **Note:**
>
> Be careful not to mix up the POTS telephony boards audio input and output with the optional high quality sound board that is used for data acquisition..

Since both ports of OPERA provide two POTS interfaces each, the splitter cables shown in **Figure 4.2** are used to split up the signals and connect two RJ11 cables to the POTS telephony interface board as illustrated in **Figure 4.3** below. There are three labels on the adapter: "0(2)", "1(3)" and "PC". The "PC" connector side must be connected to the telephony board. Depending on the port of the telephony board, the adapter connectors labelled with "0(2)" or "1(3)" represent either line 0 and line 1, or line 2 and line 3.

Figure 4.2: Splitter cable used to attach two POTS lines to one OPERA port.

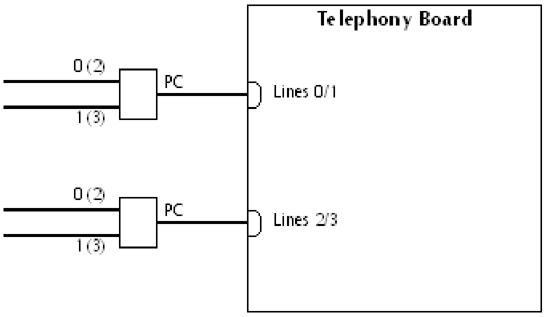


Figure 4.3: The configuration and labelling of the adapter wires

Monitoring Telephony Signals

The POTS telephony boards audio output may be used for monitoring purposes on the portable OPERA™ systems. To accomplish this, connect the POTS telephony board audio output to the on-board audio input of the portable machine, and the on-board line out signal to the speaker input, which is located next to the power supply fans on the left side of the machine. All connections here are using 3.5mm jacks and should be preconfigured. Again, be careful not to mix up the on-board sound and the audio output of the POTS telephony board with the optional high quality sound board. Next, the volume should be adjusted. This needs to occur in two locations. First, the hardware control on the right side of the portable OPERA™ system, and second, by using the mixer control of windows. If the set-up is correct, double-click on the "Monitor Line X" icon on the desktop, after pressing start in OptiCall, what is happening on the telephone line X can then be heard.

Country Specific Settings

Although the signalling on analog telephone networks differs significantly from country to country and between individual PBXs, OPERA will usually successfully establish test calls under most circumstances. Nevertheless it is possible to adapt OPERA systems exactly to most networks. Preinstalled on all OPERA systems is a variety of country/PBX specific settings. To change the settings, open an Explorer window and go to:

c:\Programme\QXCustomParameters1999-4

In this folder a number of batch files is stored which are all named Setxxxx.bat, where xxxx stands for a specific country or switch. To choose one of them, close OptiCall as well as all other TAPI applications, and double click on the Setxxxx.bat file. The system will ask several times to press any key to continue, which should be done. After the batch is finished, go to **Start | Programs | Natural Microsystems QX | Hardware Initialisation**. An other command line program will run. After this is finished, the new settings are available.

4.2.2 Audio Interfaces

The Broadcast Version of OPERA™ includes audio interfaces (OPR-110-EAQ-x) that enable real-time-measurement performance, necessary in the case of on-line network monitoring for example. This system version can handle digital and analog audio interfaces with the audio interface board integrated in the system.

If your OPERA™ system has been manufactured before June 2000, a Digigram audio board is installed. If the manufacturing date of the system is after June 2000, it may be equipped with LynxONE audio boards. For information concerning the audio interfaces, please refer to the section below, that deals with the board(s) installed in your OPERA™ system.

Digigram Audio Interface - Setup of the Cable Connections

OPERA™ systems equipped with a Digigram audio board come with a multi cable connector whose layout and labelling is shown in **Figure 4.4**. The multi connector consists of one 62-terminal D-Sub connector on the one end of the cables as well as 14 XLR connectors (7 female, 6 male) and one BNC connector on the other end.

Connect the D-Sub plug to the appropriate interface on the audio board of your system. The XLR connectors labelled "In1", ..., "In4" and "Out1", ..., "Out4" are the input and output plugs for analog signals. the connectors labelled "In1" and "In2" are assigned to the first stereo channel of the board, "In3" and "In4" are assigned to the second stereo channel. Correspondingly, the output connectors are assigned to the stereo channels.

You can use the BNC connector labelled "Word Clock In" to synchronize your OPERA™ system with an external clock or the AES-EBU Sync connection.

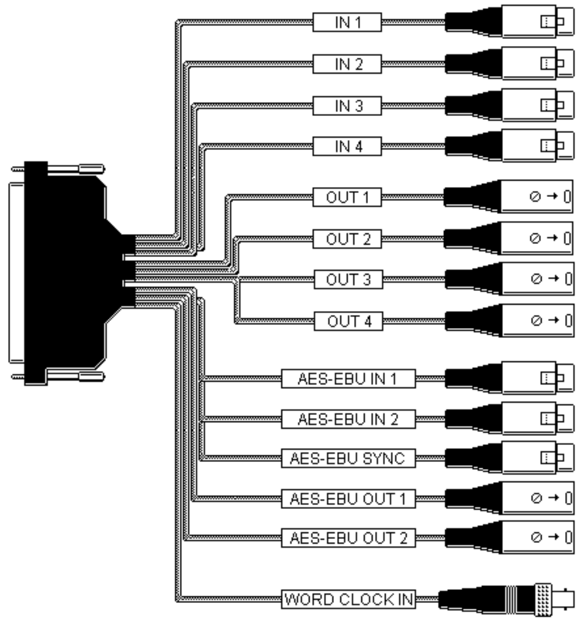


Figure 4.4: Layout and labelling of the multi cable connector.

In addition to the connectors for analog signals there are five XLR connectors for digital signals, two inputs ("AES-EBU In1" and "AES-EBU In2"), two outputs ("AES-EBU Out1" and "AES-EBU Out2") as well as one connector for synchronizing ("AES-EBU Sync").

Digigram Audio Interface - Setup of the Audio Board Mixer

Before you perform measurements with the Digigram audio interface board, settings need to be made in the mixer program of the audio board. The mixer program can be opened from the Windows NT Start menu **Start|Digigram|DigiMix**. The mixer dialog is shown in **Figure 4.5**. Here, select whether the digital AES-EBU input connections or the analog inputs shall be used, then click on the corresponding buttons in the "Record Source" field. In the field below – titled "Clock Select" - the settings for the clock can be chosen **Figure 4.6**, There are three kinds of synchronizing your digital signals. Either the Word Clock Input, the AES Sync Input or the Digital Input 1 can be chosen.

It is not necessary to change the settings of the input gain settings since OPERA™ usually works fine with the default settings.

Note:

The Digigram Audio Interface Board is designed as a dual stereo board. Consequently, two mixer displays can be called, each taking effect of one stereo channel. To switch between the two mixer displays, select either one from the menu option "Mixer".

Note:

The faders of the audio mixer take effect on the gain of the digital side of the A/D converter. This means that the S/N ratio is decreased when the analog output gain is decreased.

Note:

The latency of the Digigram driver is approximately 100ms.

Both digital inputs of the Digigram board can be used, by making sure that both inputs are running at exactly the same, synchronized sampling frequency. If feeding back the digital outputs to the digital inputs is desired, supply an external clock to one of the synch inputs of the board. To use OptiCall for the data acquisition, as described later in this chapter, connect AES/EBU Out 1 to AES/EBU In 2, select Audio #2 as the originating and Audio #1 as the terminating interface. An external clock signal (e.g. another AES/EBU signal at the same sample rate as required for the played file) must be connected to AES/EBU In 1 in this case. Alternatively, feed back AES/EBU Out 2 to AES/EBU In 1 and connect any of the external synch inputs a valid clock signal.

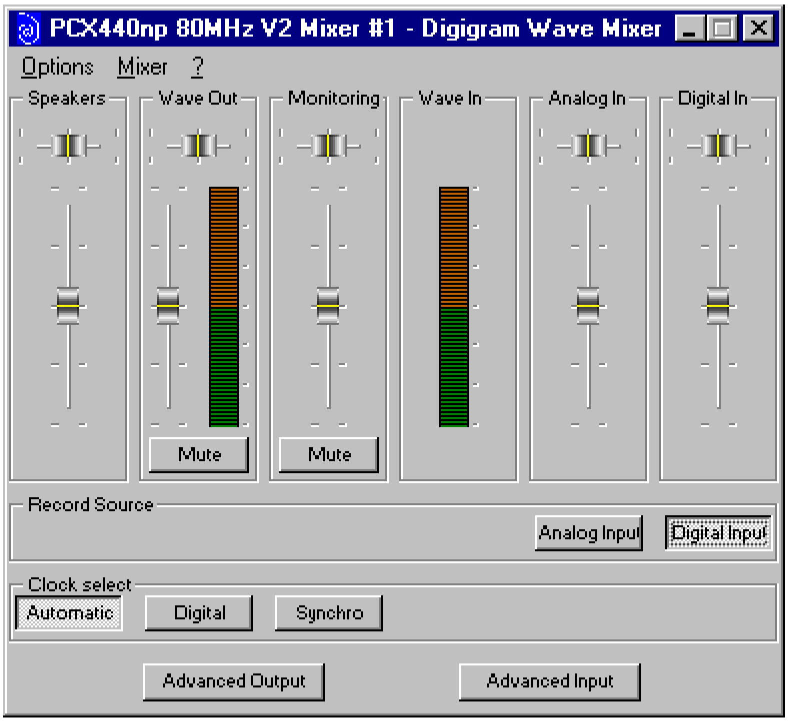


Figure 4.5: Mixer Window of the installed Audio Interface Board

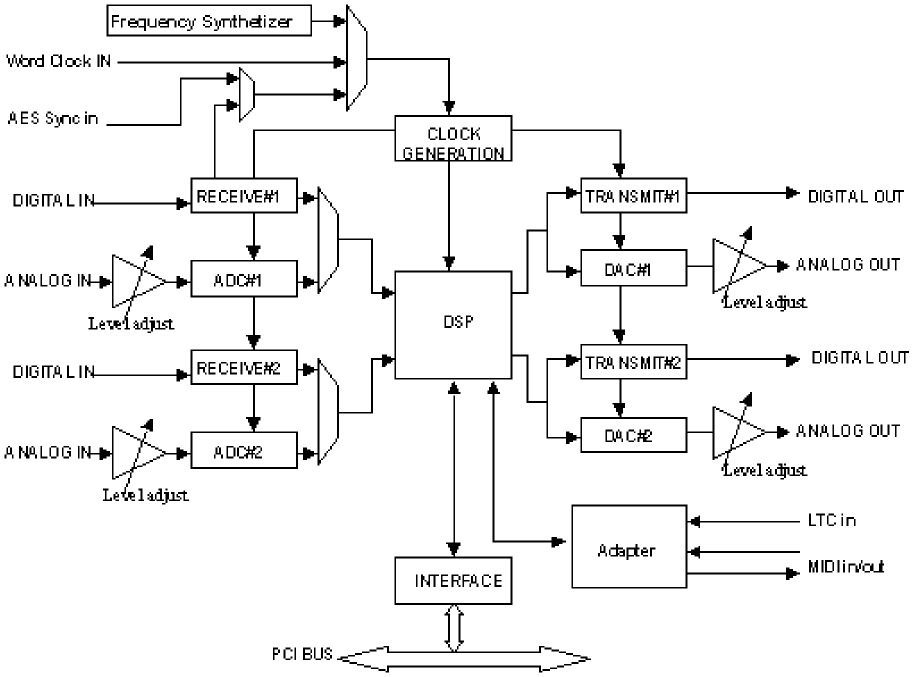


Figure 4.6: Block diagram of the Digigram audio board.

LynxONE Audio Interface - Setup of the Cable Connections

OPERA™ systems equipped with two LynxONE audio boards include two multi cable connectors for each board. The six-foot Audio Cable provides XLR connectors for analog and digital audio signals. The two-foot MIDI/Clock Cable provides 5-pin DIN connectors for the two MIDI ports (not used with OPERA™) and female BNC connectors for clock input and output.

Two bracket mounted D-connectors provide connection ports for the two multi cables: the Audio Port is a 25-pin female D-connector and the MIDI/Clock port is a high-density 15-pin female D-connector (see **Figure 4.7**).

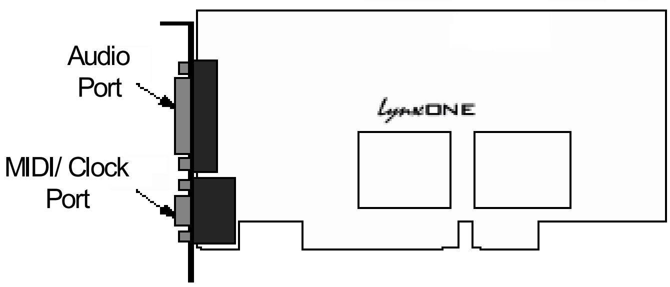


Figure 4.7: Mounting bracket of the LynxOne audio board [LYNX00]

The use of both cables is not a requirement. When not using the clock capabilities of the LynxONE, for example, there is no reason to connect the MIDI/Clock cable.

The XLR connectors on the Audio Cable, labelled LEFT IN, LEFT OUT, RIGHT IN, and RIGHT OUT, are used to connect balanced and unbalanced analog equipment to the two analog inputs and outputs of the LynxONE. Make connections to professional consoles and other balanced equipment directly with these connectors.

The nominal analog signal levels are compatible with either professional or consumer equipment. Use the Trim control on the LynxONE Mixer application to select +4dBu for balanced professional devices or –10dBV for balanced or unbalanced consumer devices. The Trim control affects both inputs and outputs. On request special adapter kits are available as an option for interfacing to consumer devices like e.g PC soundcards (3.5mm phone jack or cinch), headset (2.5mm phone jack) or handset (RJ22) connectors.

Note:

Please note that with 16 dB of headroom, the LynxONE's analog outputs are capable of delivering +20 dBu signal levels. It is important to verify that the equipment is capable of handling these signal levels in order to prevent clipping or possible damage.

The XLR connectors on the Audio Cable, labelled DIGITAL IN and DIGITAL OUT are used for AES/EBU and S/PDIF digital audio connections. Connect AES/EBU devices directly and select **AES/EBU** Digital Format in the LynxONE mixer application. Connect to S/PDIF devices using XLR-to-RCA adapters. Select **S/P DIF** Digital Format in the LynxONE mixer application.

The BNC connectors on the MIDI/Clock Cable, labelled CLOCK IN and CLOCK OUT are used to synchronize the LynxONE with external equipment. The connectors support TTL level signals and should be connected with 75-ohm coaxial cable. Connect the CLOCK IN connector to the clock output of an external device and select **External** as the Sample Clock Source in the LynxONE mixer. Adjust the clock reference to match the incoming clock type. CLOCK OUT is a word clock that tracks the sample rate of the LynxONE. Connect this output to the word clock input of an external device. [LYNX00]

The MIDI connectors are not used with OPERA™.

LynxONE Audio Interface - Setup of the Audio Board Mixer

Before performing measurements with the LynxONE audio interface board, make some settings in the mixer program of the audio board. If there is no symbol for the LynxONE mixer available in your Windows NT systray, first start the application by clicking on: **Start|Programs|Lynx One|Lynx One Mixer**. Otherwise double click on the systray icon.

The audio board labeled "**Audio 1 Master (Ref)**" is configured as the master board. By convention use this as the input of the reference signal. The audio board labeled "**Audio 2 Slave (Test)**" is the slave board. Connect the test signal to board number two. When working with digital (AES-EBU) signals, the two signals must be synchronized to the same word clock before they are fed as input to the OPERA™ system.

The LynxONE mixer application dialog displays the settings for one of the two boards. To switch between the boards, use the menu **Mixer**.

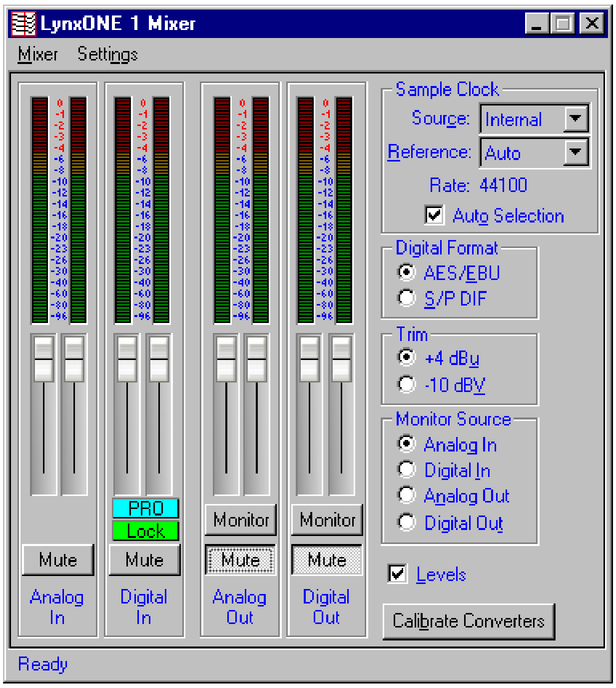


Figure 4.8: LynxONE mixer setting for an analog reference signal

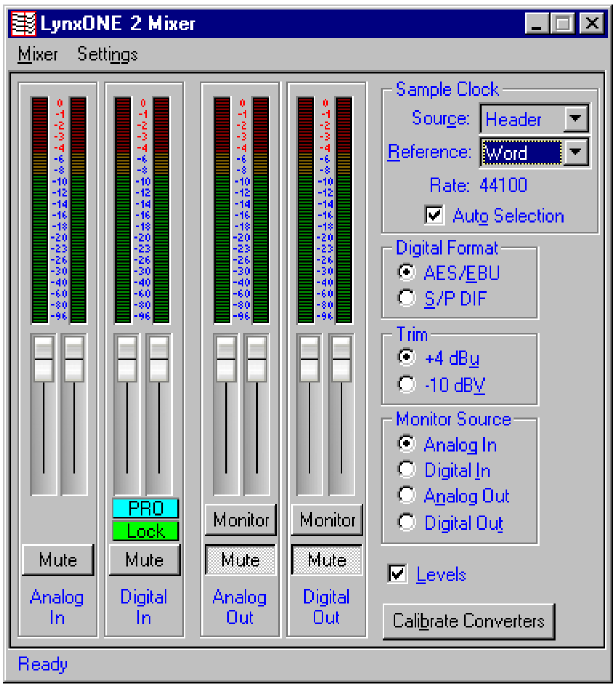


Figure 4.9: LynxONE mixer setting for an analog test signal

Figure 4.8 and **Figure 4.9** represent the typical settings for a measurement with **analog** input signals. As shown in the example in **Figure 4.8**, the Source for the Sample Clock is **Internal**, the Reference is set to **Auto** and Auto Selection is turned on. **Figure 4.9** shows the mixer settings for the test signal. The Source for the Sample Clock is **Header**, the Reference is set to **Word** and Auto Selection is turned on.

Figure 4.10 and **Figure 4.11** represent the typical settings for a measurement with **digital (AES-EBU)** input signals. As shown the mixer settings for the reference signal in **Figure 4.10** the Source for the Sample Clock is set to **Digital**, the Reference is set to **Auto** and Auto Selection is turned on. **Figure 4.11** shows the mixer settings for the test signal. The Source for the Sample Clock is **Digital**, the Reference is set to **Auto** and Auto Selection is turned on.

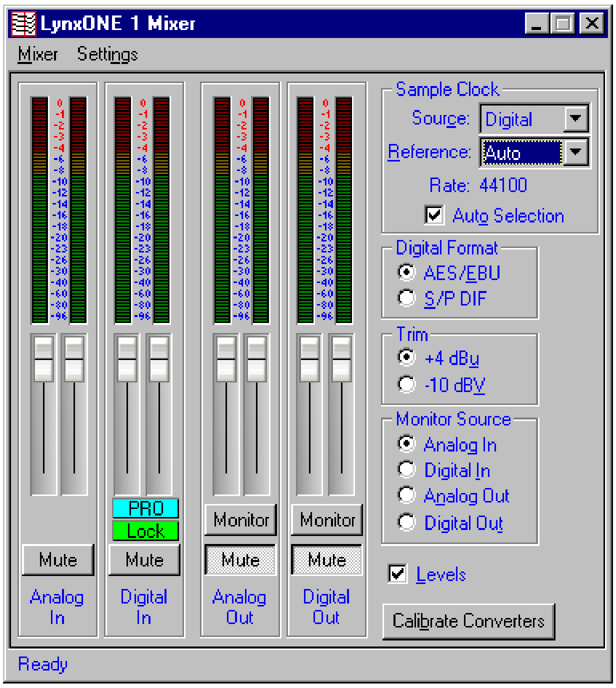


Figure 4.10: LynxONE mixer setting for a digital (AES-EBU) reference signal

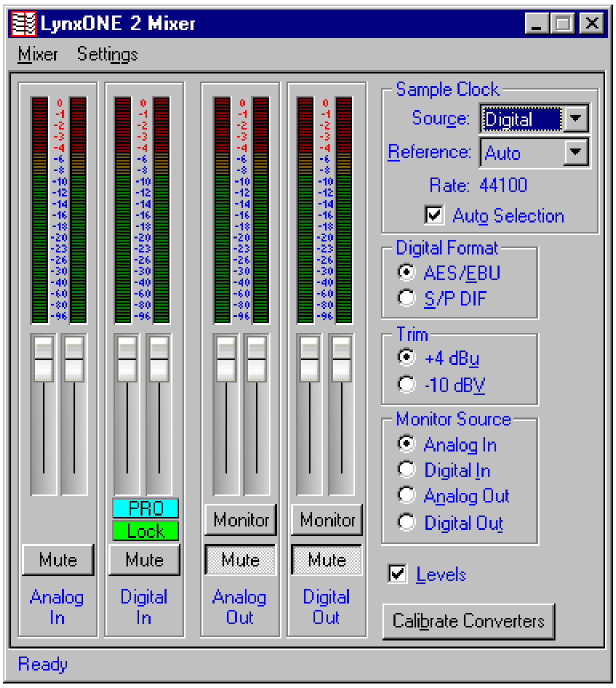


Figure 4.11: LynxONE mixer setting for a digital (AES-EBU) test signal

LynxONE Audio Interface – Recording Format

To achieve the highest possible accuracy, all recordings from the LynxONE interfaces will be made in stereo and at 24bit resolution. After applying any eventually required gain factors, the files will be stored as 16bit stereo. When using the interfaces for mono measurements on e.g. VoIP terminals, simply use the OPERA crossbar switch matrix as described in chapter 4.4.3 to select the channel on which your device under test was connected.

E1/T1 Interfaces

E1/T1 interfaces are optionally available for OPERA™ as well. The standard version includes two trunks (four port versions are available on request too) which may be software configured as E1 or T1. Various protocols and protocol versions are supported, including CAS, MFCR2 and ISDN PRI. Each individual time slot will show up in OptiCall™ as an individual interface. All timeslots may be used simultaneously. The detailed configuration and usage of the E1/T1 interface option is described in a separate manual.

4.2.3 The OptiCall™ Program

Data acquisition for offline measurements is performed using the external program **OptiCall.exe**. Online measurements are handled by the OPERA™ framework directly.

Basic Principle

The principle of OptiCall™ is easy. First a connection is established between any two selected interfaces and then the speech/audio data are transferred through the selected interfaces. Simultaneously OptiCall™ records all the incoming data on both sides of a full duplex connection. OptiCall™ is not restricted to telephony interfaces only, it can make use of audio interfaces as well. If for either the originating, or the terminating side (or both sides) an audio interface is selected, the call control is simply skipped for this particular interface, and the audio interface is in the connected state immediately. Using the audio interfaces, OptiCall™ can also be used for the assessment of wide band audio codecs (like e.g. MPEG or AAC codecs), multimedia terminals or any other terminal type equipment.

Making Calls with Remote Machines

One of the key features of OptiCall™ is running the user interface on the local machine, but the call may be executed on a remote machine. This is accomplished by simply telling OptiCall™ the name of the remote machine.

Different User Interfaces for Different Applications

Since OptiCall™ supports various options that are frequently application specific, it comes with three different graphical user interfaces. The **Audio Standard View** is mainly designed for pure audio applications that do not require complex settings for phone numbers etc. The **Telephony Standard View** is the GUI most frequently used in telecom applications for issuing test calls. It provides all required parameters for setting up simple loop calls, for originating a call or for terminating a call. If more features like bulk call generation are required, the **Expert View** is the best choice. In this view the full functionality of OptiCall™ is available. Switch easily between the views by clicking with the right mouse button into the blue title bar of OptiCall™. The following sections will explain all views in detail.

Common Settings

In all modes OptiCall™ can be configured to run on either the local machine or any other OPERA system that is connected through a TCP/IP connection. The field **Network Node** which is available in all views, must always be set to the network name of the machine which is hosting the interface which you want to use for the call. Alternatively the IP address of the host may be entered. To use two interfaces which are located on different machines, simply open two instances of OptiCall™. OptiCall™ will try to establish a connection to the remote system whenever the name in the edit field changes and click to somewhere outside of that field. Please note that searching for a remote system may take some time (between few seconds and approximately one minute). Enter the name of the local system or "*localhost*" as the network if the calls shall be issued on the local system. After having established a connection to an OPERA™ system, the drop down boxes for the interface selection will be updated. Please note that delay measurements are not possible for measurements between two systems when using OptiCall™. If delay measurements between two OPERA systems are required, the OPERA Control Center must be used.

Note:

A permanent TCP/IP connection between two systems is required to set up a call between two OPERA Systems.

The edit field **Reference** must contain the full path and the name of the file containing the speech sample. Instead of entering the path manually, click on **Browse** and search the file by using the explorer. If a valid file is selected, some information on the encoding of the data below of the edit field will be observed. In case there is no information displayed, the file does either not exist, or is of an invalid format. Depending on the view and operating mode there may be one or two reference files required, one for each side. In the Telephony Standard and Audio Standard Views the system will automatically choose the same reference file for both sides. In the Expert View choose different files, which is useful for measurements under double talk situations. When choosing two files however, take care that they are more or less equally long, since the measurement will stop after the shorter file is finished.

Note:

Currently supported file formats are *.wav files containing 8 kHz sample rate, 8bit G.711 (a-law / µ-law) for telephony interfaces and all standard sample rates, plain PCM or G.711 for audio interfaces.

The next edit field **Dest. Directory** indicates to where the recorded files shall be written. The entry starts with a path to a directory that must exist on the local machine, e.g.

c:\temp\test

in **Figure 4.12** After the path, a "#" symbol is used to separate the path from the root file name that will be used to create the final file name, e.g.

Demo

in **Figure 4.12**. The final filenames that are recorded to the harddisk will start with this root file name and a unique identifier for the line interface on which the data were recorded, plus the extension ".wav" will be appended to this name.

To start the data acquisition press the **Start** button. The blue progress bar at the bottom of the window will start cycling from one end to the other until the acquisition is finished. A status field above the progress indicator indicates the current state of the process. Abort the acquisition at any time by clicking on the **Stop** button. In case a connection could not be established, or one of the lines was busy or not obtainable for any reason this will be displayed in the status window for approximately two seconds after the end of the acquisition. When selecting one of the POTS interfaces, **monitor** the signal using the internal speakers of the portable OPERA™ systems. Simply double click on the icon "Monitor Line x" (x stands for the line interface that you want to monitor) after pressing the start button. The monitoring icons can be found on the desktop. If nothing can be heard check the volume controls of the portable PC, and those of Windows (microphone in and line out of the on-board sound).

Audio Standard View

The Audio View as shown in **Figure 4.12** is used for all applications that require neither telephone lines, nor special settings for the record gain, bulk call, and trending analysis or other parameters available in the Expert view only. The full functionality of the Audio Standard View is available in the Expert view as well – perhaps not as easy to use as here. In the Audio Standard View all unnecessary parameters are left out, e.g. phone numbers and the names of some of the field in order to be more meaningful for pure audio applications are modified. Essentially the Audio Standard View is targeting two different set-ups. Depending on your application choose the set-up according to **Figure 4.13** or **Figure 4.14**.

The set-up as in **Figure 4.13** is using one Audio Interface Option. The output is connected to the input of the device under test and the input is connected to the output of the device under test. For this set-up, click on the radio button "One Audio IF" in OptiCall (IF stands for interface). The single audio interface will then full duplex play and record at the same time.

For the test setup according to **Figure 4.14** two Audio Interface Options are required. The first one is connected to the input of the codec and is playing only, while the second is connected to the output of the device under test and is recording only. For this type of set-up, click on the radio button "Two Audio IF" in OptiCall (IF stands for interface)

The only disadvantage of the single interface solution is that it requires the device under test to permanently generate a digital clock on the output if the AES/EBU inputs of OPERA shall be used. Also the output of the device under test must be fully synchronous to the input. For analog set-ups, however, the first solution may be the easier one, since no synchronisation between the A/D and D/A converters is required (each board taken for itself is always synchronous). For applications using the digital interfaces however, the second mode with two interfaces is much more convenient, since it avoids most problems with asynchronous or non-existing digital clocks. Just be sure, that in the Lynx Mixers the proper clock sources have been selected. For the playing interface, the clock source must be switched to internal and for the recording interface it must be set to digital. Also take care, that on the playing side the digital interfaces are neither selected as the monitoring source, nor as the monitoring output.

It may seem unusual, but from the drop down list box for the interface selection you can also choose telephony interfaces. Do not select those when in the audio view. Nothing serious will happen, only the call will never connect due to the missing telephone number etc. (There must be something left for the next release...)

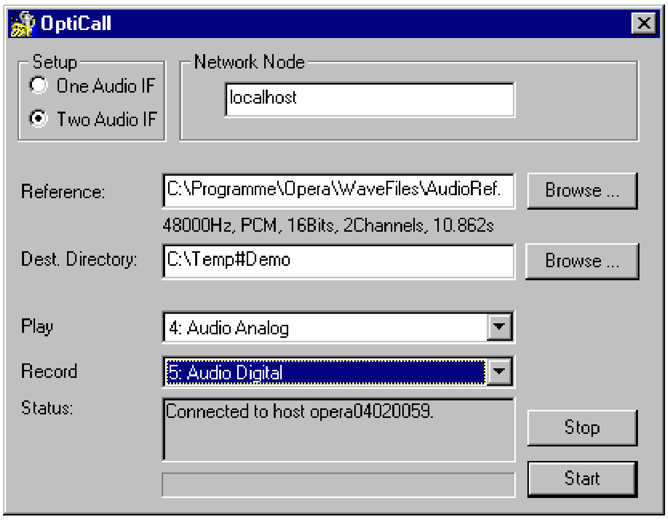


Figure 4.12: OptiCall™ - Audio Standard View.

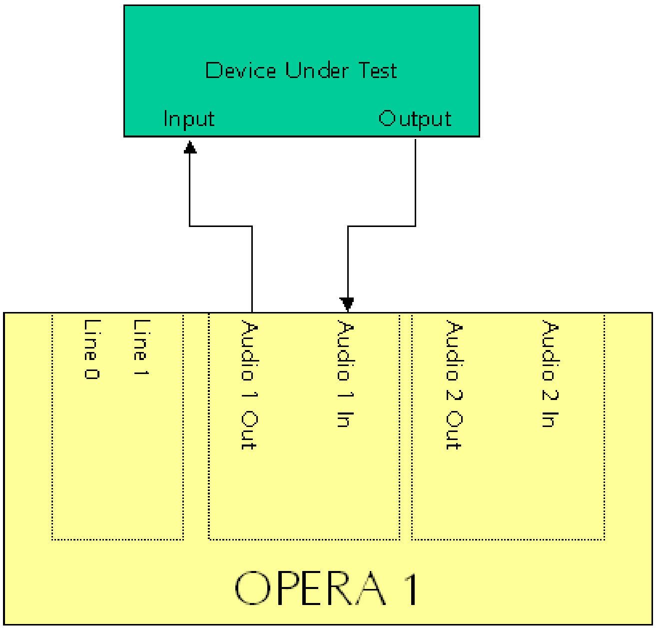


Figure 4.13: Typical audio setup using only one interface.

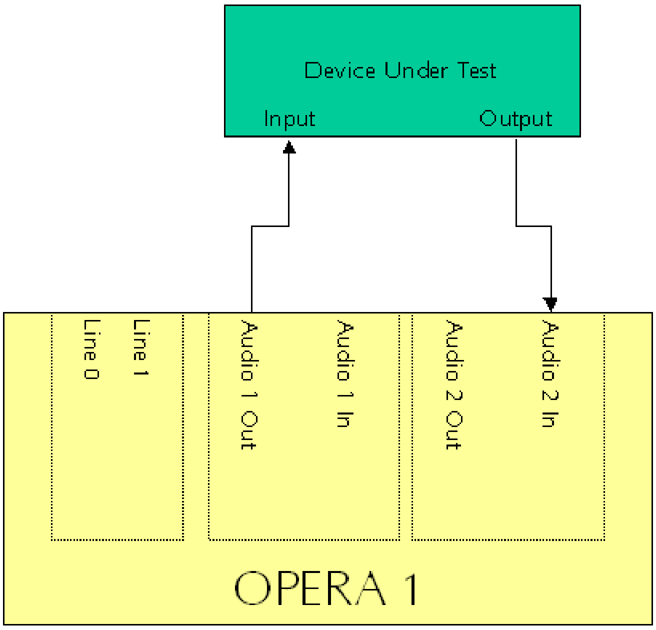


Figure 4.14: Typical audio setup using two audio interfaces.

Telephony Standard View

OptiCall™ may be used in one of the following four operating modes:

- Loop
- Origin
- Termination
- Terminate All

In the loop mode, one instance of OptiCall™ controls both, the originating as well as the terminating side of a connection. Both sides must be connected to the same OPERA™ system.

If set to the Origin mode, OPERA™ is connected to just the originating side of a connection. OptiCall™ will initiate a phone call on the specified interface and start transmission as well as recording of the audio data as soon as the call is answered by the terminating side (which may be another OPERA™ system, or just another instance of OptiCall™ running on the same machine).

In the Termination mode, OPERA is connected to just the terminating side of a connection. OptiCall™ will wait for an incoming call on the specified interface and start transmission as well as recording of the audio data immediately. The origin of the call may be another OPERA™ system, or again just another instance of OptiCall™ running on the same machine.

The terminate-all-mode does exactly what the Termination mode does. The only difference is that the system will wait for incoming phone calls on any of the line interfaces, and after a call has completed, the system will automatically wait for the next call.

"Loop" Mode

In this mode a call is initiated on one interface of the OPERA™ system ("Origin"), while another interface ("Termination") is waiting for an incoming call. As soon as a call is established, a speech sample is transmitted from the terminating side to the originating side. Simultaneously on both sides all incoming signals are recorded to files. The signal recorded on the originating side is usually used for end-to-end quality measurements. The signal recorded on the terminating side contains the echo from the network, as well as the side tone (=desired echo with short delay). The near end signal is usually used for echo measurement. This operating mode is called the "**Loop**" mode.

Figure 4.15 shows the OptiCall window configured for loop measurements. By using the radio buttons in the upper left corner of the OptiCall™ window ("Call Type"), select the operating mode you want to use. In the edit field next to **Number to dial** enter the number that shall be dialled by the system. Numbers can be entered plus an additional "**,**" at any position to insert a **pause** into the dialling sequence. This may be required on some PBXs since OPERA™ will immediately start dialling as soon as a dial tone is detected. For some systems this might be too fast. Please refer to section "Common Settings" above for a description of the other parameters.

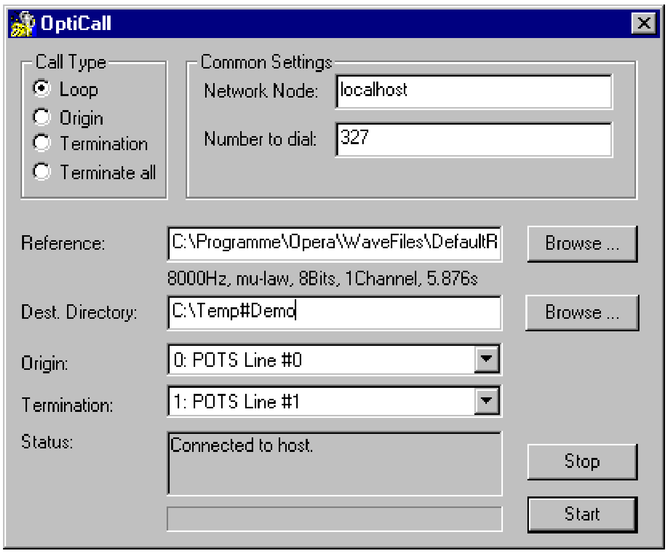


Figure 4.15: The OptiCall window in loop mode configuration

Note:

By using OptiCall – Telephony Standard View, it is always the terminating side of a call that is talking. This is also true when using the audio interfaces. When it is necessary for the other party to speak , then use the Expert View

Next choose the interface which originates the call and the interface which is terminating the call. Click with the mouse on the drop down list boxes for **Origin** and **Termination**. The options shown may look similar to **Figure 4.16**. Each entry in this list box starts with an index. This index is the line number, which is used to uniquely identify on which interface each file was recorded. Following is a short text that describes the type of interface. Names starting with **POTS** refer to the analog loop start interfaces. Names starting with **Audio** identify the audio interfaces. The list of available devices may depend on which hardware is installed on this OPERA™ system. The list always shows the interfaces available on the machine that was selected as the **Network Node**.

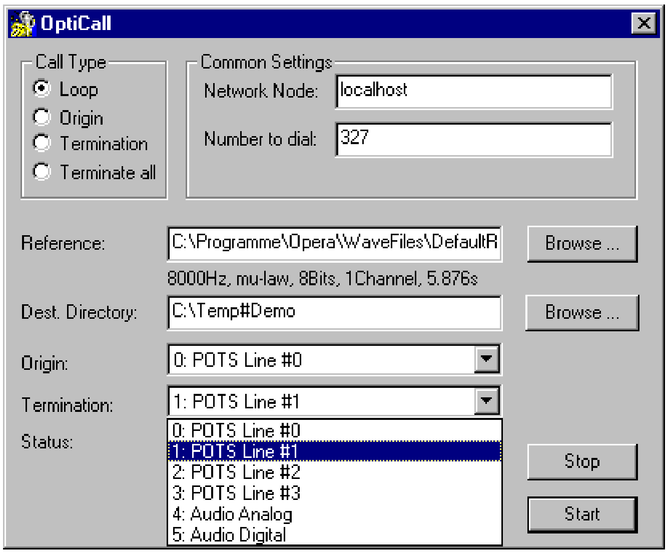


Figure 4.16: Selection of interfaces

If in our example choose "1: POTS Line #0" as the Origin and "1: POTS Line #1" as the Termination, the resulting file names will be:

c:\temp\demo—line0.wav	Data recorded at the originating side (degraded speech signal)
c:\temp\demo—line1.wav	Data recorded at the terminating side (Echo signal)

If "4: Audio 1 Analog" as the Origin and "6: Audio 2 Analog" as the Termination were chosen instead, the resulting file names would be:

c:\temp\demo—line4.wav	Data recorded at the originating side (degraded speech signal)
c:\temp\demo—line6.wav	Data recorded at the terminating side (Echo signal)

"Origin" Mode

In this mode, distributed systems are used to monitor the quality. A call may be initiated from line X of one OPERA™ system to another OPERA™ system. As soon as the call has been established, a speech sample is transmitted from the far end to line X of the near end. Simultaneously all incoming signals on line X are recorded to a file. Line X may be any of the available interfaces of the system initiating the call. The calling OPERA™ system must not necessarily be the local system. Both systems involved in the measurement may be remote systems. The calling as well as the receiving process may run on the same (remote or local) system as well. This operation mode is called the "**Origin**" mode.

Figure 4.17 shows the OptiCall™ window after **Origin** had been selected. The operation is the same as in the loop mode, except for the missing selection of the interface that terminates the call. When using the Origin mode, no speech file will be send, just the recording is active.

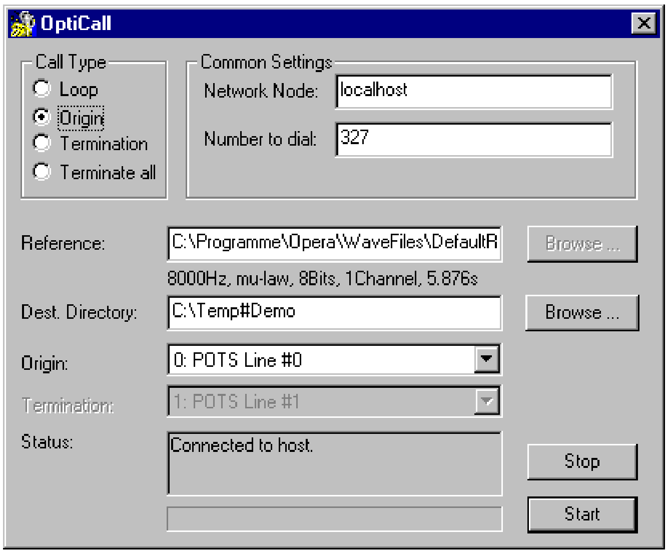


Figure 4.17: The OptiCall ™window in Origin mode

"Termination" Mode

In the termination mode a call from another OPERA™ system is expected on on one specific interface X. As soon as the call has been established, the reference file will be played through the selected interface and simultaneously all incoming signals on interface X are recorded to a file. Interface X may be any of the available interfaces of the system expecting the call. The OPERA™ system waiting for the call must not necessarily be the local system. Both systems involved in the measurement may be remote systems. The calling as well as the receiving process may run on the same (remote) system as well.

Figure 4.18 shows the OptiCall™ window after **Termination** has been selected. Operation is the same as in the **Origin** mode, except for the missing **Phone Number** field that is not required for terminating a call.

Since in the Termination mode OptiCall™ is sending and receiving simultaneously on the same interface, this mode may also be used for assessing audio codecs with a single Audio Interface Option.

Hint:

By using OptiCall™ in the termination mode echo measurements may be performed to remote sites even if there is no other OPERA system available. Simply start OptiCall™ and ask somebody on the remote site to call your OPERA system using a regular telephone. OptiCall™ will answer the call, play the speech file and record the echo signal. Of course it is advisable that the calling person is either switching off the microphone or at least covering it with the hands.

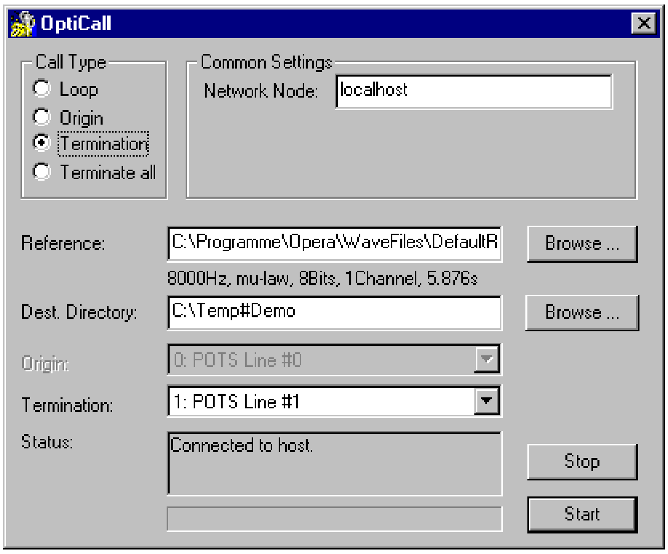


Figure 4.18: The OptiCall™ window in termination mode

"Terminate all" Mode

If OptiCall is put into the **Terminate all** mode, it behaves like in the Termination mode on all the available telephone lines. That means all incoming calls on all lines are answered. After answering any incoming call, the system is waiting for the next call. This procedure is repeated endlessly for all line interfaces. Multiple calls arriving on separate interfaces may be answered simultaneously. The Terminate all mode is especially useful if no IP connection exists between two OPERA systems. In this case it is enough to put the remote system in the Terminate all mode and use the local machine in the Origin mode. Since the terminating side is sending the speech file, voice quality can be evaluated, but the echo signal will be lost since it is recorded on the remote machine.

Note:

The **Terminate all** mode is not available for audio interfaces!

As you can see in **Figure 4.19**, the list boxes for selection of the originating and the terminating interfaces are disabled now. Apart from that, the OptiCall™ window is the same as in the Termination mode.

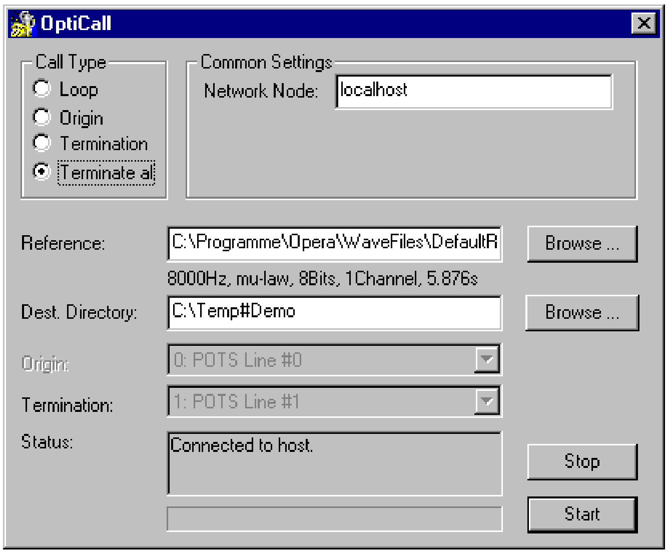


Figure 4.19: The OptiCall™ window in Terminate all mode configuration

Since OptiCall™ will only perform the data acquisition you will have to employ the OPERA™ program for the actual measurement. In OPERA™ the following assignments must be made to achieve proper results when using the PSQM or PESQ algorithm (all referring to the file name settings mentioned above):

Reference file: c:\programme\Opera\DefaultRefFile.wav

Test file: c:\temp\demo-line0.wav

Expert View

The Expert View is exactly the same as the Telephony Standard View, just there are many more options available which are otherwise accessible through the command line interface only. The dialog layout is shown in **Figure 4.20**.

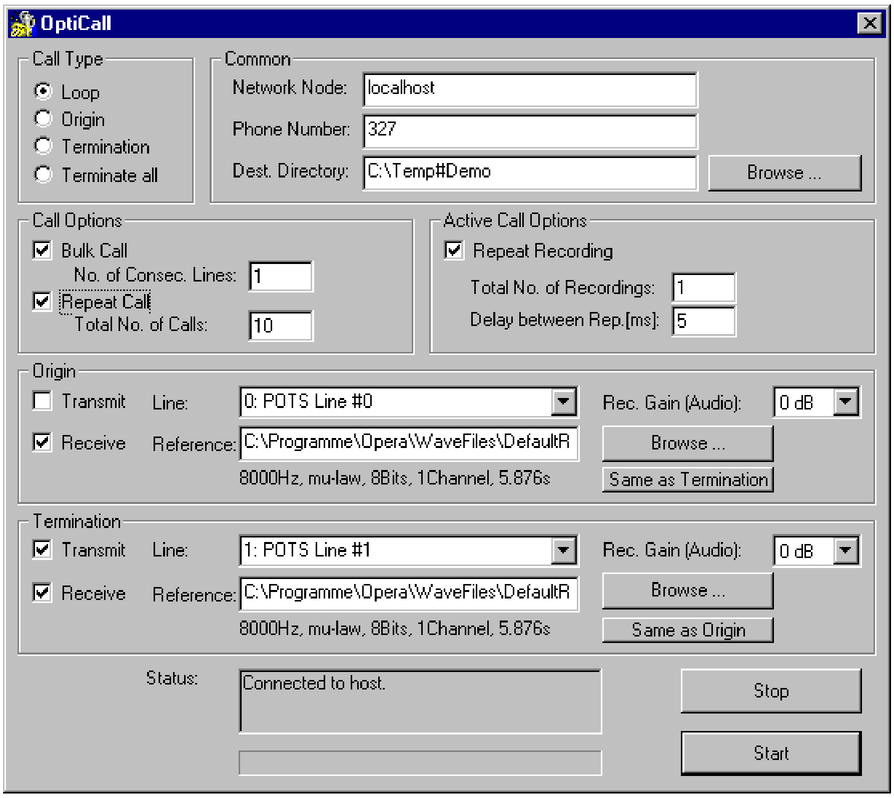


Figure 4.20: OptiCall™ Expert View

While in the Telephony Standard View, the termination side was always sending the speech file. The Expert View allows for both sides independently the selection if a speech file shall be sent and/or received.. You can select this with a check mark besides **Transmit** and **Receive** for both interfaces separately.

With the **Call Options** select

- How many consecutive interfaces are to issue a call **(Bulk Call**) and
- How often each call shall be repeated (**Repetitions**).

If **Bulk Call** is selected and 10 entered for **No. of Consec. Lines**, then the calls will be issued from the interface selected under **Origin**, plus the next 9 interfaces and the calls will be terminated on the interfaces selected under **Termination** plus the next 9 consecutive interfaces. To distinguish the recorded files, -Bulksetxx will be appended to the file name, where xx is replaced by a number indicating the offset to the first originating interface.

All of the above may be repeated several times by checking **Repetitions** and entering the desired number of calls under **Total No. of Calls**. The resulting files will have a –xx appended to their file name, where xx is the index of the repetition, e.g. xyz-2-line0.wav for the second repetition of a call on line 0.

With the settings under **Active Call Options,** adjust how often the speech file is played and recorded during each individual call, before the call is cleared again. If **Repeat Recording** is checked, enter how often the data acquisition shall happen and how many milliseconds delay between the recordings shall be used. The shortest possible delay is in the rage of one to two seconds, depending on the load of the system and the length of the files. A –Repxx will be appended to names of the recorded files, where xx stands for the index of the repetition.

All of the above settings may be freely combined and can be used for audio as well as for telephony interfaces and a combination of both. On audio interfaces call control is skipped and the call is immediately connected.

Another difference compared to the Telephony Standard View is related to the selection of Origin and Termination. In the Expert View it is now possible to choose different reference files for both sides of the call. This is especially interesting if tests are being performed under double talk conditions. In this case it is advisable to use different files in both directions in order to avoid, that echo-cancelers cancel one of the two signals in the system. It is also possible to amplify the recorded signals before the files are written. This is mainly required for recordings made with Opera's audio interfaces, if the device under test operates with very low levels, like e.g. head- or hand set interfaces on telephones / mobile phones. In this case OptiCall™ makes the recording with a bit resolution of 24bits. On this data the gain factor is applied, and afterwards the signals are scaled to 16bit resolution and stored on the disk. App. 70dB gain can be applied without loss compared to an analog amplifier and a subsequent G.711 coding.

Note:

The gain factors are effective for recordings made on the audio interfaces only. For telephony interfaces they are ignored.

Delay Accuracy / DDLC

OptiCall uses a mechanism called DDLC™ to compensate for the unpredictable latencies of the Windows NT operating system. During this process some internal delays may be adjusted. If OptiCall™ detects that it can not compensate for the system latency, it will repeat the call automatically with adjusted parameters. This may happen up to four times. DDLC™ enables OPERA to measure delays with very high accuracy. Experiments showed that 99.5% of the results had a delay accuracy within a range of ±3ms (using the POTS interfaces at 8kHz sample rate). Nevertheless, the accuracy of the results will follow a statistical distribution. For most accurate results we recommend performing a number of measurements and to take the average after brushing off 5% of the results as outliers.

Note:

Even if DDLC is used, there may be a system imminent offset to the delays that are measured. This offset is generally compensated automatically by the system. Since these delays may vary with different board driver versions, please contact OPTICOM if your results tend to show such an offset.

OptiCall™ Command line Parameters

To allow for automated execution from scripts, OptiCall™ can also be started from a DOS window. It understands the following parameters:

/Exec		This must always be the first parameter!
-Loop		Perform a loop call
-Termination		Terminate the call
-Origin		Originate the call
-Cfg	<file name>	Read more parameters from configuration file
-Phonenumber	<phone number>	Phone number to dial
-RefFileOrigin	<file name>	Play file used on calling side
-RefFileTermination	<file name>	Play file used on terminating side
-DestinationPath	<drive:\\path | UNC path>	Destination directory for recorded files
-RootFilename	<root file name>	Root file name used for recorded files
-DoubleTalk		Let both sides of the call talk simultaneously. By default only the terminating side is talking.
-Mirror	<offset>	The call will be terminated on the originating interface plus *offset*.
-NumberOfCalls	<n>	Perform *n* calls
-Bulk	<k>	Perform the call on *k* consecutive interfaces simultaneously.
-NumRecordings	<j>	Perform the data acquisition *j* times during one call.
-DelayBetweenRecordings	<xxx>	Wait *xxx* seconds between two data acquisition phases during one call. Must be used together with *-NumRecordings*.
-RecordGainOrigin	<xxx>	Amplify the signal recorded at the originating side by *xxx* dB.
-RecordGainTermination	<xxx>	Amplify the signal recorded at the terminating side by *xxx* dB.
-Quiet		Suppress output to stdout
-OriginatingLine	<0|..N>	Index of the calling

		interface
-TerminatingLine	<0|..N>	Index of the terminating interface
-Player	<Bitmask>	The bitmask defines which interface is sending (playing) the file. Bit 0 is the originating interface and bit 1 is the terminating interface. Enter 3 for both interfaces.
-Recorder	<Bitmask>	The bitmask defines which interface is receiving (recording) the file. Bit 0 is the originating interface and bit 1 is the terminating interface. Enter 3 for both interfaces.
-Host	<hostname>	Name of the OPERA system on which the program should execute. This parameter is subject to a special network license!
-ListDevices		List all interfaces available for test calls. May be used together with –Host.

- Both play files should be of approximately the same duration (±0.25s).
- To find the proper index of an interface, open OptiCall™ in the GUI mode, and click on the drop down list box as if to change the terminating or the originating interface. Find that all entries in the list box start with a number. This number is the index of the line. In the future, a command line tool that may be used to query the index for a certain interface may be provided.
- If either the terminating or the originating side is an audio interface and not a telephony device, the timing printed to stdout as a result of the call is meaningless. It is only provided in order to maintain compatibility between scripts.

Example (should be written on one line):

- Make a loop call from line 0 to Line 1, dial 01234, use the default reference file and store the results in C:\\temp, as Test-line0 (degraded file) and Test-line1 (echo signal).

```
Opticall /Exec -Loop -OriginatingLine 0 -TerminatingLine 1 -
Phonenumber 01234 -RefFileOrigin
C:\programme\opera\WaveFiles\DefaultReffile.wav -
RefFileTermination
C:\programme\opera\WaveFiles\DefaultReffile.wav -
DestinationPath c:\temp -RootFilename Test
```

Some Important Hints

There are some important rules that should be obeyed in order to perform successful measurements:

- If a POTS interface is selected for either the terminating or the originating side, you must use an 8kHz, 8Bit G.711 mono reference file.

- The recorded files will always be of the same format as the reference file was.

- Never directly feed back a digital AES/EBU output signal to the input of the same board. (also not indirectly by feeding it through the second audio interface). If the digital input is used, the digital output is synchronized to the clock of the digital input signal. The result of feeding back the output to the input is than a deadlock since no clock is available in this case.

- When using both analog audio interfaces simultaneously, make sure that both boards are synchronized, as explained earlier in this chapter.

- Please note that synchronizing two LynxONE boards works for sample rates above 24kHz only. At lower sample rates the drift between the clock oscillators of the two boards will generally be low enough to perform measurements without additional synchronisation.

- On the LynxONE boards each board may be operated fully independent from the other.

- On the Digigram audio boards, all signals must be fully synchronous.

- Take care to not overload the analog inputs. Using the mixer controls on Windows based PCs will in general degrade the signal quality, since the attenuation/amplification is applied to the digital sources, and not to the analog signals. This will in general result in a decrease of the signal to noise ratio. If possible you should leave the mixers in their factory default settings.

- If you assess PC based equipment, we recommend using the best sound cards available (e.g. the LynxONE boards which are used in OPERA as well).

- Recordings made from the audio interfaces will always be stored as 16bit linear stereo wave files.

4.3 The OPERA™ Framework

4.3.1 The Underlying Generic Algorithm Model

OPERA™ is based on a generic algorithm model, as outlined by **Figure 4.21** below. It consists of two inputs, one for the (unprocessed) reference signal and one for the signal under test. The latter may be, for example, the output signal of a codec that is stimulated by the reference signal. In a first signal processing step the peripheral ear is modelled ("perceptual model", or "ear model"). In a consecutive step, the algorithm models the audible distortion present in the signal under test by comparing the outputs of the ear models. The information obtained by this process results into several values, so called MOVs ("Model Output Variables") and may be useful for a detailed analysis of the signal.

The final goal instead is to derive a quality measure, consisting of a single number that indicates the audibility of the distortions present in the signal under test. To achieve this, some further processing of the MOVs is required which simulates the cognitive part of the human auditory system. Therefore the PEAQ algorithm (see Chapter 5) available for OPERA™ uses an artificial neural network, whereas the also available PSQM and PESQ algorithms (see Chapter 6) use algorithmic descriptions. The quality measure in the case of PEAQ is the Objective Difference Grade (ODG, see Chapter 5), whereas the PSQM and PESQ algorithms return the a Mean Opinion Score (MOS, see Chapter 6).

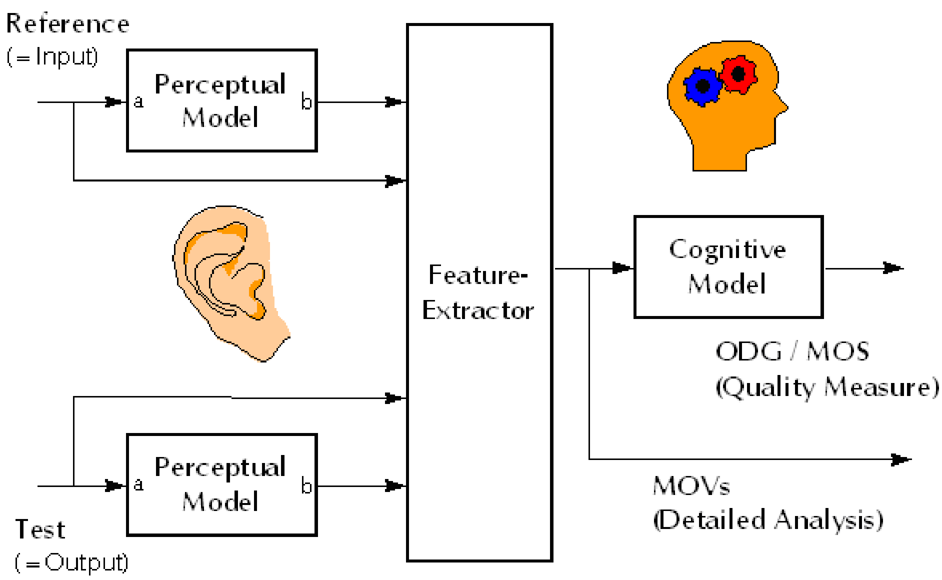


Figure 4.21: The structure of the generic perceptual measurement algorithm

It should be noted that even traditional, non-perceptual algorithms can be described by this generic structure.

4.3.2 The Structure of the OPERA™ Framework

The Difference between an Algorithm and a Measurement System

Scientific proposals as well as international standards like ITU-R BS.1387 (PEAQ) or ITU-T P.861 (PSQM) usually describe a measurement algorithm and do not take into account all the constraints of a realistic measurement situation. For example, all proposals except for P.862 (PESQ) assume that the input signals are time and level aligned, and should not contain any DC offset. Also, most standards do not take care of data acquisition methods, or even the user interfaces. But all these practical requirements add additional complexity to a measurement system that should be suited to perform measurements in the field, under "real world" situations. The resulting consequences for a comprehensive measurement system like OPERA™ are outlined in a block diagram in **Figure 4.22**.

The system structure shown in **Figure 4.22** provides two physical inputs, Input 1 and Input 2, which are suited to interface to different sources, like files on a hard disk, analog or digital audio connections, as well as a/b or E1/T1 telephone line interface. The signals acquired by these inputs must be DC filtered, since DC offsets in general are inaudible, but most measurement algorithms are not able to handle them.

Delay and Gain Compensation

The following step "Delay and Gain Compensation" is as complex as some perceptual algorithms are themselves, especially if a robust and computational efficient implementation is required. Special care has to be taken to treat varying delays as they might occur due to missing synchronization of the input signals, or for instance with Voice over IP connections. As a result of the delay compensation, time aligned signals will be provided, together with values that are suitable to characterize the delay which was introduced by the device under test.

Gain compensation is also mandatory. A level difference between the two input signals will be clearly perceptible in an A/B comparison test but usually it is not intended to be measured as a degradation of the sound quality. Nevertheless, the information that such a gain difference exists and its measured value, may be an important information for the user.

Other features required by a modern measurement tool like OPERA™ are means to generate a test signal (to be seen on the top of **Figure 4.22**), to remote control the system (Remote Control), and also to control it by other programs and scripts (Scripting Interface). Scripts written in an easy to learn macro language (e.g. VBScript, Java Script etc..) will allow for unattended, automated measurements. From the user's point of view it is also desirable that there is a convenient way to copy measurement results and graphs into other documents, and to facilitate the publication, documentation and interpretation of the results.

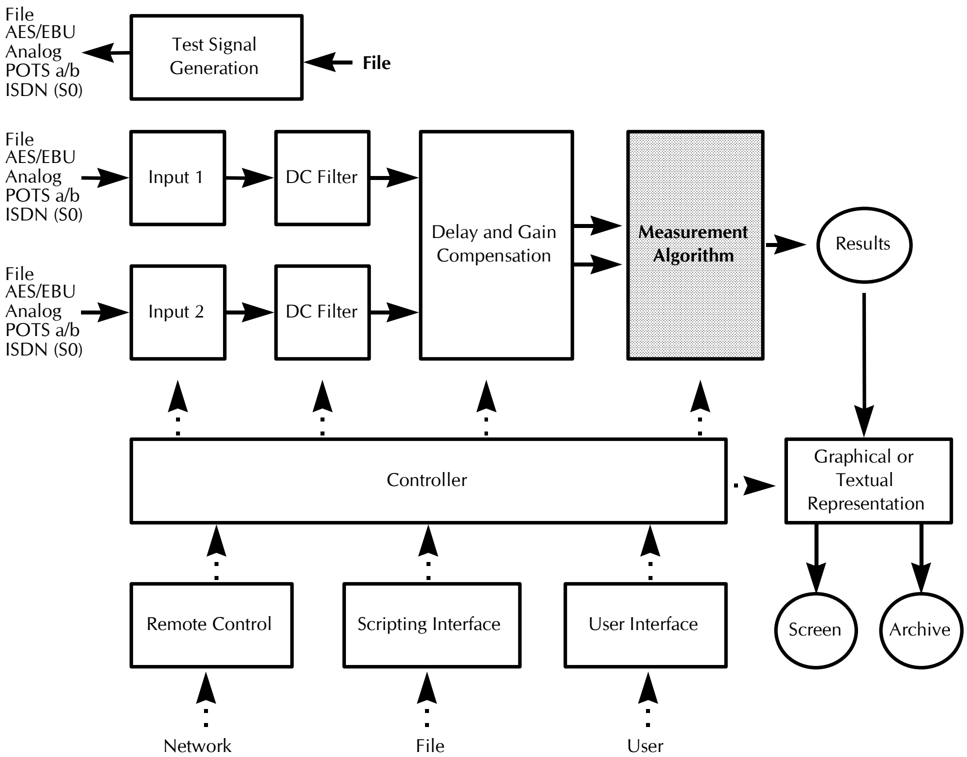


Figure 4.22: Block diagram of the OPERA™ measurement system

As **Figure 4.22** illustrates, the measurement algorithm itself is only a very small part of the entire measurement system. It is obvious that a variety of audio processing algorithms will require the same input handling, filtering, delay compensation and a similar user interface.

Algorithms are just plug-ins

Consequently, the idea behind the framework is to share these parts between various measurement algorithms. The framework will provide the required resources, and the algorithms are just plug-ins, that focus on their real task. And of course it is advantageous that the system can easily be expanded by adding new plug-ins as technology advances and if new, more precise perceptual models become available. The other way round, all implemented algorithms benefit from improvements of the user interface, the signal pre-processing or the implementation of new interfaces. The example screen shot, shown in **Figure 4.23** illustrates the graphical user interface realized under Windows NT for a typical measurement situation.

As mentioned before, there are currently three standardized measurement schemes implemented in OPERA™ - PEAQ, PESQ and PSQM- which will be explained in more detail in the corresponding chapters.

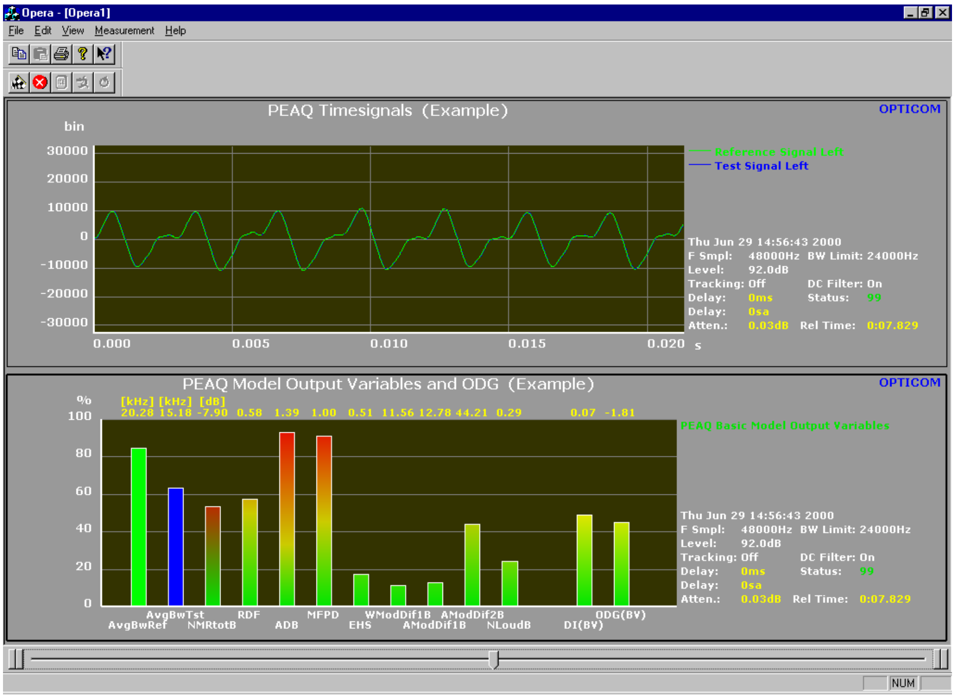


Figure 4.23: An example screen shot of the graphical user interface of OPERA™

The following sections will explain all menu options and command line parameters of the framework.

4.4 Basic Operation

4.4.1 The Main Window

Menu bar and Toolbar

If you first start OPERA™, the screen will look somehow like **Figure 4.24**. OPERA™ uses a standard Windows operating environment, so most users will be familiar with menu bars, toolbars and so on. The main window of OPERA™ contains a menu bar with the entries **File**, **Edit**, **View**, **Measurement** and **Help** on top. A toolbar below the menu bar contains some buttons as short cuts to the menu bar items.

Diagram panes

The core of the main window is the two diagram panes that will display the results of your measurements. Each of them may show different measurement result views. When you have started OPERA™ for the first time, the diagrams display an incidentally chosen diagram type. Otherwise, the system will recall the last diagram type you have displayed before you closed OPERA™.

Display of the Measurement Settings

On the right side of each diagram information about the measurement settings is shown. The content of the display varies depending on the algorithm you have chosen.

Slider

At the bottom of the main window you will find the slider which is used to move the cursor in the time domain of a diagram. At the moment the slider is not active since no measurement has been performed yet. After having performed a measurement, drag the slider with the mouse and scroll through the signal. Other options to move the slider are to click into the grey area besides it, or to click once onto the slider to give it the input focus and then move it with the cursor left and cursor right keys.

Scope Algorithm

The meaning and operation of all menus will be explained in detail later on while presently focussing on those items that are required to perform a first measurement. To that purpose the measurement algorithm "**Scope**", which is available on all OPERA™ Systems will be used. "Scope" does nothing else but performing data acquisition, signal preprocessing and displaying the time signals like an oscilloscope.

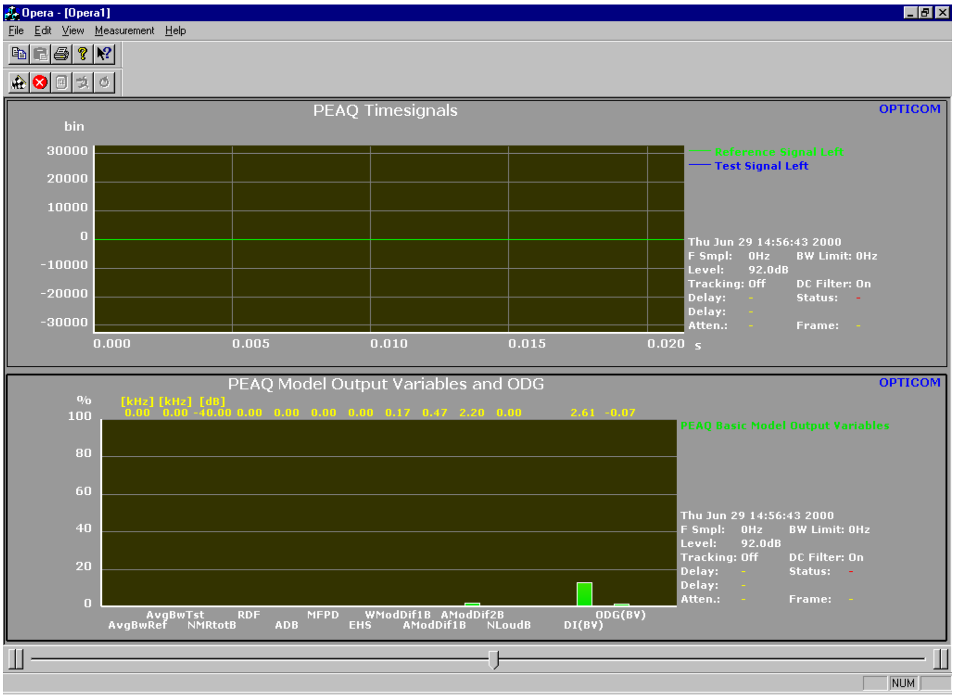


Figure 4.24: OPERA™ main window

4.4.2 How to Select a Measurement Algorithm

To select a measurement algorithm, select the menu **Measurement|Algorithm Parameters...** After doing so a dialog like the one shown in **Figure 4.25** will appear. This dialog will show all algorithms currently installed on your OPERA™ system.

Note:

Depending on which algorithms are actually installed the dialog may look a bit different than **Figure 4.25** The order of the algorithms may be different as well.

Now highlight the radio button next to the algorithm , by clicking on it with the mouse. Now use the "**Scope**" measurement algorithm will be demonstrated.

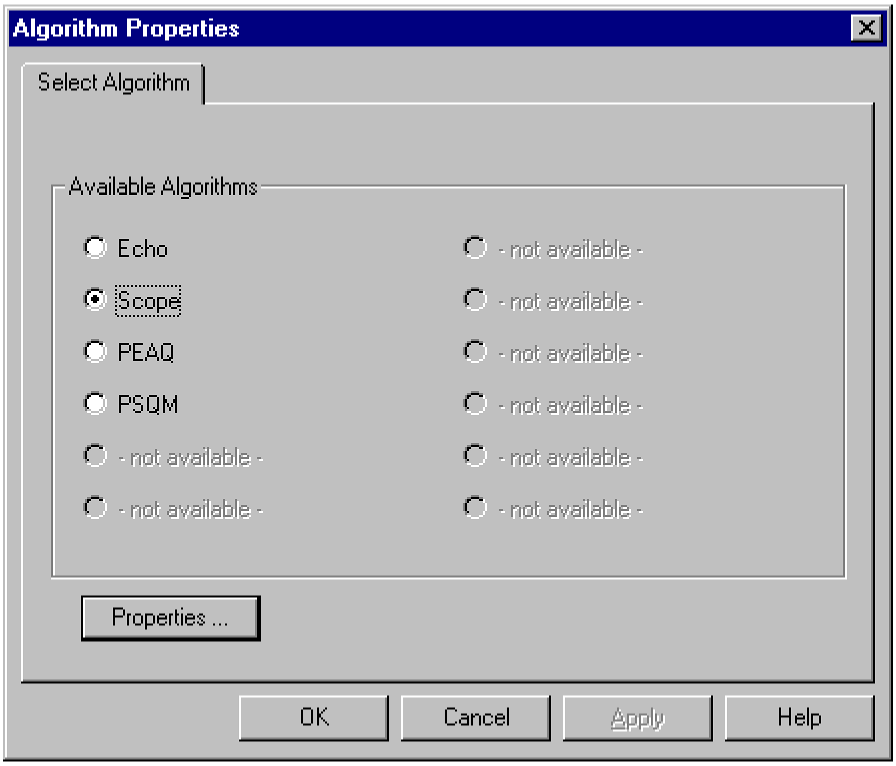


Figure 4.25: Selection of Algorithm Parameters

Properties

After selecting the algorithm ´modify some of the parameters used, by the algorithm by clicking on **Properties**. This will bring up another dialog, allowing enterance of some algorithm specific parameters. These parameters will be explained in detail in the chapters describing each algorithm. For our example there are no properties to configure.

Remember: **It is always a good idea** to check whether the settings are correct, since the system starts with the parameters it used for the last measurement.

Now press **OK** after selecting an algorithm to accept the choice or **Cancel** to leave the settings as they were initially. If changing the algorithm or any of its parameters, a dialog similar to **Figure 4.26** will appear, asking for confirmation of the changes.

Note:

For consistency reasons the diagrams and/or the history buffer will be cleared when changing the algorithm or its parameters.

After having selecting a valid measurement algorithm now, the measurement may begin.

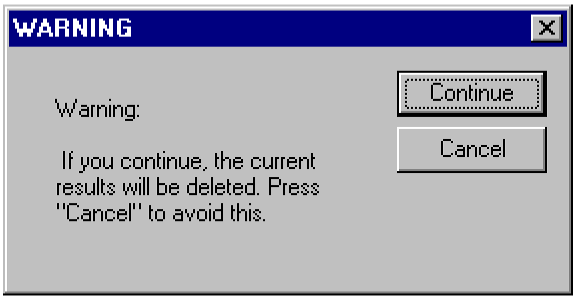


Figure 4.26: Warning Message

4.4.3 How to Start a Measurement

After selecting a measurement algorithm, continue to start a measurement. To do so, there are two possibilities. Either

1. Select **Measurement|Start** from the menu bar,

Or

2. **click** on the appropriate toolbar button shown in **Figure 4.27**.

Figure 4.27: Toolbar button for starting the measurement

This will start the "**Measurement Setup Wizard**". This wizard is a step by step guide through the set-up, making sure that all parameters that are required to obtain correct results are set.

Measurement Setup Wizard Step 1, Physical Input 1

With the first wizard step (see **Figure 4.28**) select the physical source of your **first input signal**. Choose between the sound board and a file as the source for the input signal. Select the Sound Board radio button when performing online measurements. In the case of a file as the input, please note that although the wizard allows you to select between various source formats, only WAVE files containing plain PCM or G.711 (a-/mu-law) are supported by the current version of the OPERA™ system.

First click on the radio button next to "**File**" if this button is not already highlighted. Next enter a valid filename for this input. Either manually type it into the edit field, then select one from the list of the edit field, or click on

Browse to bring up a standard windows file select dialog box. If the file type is valid, the file format parameters in the line below the edit box will be visible. The WAVE file chosen in the example shown in **Figure 4.28** is located in the directory where your OPERA™ system is installed.

Currently OPERA™ can read **WAVE** files containing either plain PCM with **8 or 16bit per sample**, mono or stereo, or WAVE files containing **a-law** or **µ-law** at **8bit per sample** mono. The supported sample rates depend on the selected algorithm.

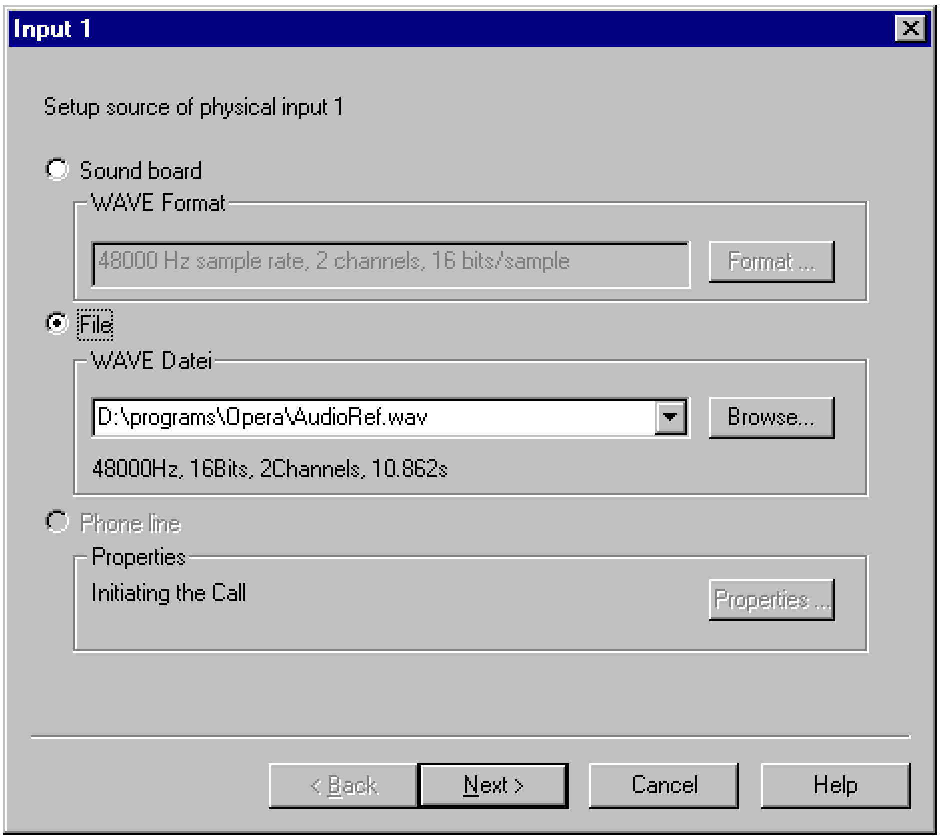


Figure 4.28: Selection of Input 1

After having selected the input source click on "**Next**" to get to the next wizard step.

Note:

The option to select an input signal from an audio soundboard is only available to users of OPR-110-xxx-x OPERA™ systems.

Measurement Setup Wizard Step 2, Physical Input 2

Determine the **physical input source 2** with this step. The dialog looks almost the same as the one in step 1, also the operation is the same. The only difference is, that this input may be **switched off**, if the second source is not required. For instance this is the case if the reference signal is in the left channel and the test signal in the right channel of a file. **Figure 4.29** shows a screen dump of this wizard step.

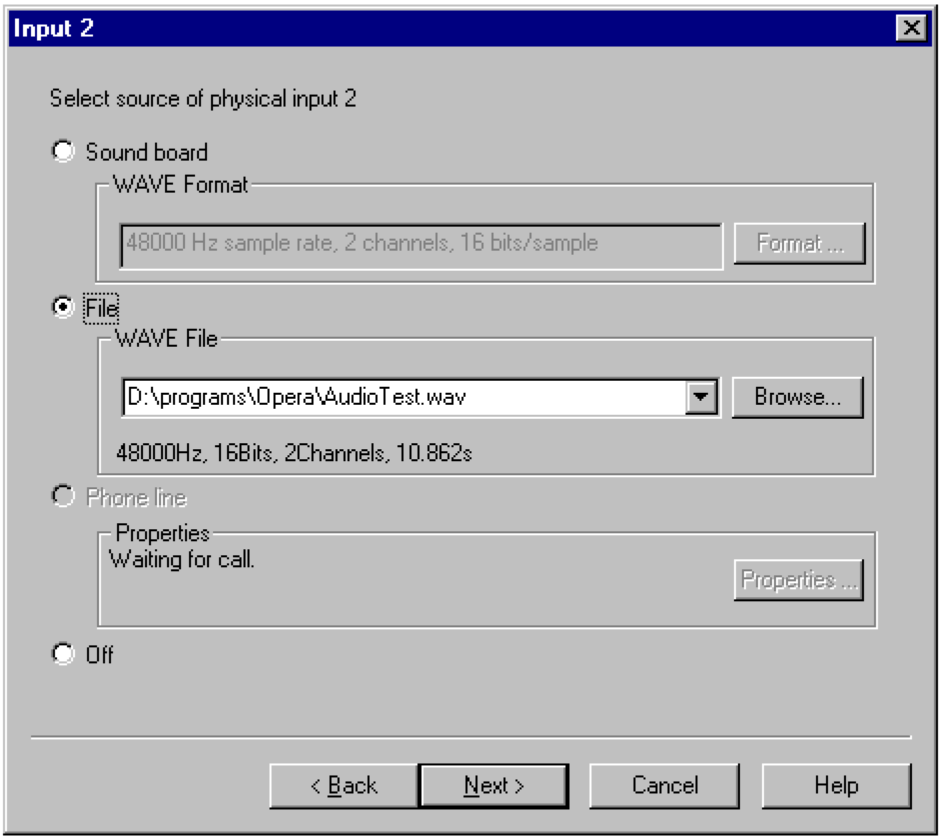


Figure 4.29: Selection of Input 2

Measurement Setup Wizard Step 3, Mapping between physical inputs and logical measurement signals.

OPERA™ has a **built-in multiplexer** that works like a crossbar switch matrix. This multiplexer separates the physical input signals from the signals used as the input signals of the measurement algorithm. This allows, for example, using only one input file that contains the reference signal in one channel and the test signal in the other channel. Also, in the case that the device under test is swapping the channels, the multiplexer can be used to correct this. The chapter on applications will show other useful example settings.

Figure 4.30 shows a screen shot of this wizard step. The dialog is organized like a crossbar switch matrix. Each highlighted radio button indicates a connection between one of the input signals and one of the measurement signals. The example shown in **Figure 4.30** represents a **1:1** relation between the physical inputs and the logical measurement signals. This means that input 1/left is connected to reference/left, input 1/right to reference/right, input 2/left to test/left and input 2/right to test/right.

If not all physical input channels are available, the according radio buttons are disabled (e.g. if the input signals are mono, or input 2 is switched off).

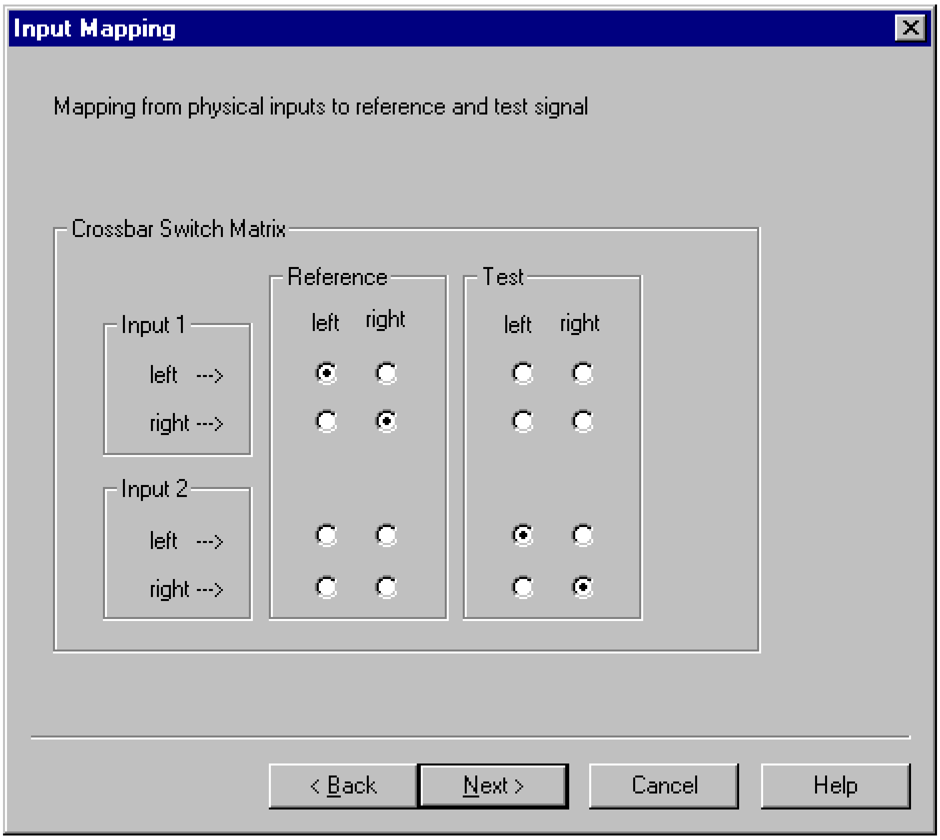


Figure 4.30: Input Mapping

Measurement Setup Wizard Step 4, Delay compensation and signal preprocessing

In this wizard step various parameters of the signal preprocessing applied to reference and test signal can be influenced, before they are processed by the measurement algorithm. The appearance of this dialog is shown in **Figure 4.31**.

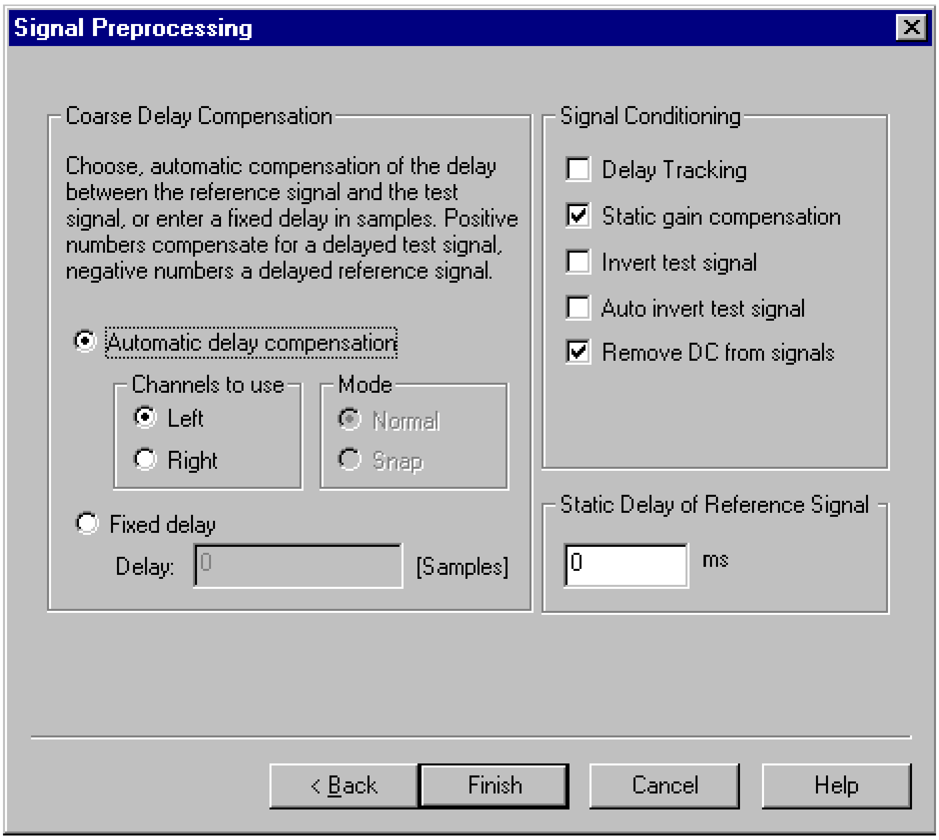


Figure 4.31: Signal Preprocessing Options

Automatic Delay Compensation / Time Alignment

In the group box **Coarse delay compensation** choose the way in which the average delay between the reference and the test signal is determined and applied to the signals. By selecting "**Automatic delay compensation**", this will be done automatically. In this case the radio buttons "**Left**" and "**Right**" determine on which channel of a stereo input signal the delay detection is performed. Nevertheless the delay found will be used for compensation on both channels. The delay range that can be compensated for in this mode is depending on the measurement algorithm used.

The maximum delay that can be compensated for automatically is ±1000 ms for PSQM and PEAQ offline measurements. For PEAQ online measurements it is ±500 ms. PESQ uses a different time alignment algorithm and can compensate for to approximately ±20000 ms. For further information, please refer to the Technical Specifications Section.

When performing **online measurements**, two modes are available for the automatic delay compensation, the **Normal Mode** and the **Snap Mode**. In the **Normal Mode**, the delay between the reference and the test signal is permanently checked. When the delay has changed, the new delay is calculated which can take a duration of several frames. During this time, the signal data is not evaluated by the measurement algorithm.

In the **Snap Mode**, the delay is determined and compensated only once. This delay is maintained for the rest of the measurement or until **Snap Again** (see below) will be activated. Please note: these functions are disabled when **file-based measurements** are performed.

Note:

The toolbar (see **Figure 4.32** for the button symbol) and menu option **Freeze Delay** takes effect in Normal Mode of the Automatic Delay Compensation. When activated, the current delay is used for the rest of the measurement.

Freeze Delay

Figure 4.32: Toolbar button for the command Freeze Delay

Note:

The toolbar (see **Figure 4.33** for the button symbol) and menu option **Snap Again** takes effect when the Snap Mode of the Automatic Delay Compensation has been chosen. When performing **Snap Again**, the delay is re-determined and used for the delay compensation from now on.

Snap Again

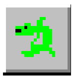


Figure 4.33: Toolbar button for the command Snap Again

Fixed Delay

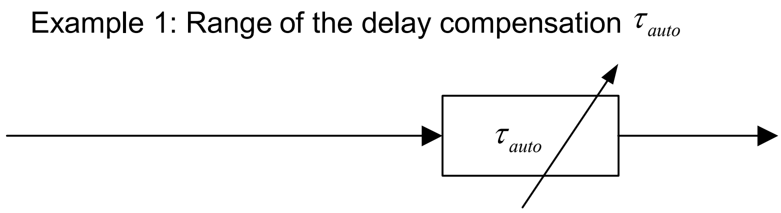
When choosing "**Fixed delay**", manually enter the delay by which the reference signal will be delayed in order to compensate for the delay introduced by the device under test. Enter the delay in samples. The delay range that can be compensated for in this mode is depending on the measurement algorithm used. For the corresponding values please refer to the Technical Specifications.

Note:

Please note that negative delays may be entered. In this case the test signal will be delayed.

Static Delay of Reference Signal

The right field at the bottom of the dialog window shown in **Figure 4.31** shows an edit control titled **Static Delay of Reference Signal**. To compensate for delays longer than ± 1 second (± 500 ms in the case of PEAQ online measurements), an additional static delay can be applied here. The range of the static delay is ± 10000 ms. As indicated by the examples in **Figure 4.34**, the static delay is introduced in the signal flow before the automatic delay module.



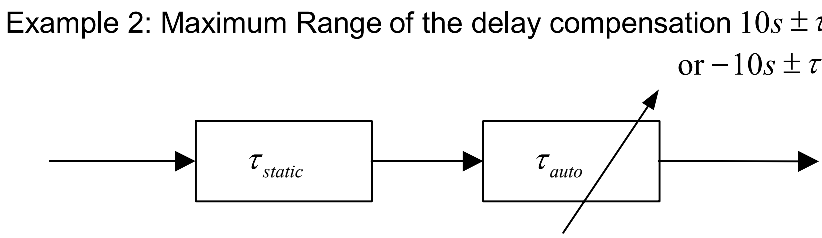


Figure 4.34: Total range of the delay compensation using only Automatic Delay Compensation (Example 1) and using Static Delay and Automatic Delay Compensation (Example 2)

Delay Tracking

In the top right group box **Signal Conditions** some additional functions are available to treat your measurement signals. When the **Delay Tracking** option is selected, OPERA™ tries to compensate for small delay variations on a frame by frame basis.

> **Note:**
>
> This feature should be used carefully since the resulting delays may be less reliable.

The complete functionality of the delay compensation – using Static Delay, Automatic Delay Compensation, Fixed Delay and Delay Tracking - is depicted in **Figure 4.35**.

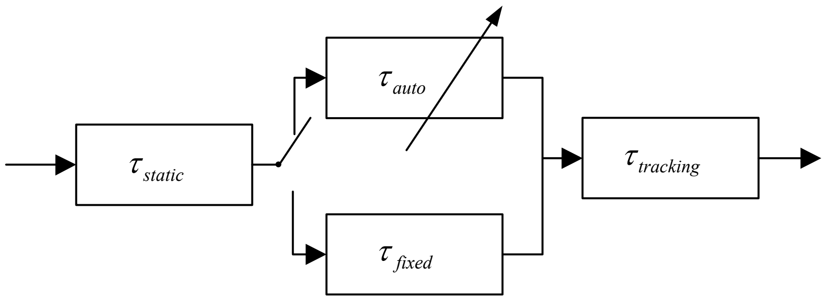


Figure 4.35: Block diagram of the complete functionality of the delay compensation

Static Gain Compensation

If **Static Gain Compensation** is checked, OPERA™ tries to determine the overall attenuation between the two input signals and compensates for it. This option should always be checked, if not otherwise stated in the description of the active measurement algorithm. The total range that can be compensated for is ±60 dB.

Invert Test Signal

If **Invert Test Signal** is checked, the test signal is always inverted (multiplied by –1.0).

Auto Invert Test signal

If **Auto Invert Test Signal** is checked, the test signal is automatically inverted if the system detects that reference and test signals are of different polarity. Enabling this function also has an influence on the Automatic Delay Compensation. The Delay compensation does not accept delays with a 180° phase shift, unless **Auto Invert Test Signal** is selected.

Remove DC from signals

If **Remove DC From Signals** is checked, a DC filter is applied to all input signals, before they are processed by the measurement algorithm. If the signals show a DC bias, the filters will need approximately 2000 samples to settle. DC filtering may be used safely for signals containing no DC bias. The only side effect may be that applying the DC filter requires additional computational power.

Actual Measurement

After confirming the last wizard step by pressing "**Finish**", the actual measurement starts. If automatic delay compensation was selected, there may be some seconds delay before any action becomes visible. After that the process of the measurement is visible, since the diagram panes will be updated on a regular basis. After the measurement is completed, a small message box will appear confirming successful termination, consequently, confirm. Use the scrollbar at the bottom of the screen to scroll through the history buffer to view the measurement results of past frames. The number of frames stored in the history buffer depends on the algorithm you actually selected. The values shown in the diagrams will be explained in the chapters describing the algorithms.

Reset Averaged Values

Note:

Many measurement result values in OPERA™ are averaged from the start to the current point of time in the measurement. When choosing the menu option **Reset Averaged Values** (see **Figure 4.36** for the toolbar button) during a measurement, these values will be reset to zero and the system begins to re-determine the values starting from that point of time. This option is reasonable for real time measuring over a long period of time.

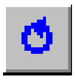


Figure 4.36: Toolbar button for the Reset Averaged Values command

To stop the measurement before the entire files are processed, do so by either selecting **Measurement|Stop** from the menu bar, or by pressing the related **toolbar button** shown in **Figure 4.37**.

Figure 4.37: Toolbar button for stopping the measurement

We will now proceed by selecting the measurement result values we want to display in the diagrams on the screen.

4.4.4 How to Display the Results

To select a diagram you may either

1. Click with the right mouse button on one of the diagram panes and select **Diagram Type** from the pop up context menu (see **Figure 4.38**).

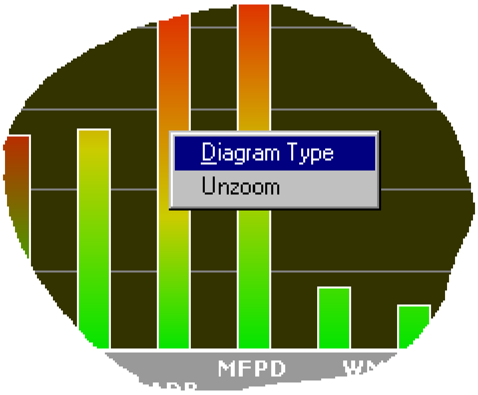


Figure 4.38: Selecting the diagram type from the pop-up menu

or

2. Select a diagram pane by clicking with the left mouse button on it (this makes it the active diagram, indicated by a fat border around the diagram) and then select **View - Diagram Type** from the menu bar.

A wizard-style dialog similar to **Figure 4.39** will appear. This dialog presents all the measurement diagrams that are provided by the currently selected algorithm. This first step of the wizard is also referred to as **Result Type** dialog.

Note:

The actual contents of the dialog shown in **Figure 4.39** is depending on the algorithm you have selected and will be explained in the chapters describing the corresponding measurement algorithm. Our example uses **Timesignals** since these diagrams are the same for all currently implemented algorithms (except for PESQ).

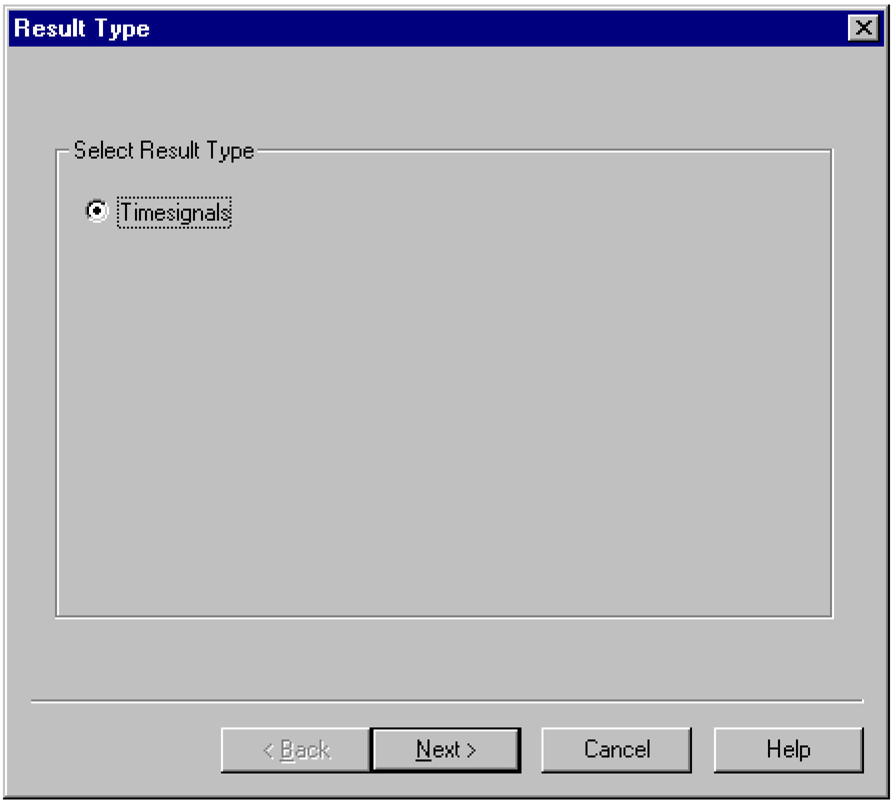


Figure 4.39: Select Result Type Window

Now highlight the radio button next to **Timesignals**, and press **Next**. This leads to the next wizard step, the **Signal Select** dialog.

The **Select Signal** dialog (see **Figure 4.40**) allows a selection of channels and input signals required for the results in one diagram. Modify the selection by clicking with the left mouse button on any of the option buttons. This will add or remove the check mark in the button. A checked button means that the results for the selected signal will be drawn in the diagram. In **Figure 4.40** the results for the left channel of the reference and the test signal were selected.

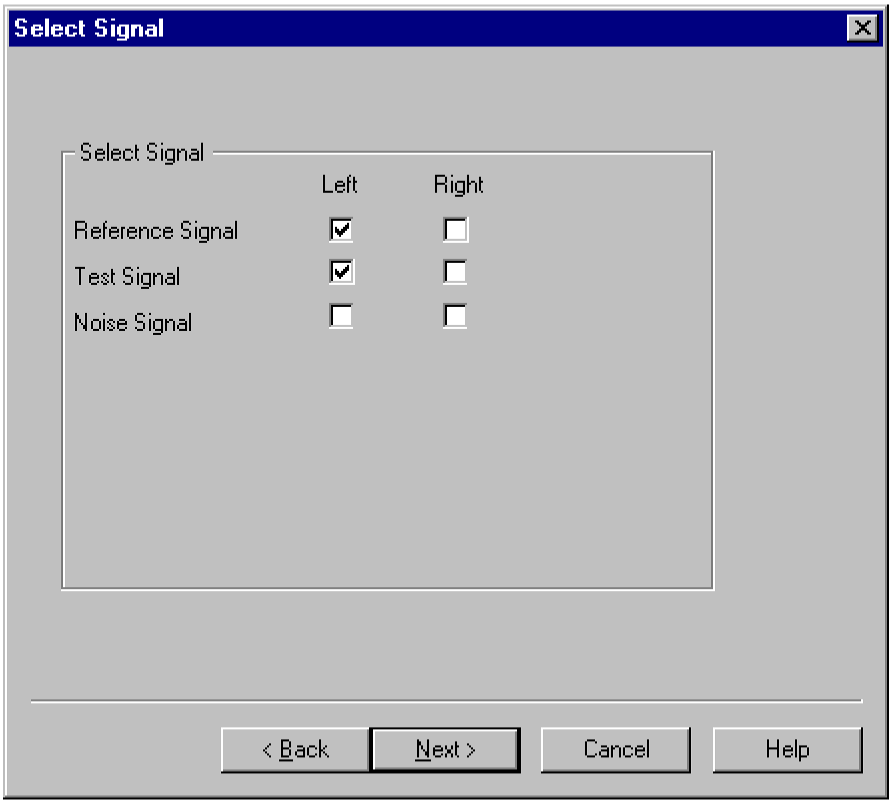


Figure 4.40: Select Signal Window

Pressing **Next** again leads to the next step, the **Result Style** dialog (see **Figure 4.41**). Here select the way data is shown on the screen. Usually this is identical to selecting the units of the diagram axes. For the time signals choose between a binary, linear representation in which the input signals are always scaled to **[-32768 ... +32767]**, or a **dB FS** scale.

Note:

Independent of the input data format, samples are always converted to **16bit/sample**. This means that 8bit/sample data are multiplied by 256 before they are processed any further.

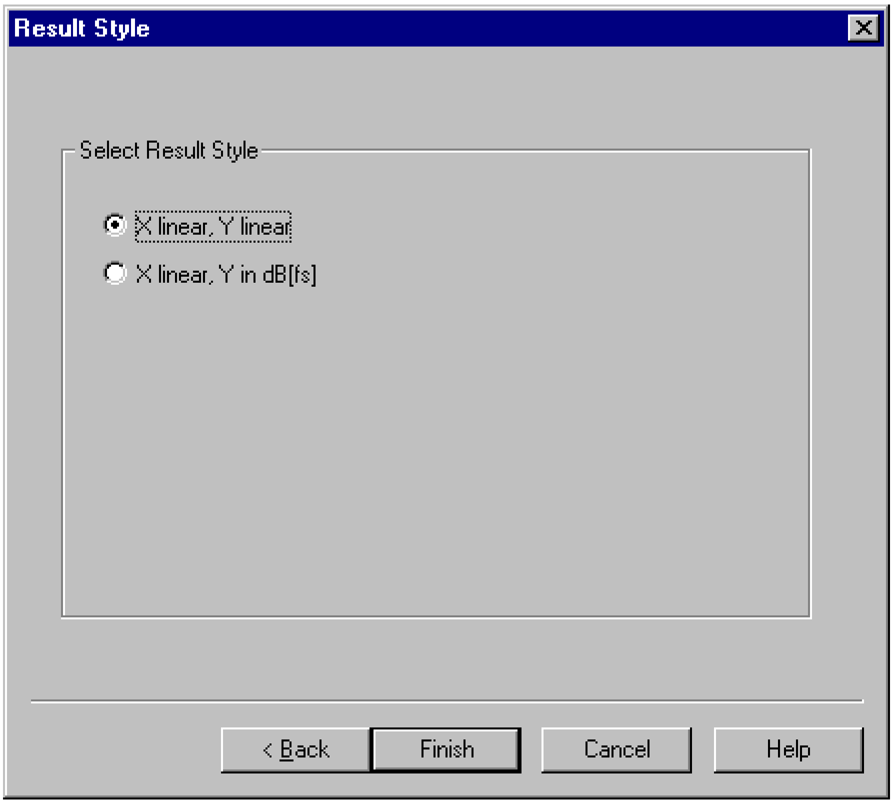


Figure 4.41: Result Style Window

After this last step click on **Finish** and the selected diagram will appear in the diagram pane. Follow the same procedure for the second diagram pane, selecting the same or other results and/or signals for the second diagram pane.

> **Note:**
>
> At any time while the wizard dialog is active click **Back** to go to the previous wizard step, or click on **Cancel** to leave the wizard without performing any changes.

> **Note:**
>
> Depending on the selected result type in the "**Result Type**" dialog, not all wizard steps may be available.

Display of Measurement Settings

The display of the measurement settings shown on the right side of each diagram is depicted in **Figure 4.42**. The meaning of the values is as shown in **Table 4.1**:

Displayed Values	Interpretation
Time:	The time when the measurement has been finished.
F Smpl:	Sample rate of input signals
Tracking:	Status of the delay tracking function (on or off)
DC Filter:	Status of the DC filter (on or off)
Delay:	Delay in ms (first from top) as well as in samples (second from top)
Status:	Reliability of the automatic delay compensation (0..100%, Fixed=fixed delay set).
Atten:	Level difference between reference and test signal (dB)
Rel Time	Current point of time in the measurement

Table 4.1: Interpretation of the displayed values

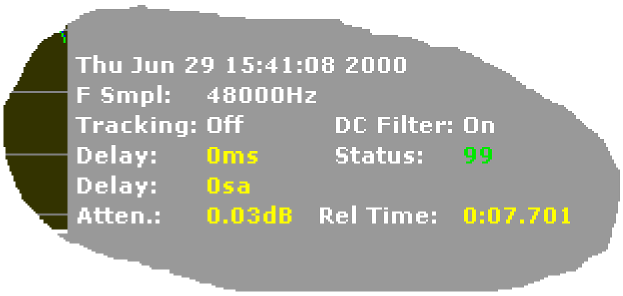


Figure 4.42: General information related to the current measurement settings

4.4.5 Setting Markers in Diagrams

In most diagrams small markers can be set by double clicking with the mouse anywhere in the diagram, or selecting **Set Marker** from the context menu. A small cross as in **Figure 4.43** will appear at the place of the click, with it's coordinates written next to it. When setting a second marker into the same diagram, not only the coordinates will be shown, but also the difference between the two markers. To remove a marker again, either double click on it, or use the context menu (right mouse button), which also enables you to clear all markers at once.

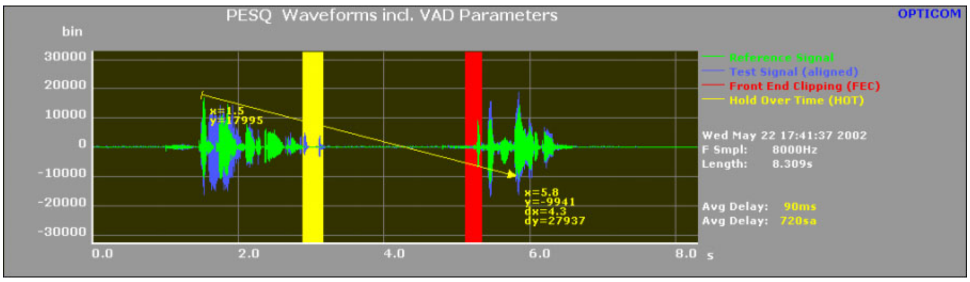


Figure 4.43: Markers

4.4.6 Logging Results

After a measurement has been performed, the final results that were achieved, can be logged to a text file. The text file contains tab separated values and can be directly opened with Excel for further evaluation. To log the current results, simply make a right mouse button click on one of the diagrams and select **Log Final Result** from the context menu. Alternatively choose **File | Log Final Result** from the menu bar. In the file select box that pops up, choose the file to which the results shall be written and wether an eventually existing file shall be deleted or wether the results shall be appended to it. The file select dialog also allows adding an optional comment to the log entry.

4.4.7 Performing Online Measurements in Realtime

After connecting your cables and making the settings in the mixer of the audio board, according to chapter 4.2 measurement may begin. During the first two steps of the Measurement Setup Wizard, select the radio button **Soundboard** (see **Figure 4.44**). When changing the current settings of the signal format, click on the **Format** button and the dialog shown in **Figure 4.45** will appear. Here change the **Sample Rate**, the number of **Channels** or the bit resolution (**Bits per Sample**) of your digital signal format. When having analog signals, the format depends on the characteristics of the audio interface board.

At the bottom of the dialog you see two radio buttons, labelled "Analog", respectively "Digital (AES-EBU)". These buttons are for defining an analog or a digital input format. If the system has a Digigram audtio board installed, these radio buttons are disabled, since the selection of analog or digital format has to be made in the mixer dialog of the Digigram audio interface. Please refere to chapter 4.2 for details.

Note:

Please note the input settings become valid with the start of the next measurement. By default the inputs are switched to analog after each start of OPERA™.

Note:

The available sample rates for online measurements are defined mainly by the selected measurement algorithm. PEAQ e.g. limits these to 44.1 and 48kHz.Digital input signals are limited to 32, 44.1 or 48kHz due to the AES/EBU format. In case of digital input signals, the sample rate and format selected with the start wizard must match the sample rate and format of the AES/EBU data.

The further proceeding of the Measurement Setup Wizard is as described in chapter 4.4.3.

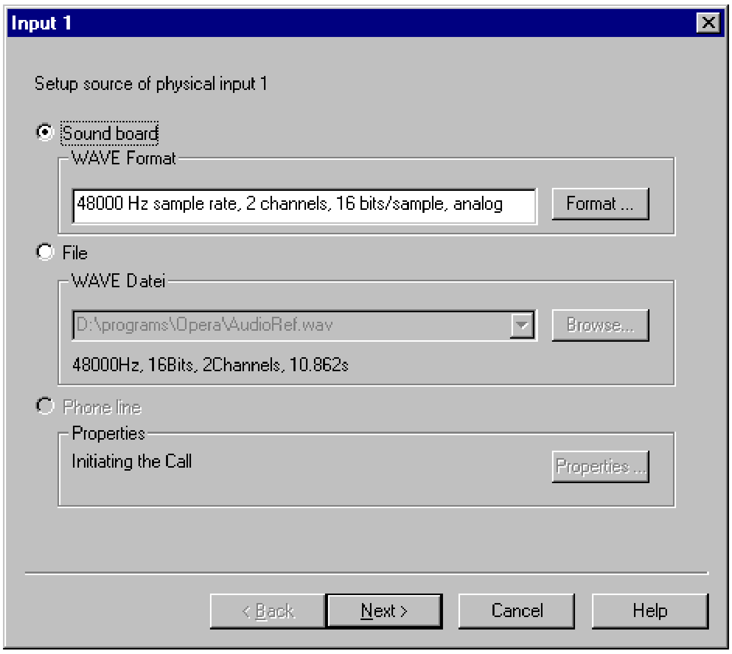


Figure 4.44: Measurement Setup Wizard Step 1. Selection of the Soundboard

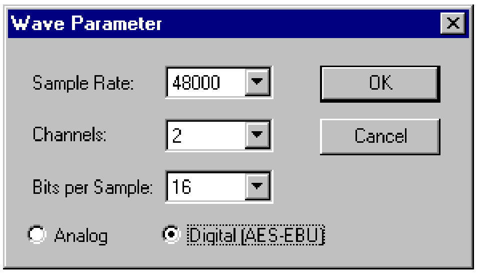


Figure 4.45: Format dialog

4.4.8 Measuring only parts of the Input Files

OPERA™ allows the use of a certain part of the input signals for the measurement only. To use this feature, select the menu option **Measurement|Trigger**. The dialog shown in **Figure 4.46** will be displayed .

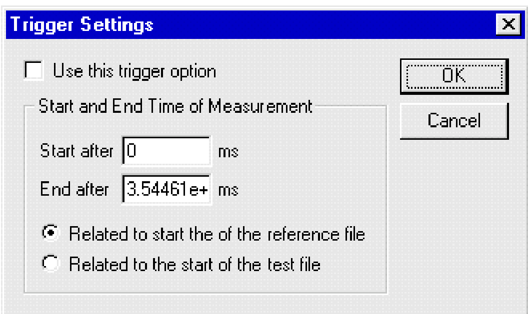


Figure 4.46: Trigger Dialog

When using this option, always check the **Use this trigger option** field. Now enter the desired start and end offsets in ms into the fields **Start after** and **End**

after respectively. Finally choose whether the start and end time should be related to the beginning of the reference or to the beginning of the test file. This selection is required since one of the two signals may be delayed by the time alignment algorithm.

4.4.9 Printing

When making a printout of the measurement results first check the printer settings. Select the menu option **File|Print Setup...** The standard windows Print Setup dialog will appear. Close this dialog with the button **OK**.

Before printing, check what the printout of the measurement results will look like. By selecting the menu option **File|Print Preview** another window will open as shown in **Figure 4.47**. By clicking on the Print button on the toolbar the printing will start.

File|Print

Print the diagram view from the menu option **File|Print** or by selecting the **toolbar button** showing the printer symbol. **Figure 4.47** shows both diagram panels and the display of the measurement settings on the right-hand side of the diagrams will be printed.

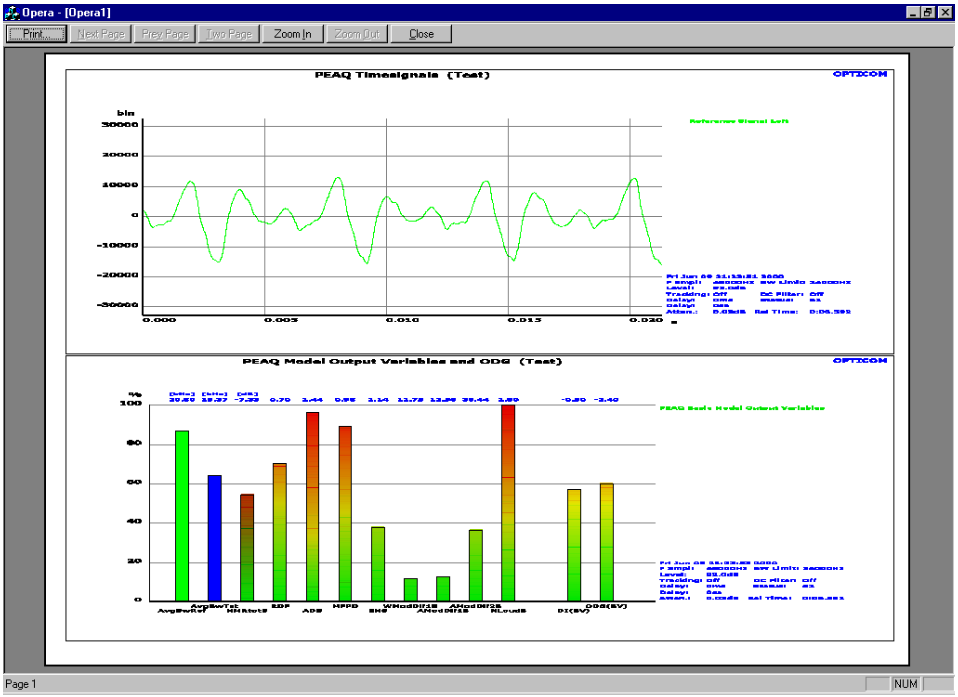


Figure 4.47: The Print Preview window

4.4.10 Exporting Graphs

For further documentation of your measurements export certain diagrams. This is possible by a copy and paste procedure.

Edit|Copy

At first select the diagram pane to be copied. When clicking on a diagram it will get the focus which is indicated by a bold frame around the entire pane. Now select the menu option **Edit|Copy** (or the Copy **toolbar button**) the diagram that has the focus will be put into the clipboard buffer. Paste it into an editor that is capable of processing both, graphics and text.

Edit|Copy (printable)

While the command **Edit|Copy** will copy the diagram as it is displayed in the OPERA™ main window, the menu option **Edit|Copy (printable)** will result in a display that is more suitable for printing as the background colour is changed to white. **Figure 4.48** depicts the result of this command.

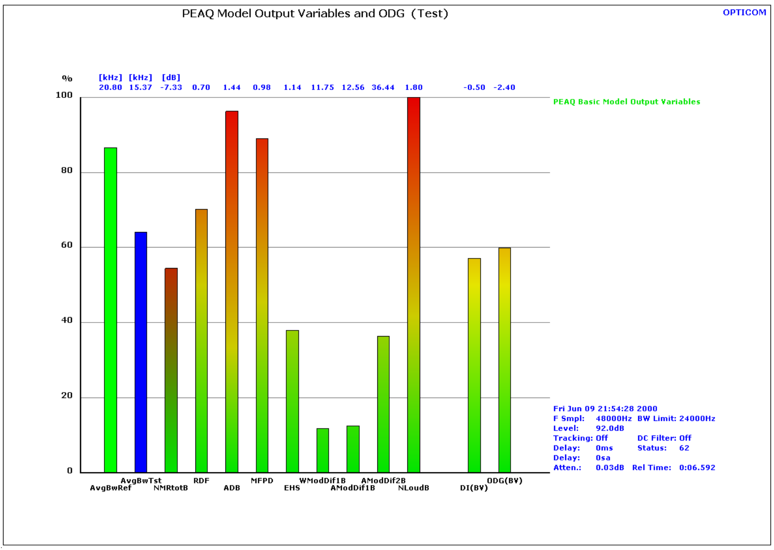


Figure 4.48: Exported measurement graph using the Edit|Copy (printable) menu option

4.4.11 Summary of the Menu Options

File Menu

File|Print...

Prints the diagrams. Use the keystroke Ctrl+P.

File|Print Preview

Shows a preview of would be printed if File|Print had been selected.

File|Print Setup...

Allows printer selection, sets up the current paper size etc.

File | Log Final Result

Write the last measurement results to a log file.

File|Exit

Leaves the OPERA™ program.

Edit menu

Edit|Undo

This menu option currentlyhas no function.

Edit|Cut

This menu option currentlyhas no function.

Edit|Copy

Copies the active diagram to the clipboard, using the same colours as on the screen. You may also use Ctrl+C.

Edit|Copy (printable)

Copies the active diagram to the clipboard, using printer friendly colours.

Edit|Paste

This menu option currentlyhas no function.

View Menu

View|Toolbar

Switches the toolbar on and off.

View|Status bar

Switches the status bar on and off.

View|Results...

This menu option currentlyhas no function.

View|Info

This menu option currentlyhas no function.

View|Diagram Type

Starts the diagram select wizard to change the diagram type of the active diagram.

View|Unzoom

Resets the zoom factors of the active diagram if any are active.

View|Set/Remove Marker

Sets or removes a marker in/from the current diagram

View|Clear all Markers

Removes all markers from the current diagram

Measurement Menu

Measurement|Algorithm Parameters...

Opens the dialog to select the active algorithm and allows changing of the parameters of the active algorithm.

Measurement|Name Measurement...

Opens a dialog that allows entering a name for the current measurement that will appear on each diagram next to the diagram title.

Measurement|Start

Starts the measurement setup wizard.

Measurement|Stop

Interrupts the current measurement.

Measurement|Freeze Delay

Freeze Delay takes effect in Normal Mode of the Automatic Delay Compensation. When activated, the current delay compensation is used for the rest of the measurement.

Measurement|Snap Again

Snap Again takes effect when the Snap Mode of the Automatic Delay Compensation has been chosen. When performing **Snap Again**, the delay is re-determined and used for delay compensation from now on.

Measurement|Reset Averaged Values

Selecting **Reset Averaged Values** during a measurement will reset the measurement values to zero and the system will begin to re-determine the values up from that point of time. This option is reasonable for real time measuring over a long period of time.

Measurement|Trigger

Currently there is only one trigger option available. This trigger selection of a specific part of the signal for the analysis.

Help Menu

Help|Help Topics

Starts the online help.

Help|About OPERA...

Opens the About box that contains license information, the version number and the copyrights.

4.5 Performing Measurements From Batch Files

When performing measurements on a huge number of input files it is much easier to start your OPERA™ system from a batch file. OPERA™ then will process all files in a row, which will save a lot of time in comparison to performing the same task manually. To support this, OPERA™ understands a number of command line parameters. These parameters may also be written into a batch file and contain comments. For information about the syntax of batch files, please refer to the corresponding help topic in your Windows help.

4.5.1 Syntax of the Command Line Parameters in a Batch File

To start an OPERA™ measurement with command line arguments use the following syntax:

`opera -Exec` <list of parameters>

For every measurement type these keywords at the beginning of the line. The list of parameters comprises some of the parameters that are described in this section. It is imperative to put these parameters on one line for each measurement. Another option is to place the list of parameters in a configuration file. and start OPERA like:

`opera -Exec –Cfg` <name of the configuration file>

For a description how to use configuration files see paragraph 4.5.2.

In the following, the syntax of the command line parameters in the list is described. For all parameters inside a command root (i.e. **–Input** or **–Mux**) the keyword of the command root (e.g. "**-Input**") has to be typed only once, at the beginning (lines starting with "REM" are comments).

Section Identifier	Option	Parameter	Description
-Algorithm			
	Name	<PSQM | PESQ | ECHO | PEAQ>	Name of the algorithm to be used
	Settings	"more parameters"	Parameters that algorithm specific. See the algorithms description for details. Note that the parameters must be enclosed in qotes!
-Input			
	Inp=0 File="File1"		File name used for input 1
	Inp=1 File="File2"		File name used for input 2
-Mux			
	InpRefLeft	<0 | 1>	Input used to form the left channel of the refernce signal
	InpRefRight	<0 | 1>	Input used to form the right channel of the reference signal
	InpTestLeft	<0 | 1>	Input used to form the left channel of the test signal
	InpTestRight	<0 | 1>	Input used to form the right channel of the test signal
	ChannelRefLeft	<0 | 1>	Channel of input signal used to form the left channel of the refernce signal (0=left, 1=right)
	ChannelRefRight	<0 | 1>	Channel of input signal used to form

			the right channel of the reference signal (0=left, 1=right)
	ChannelTestLeft	<0 | 1>	Channel of input signal used to form the left channel of the test signal (0=left, 1=right)
	ChannelTestRight	<0 | 1>	Channel of input signal used to form the right channel of the test signal (0=left, 1=right)
-Delay			
	FixedDelay		Use a fixed delay for the measurement
	Delay	<delay>	Specify the fixed delay in samples
	TrackingOn		Switch delay tracking on
	Channel	<0 for left | 1 for right>	Channel used for the automatic delay compensation
	StaticDelay	<Delay in ms>	Additional static delay of the reference signal in ms
-Signal			
	StaticGainOn		Switch the static gain compensation on
	InvertTestSignal		Invert the test signal
	AutoInvertTestSig		Automatically invert the test signal
	DCFilterOn		Switch DC filtering on
-Trigger			
	StartTime	<Start time>	Specify start point of the measurement in ms
	EndTime	<End time>	Specify end point of the measurement in ms
	Channel	<0: Relate to reefernce | 1: realte to test>	Relate start and end point to the beginning of the reference or the test signal
-Out		"<FileName>"	Name and path of result output file
-Append			Append results to existing result output file
–PassThrough		"<Additional Text>"	The additional text will be printed to the result file
-Cfg		"<File name>"	Name and path of a configuration file that contains more command line parameters

4.5.2 How to Use a Configuration File

Create a configuration file containing default values of parameters. When starting OPERA™ with a configuration file use the following syntax:

```
opera -Exec -Cfg <Name and path of the configuration file>
```

The configuration file must have the suffix ".cfg".

A combination of both versions may also be used :

```
opera -Exec -Cfg DefaultPara.cfg -Input Inp=0
                 File=InputFile1.wav
```

The corresponding setting made in DefaultPara.cfg is overwritten by the **-Input** command in this instance.

Comments can be inserted in the configuration file using the character ";" at the beginning of a line.

Note:

The command line parameters in the batch and configuration files are case sensitive.

4.5.3 Example RunPsqm.bat

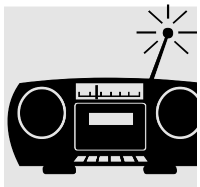
```
@echo off
rem batch file to compute PSQM from two stereo input files
rem
rem Parameters:
rem
rem   RunPsqm   <File1> <File2> <Outputfile>
rem
rem         File1:   File that contains the reference signal
rem
rem         File2:   File that contains the test signal.
rem
rem         Outputfile: Results are stored in this file. If
rem                     it exists already results are
rem                     appended to it, otherwise it will be
rem                     newly created.
rem
rem
echo *************************************************
echo ***** RunPSQM        V1.0  (c) OPTICOM, 1998   ***
echo *************************************************
echo.
pushd
echo *** TODO: change working dir according to where
rem      OPERA.exe is!
rem c:
rem cd "\programme\opera"
echo ... Processing file %1
Opera -Exec -Algorithm Name=PSQM -Input Inp=0 File=%1 Inp=1
   File=%2 -Mux InpRefLeft=0 ChannelRefLeft=0 InpTestLeft=1
   ChannelTestLeft=0 -Signal StaticGainOn AutoInvertTestSig
   -Out %3 -Append
popd
echo Done!
```

Chapter 5

5 WIDE BAND AUDIO QUALITY TESTING

Measuring the Perceived Audio Quality (PEAQ) of e.g. MPEG Encoded Music Signals.

A description of all specific concerns for assessing wide band audio quality follows. In addition, the fundamentals of the corresponding measurement method is explained. Finally, assistance with the first measurement applications is provided at the end of this chapter.



5.1 What To Know About Testing Wide Band Audio Quality

Audio quality is one of the key factors when designing a digital system for broadcasting. The rapid introduction of various bit-rate reduction schemes has led to significant efforts establishing and refining procedures for subjective assessments, simply because formal listening tests have been the only relevant method for judging audio quality in the past. As mentioned in Section 2.1, the experience gained was the foundation for Recommendation ITU-R BS.1116, which then became the basis for most listening tests of this type. This also defines the applicability of wideband audio tests like PEAQ. They can be applied wherever a subjective test according to BS.1116 would be applied. In addition to that newer research shows that PEAQ can be applied instead of MUSHRA tests as well, although this must be performed with special care. Wideband audio does not mean that speech can not be assessed. However it should be wideband speech in contrary to telephony bandwidth (300...3500Hz). Otherwise algorithms like PESQ or PSQM are more appropriate.

Since subjective quality assessments are both time-consuming and expensive, it was beneficial to develop an objective measurement method to produce an estimate of the audio quality. Traditional objective measurement methods, like Signal-to-Noise-Ratio (SNR) or Total-Harmonic-Distortion (THD) have never really been shown to relate reliably to the perceived audio quality. The problems become even more evident when the methods are applied on modern codecs that are both non-linear and non-stationary. After thorough verification, a model was recommended by the ITU-R as a measure for the perceived audio quality ("PEAQ") under recommendation BS.1387 in late 1998.

The basic concept for making objective measurements with the recommended method is illustrated in **Figure 5.1** below.

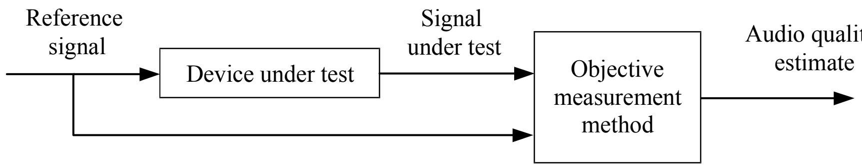


Figure 5.1: Basic concept for making objective measurements

The PEAQ measurement method is applicable to most types of audio signal processing equipment, both digital and analog. It is, however, expected that many applications will focus on audio codecs [ITUR1387].

5.2 Reference Files for Wideband Audio Measurements

OPERA Systems are delivered with a complete set of Test files. All WAVE files are stored in the folder c:\programme\opera\wavefiles. Besides the reference files used for performing life tests, there is a pair of reference and test file for each algorithm which are used for demonstrating file based measurements. These files are called AlgorithmRef.wav and AlgorithmTest.wav, where Algorithm must be replaced by PEAQ, PSQM, PESQ or Echo. PEAQRef.wav may also be used as a reference file for wideband audio measurements. The other AlgorithmRef.wav and AlgorithmTest.wav files must not be used for data acquisition. Additional files for wideband tests are shipped on the "PEAQ Test Sample CD". These files are recommended by the ITU for measurements using BS.1387/PEAQ.

5.3 Signal Acquisition

There are two different kinds of measurements in the case of the PEAQ algorithm, offline measurements and online measurements. As **Figure 5.2** demonstrates, there are two kinds of sources you can get your signals from. With all of the OPERA™ versions files can be assessed. The version with an included audio board, enables the performance of online measurements. In addition, a sound recorder software or OptiCall™ can be used as described in chapter 4 to obtain audio files.

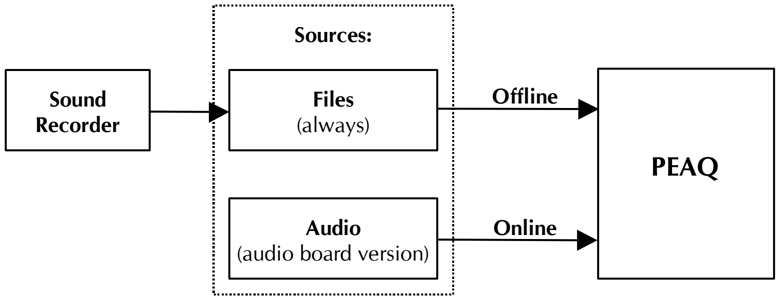


Figure 5.2: Kinds of signal sources for the PEAQ measurement

5.4 Fundamentals of the PEAQ Measurement Algorithm

As mentioned in paragraph 2.3, the ITU-R recommended an objective, perception based model to evaluate the quality of wide band audio codecs. This model was recommended as a measure for the perceived audio quality ("**PEAQ**") under recommendation BS.1387. There are two versions of **PEAQ**, a "Basic" version, featuring a low complexity approach, and an "Advanced" version for higher accuracy at the trade off of higher complexity. The following paragraphs provide some background information about **PEAQ** to improve the understanding of measurement results.

5.4.1 Background of the PEAQ (ITU-R BS.1387) Development

Not all members of the ITU task group TG10/4, which developed the new recommendation for the measurement of the perceived audio quality, were designing algorithms. On the contrary, most members represented potential users of such a system. The development therefore was influenced by the feedback of the users group with respect to their requirements. This feedback resulted in a list of applications, which the new method would have to face in a typical broadcast environment.

Basic and Advanced Version

By comparing both, the list of applications and the state-of-the-art technology, it became clear that not all requirements could be met by a single version of the algorithm. Most notably, a discrepancy exists between the demand for a real time measurement tool, highest possible accuracy and reasonable hardware efforts for an implementation. As a result, it was decided to develop a new method consisting of two versions. The "**Basic**" version was defined for computational efficiency and realtime performance, while the "**Advanced**" version yields for highest possible accuracy.

The following sections are intended to present a brief overview of the fundamental principles involved. For further details about the algorithm, please refer to the mentioned references.

5.4.2 Common Elements of PEAQ Basic and PEAQ Advanced

The structure of both versions is very similar, and fits exactly into the algorithm layout described in Chapter 4.1. The major difference between the Basic and the Advanced version is hidden in the respective ear models and the set of MOVs used. Both versions comprise an **artificial neural network** for the cognitive modelling. Since these networks are usually critical in terms of reliability, special care was taken not to over-train the network during the design phase. Subsequent investigations proved the stability and plausibility of the networks.

5.4.3 Basic version

The "Basic" version implements an FFT based ear model, as outlined in **Figure 5.3**.

Most features of this model are based on the fundamental psychoacoustic principles as described by Zwicker [ZWIC67, ZWIC82]. After the functional verification of the model, the developers tried to fine tune various parameters, but contrary to other existing proposals, the optimum was found very close to the theoretical values. Other models to-date always had to compensate for simplifications of the implementation by a misadjustment of other parameters.

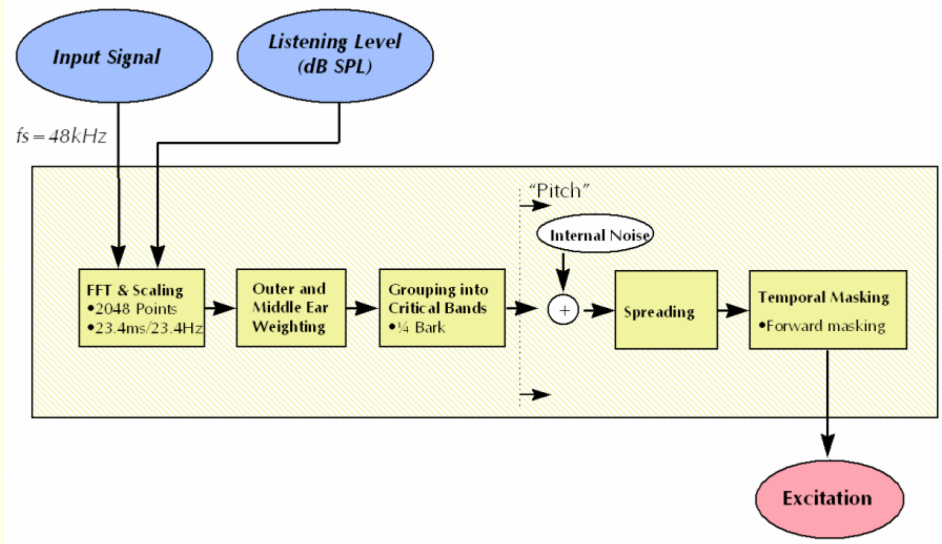


Figure 5.3: Perceptual model, PEAQ "Basic"

Following the signal flow from the input signal to the final calculation of the excitation pattern, the processing starts by a transformation of the input signal to the frequency domain. A 2048 point FFT is applied along with subsequent scaling of the spectra, according to the listening level, which has to be input by the user as a parameter. This results in a frequency resolution of approximately 23.4 Hz, and a corresponding temporal resolution of 23.4 ms (at 48 kHz sample rate).

In the consecutive block the effects of the outer and middle ear are modelled by weighting the spectrum with the appropriate filter functions. Afterwards the spectra are grouped into critical bands, achieving a resolution of 1/4 bark per band. The subsequent adding of "internal noise" is intended to model effects, such as the permanent masking of sounds in our auditory system caused by the streaming of blood and other physiological phenomena. This step is followed by the calculation of masking effects. Simultaneous masking is modelled by a frequency and level dependent spreading function according to [TERH79] with slight modifications. Temporal masking is modelled only partly since the temporal resolution is in the same range as the timing of any backward masking effects, which therefore can not be modelled. Nevertheless, experiments have shown that backward masking is very coarsely modelled by side effects of the FFT.

Using the feature extractor, eleven MOVs are extracted from the comparison of the Ear model output. **Table 5.1** shows a list of those MOVs and their interpretation. For further information about the MOVs please refer to the papers of the ITU-R recommendation BS.1387 in the appendix.

Note:

The lowered index "B" indicates an MOV of the ear model of the Basic version. A lowered "A" (see **Table 5.2**) indicates an MOV of the ear model of the Advanced version, respectively.

Model Output Variable (MOV)	Interpretation
$WinModDiff1_B$	Changes in modulation (related to roughness)
$AvgModDiff1_B$	
$AvgModDiff2_B$	
$RmsNoiseLoud_B$	Loudness of the distortion
$BandwidthRef_B$	Linear distortions (frequency response etc.)
$BandwidthTest_B$	
$RelDistFrames_B$	Frequency of audible distortions
$Total\ NMR_B$	Noise-to-mask ratio
$MFPD_B$	Detection probability
ADB_B	
EHS_B	Harmonic structure of the error

Table 5.1: MOVs used by the PEAQ "Basic" version, and their interpretation

5.4.4 Advanced Version

The "Advanced" version uses some MOVs derived by implementing the ear model of the "Basic" version but in addition to that, the "Advanced" version introduces a second ear model with improved temporal resolution, as illustrated in **Figure 5.4**.

Compared to the "Basic" version, the “Advanced” version model performs the time to frequency warping using a filter bank, consequently grouping the signal into 40 auditory bands with a temporal resolution of approximately 0.66 ms. This allows for a very accurate modelling of backward masking effects. After the calculation of backward and simultaneous masking, the signal is sub-sampled by a factor of 1:6 in order to improve the computational efficiency. Following the addition of the internal noise to the sub-sampled signal and finally modelling the forward masking effects, the output of this model is again the excitation.

In comparison to the FFT based "Basic" approach, the temporal resolution is improved, thus allowing for a better simulation of temporal effects, at the cost of frequency resolution and computational complexity.

The MOVs used by the "Advanced" version are listed in **Table 5.2**. It should be noted that due to the combination of parameters derived from both of the ear models, the number of MOVs required to derive the final quality measure could be reduced to five, while simultaneously the accuracy of the algorithm was slightly improved compared to the "Basic" version. For more detailed information about the Advanced Version, see the paper of the ITU-R recommendation BS.1387 located in the appendix.

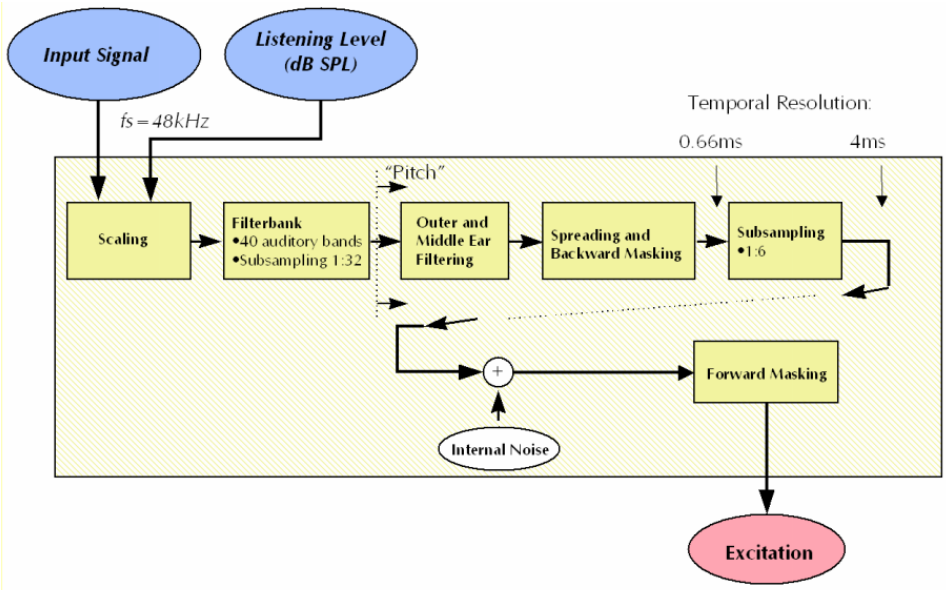


Figure 5.4: Perceptual model, PEAQ "Advanced" version

Model Output Variable (MOV)	Interpretation
RmsNoiseLoudAsym$_A$	Loudness of the distortion
RmsModDiff$_A$	Changes in modulation (related to roughness)
AvgLinDist$_A$	Linear distortions (frequency response etc.)
Segmental NMR$_B$	Noise-to-mask ratio
EHS$_B$	Harmonic structure of the error

Table 5.2: MOVs used by the PEAQ "Advanced" version, and their interpretation

5.5 Using PEAQ

This chapter will introduce the PEAQ measurement algorithm in general as well as the usage and application of PEAQ as implemented in OPERA™.

5.5.1 OPERA Software Suite - PEAQ

When using the OPERA Software Suite, there is no access to audio interfaces provided. Measurements are restricted to the comparison of files. Only audio files can be used as input signals. Supported file formats are **WAVE** files containing either plain PCM with **8 or 16bit per sample**, mono or stereo. The supported sample rates are **48kHz, as well as 44.1kHz**. Running PEAQ at 44.1kHz is not conforming to the standard and a proprietary extension of PEAQ.

As described in Chapter 4, select the files you want to assess during the first two steps of the measurement setup wizard.

5.5.2 OPERA Portable Tester with Audio Interface Option

OPERA™ systems including audio interfaces (OPR-110-EAQ-x) enables the performance of real time measurements as required in the case of on-line monitoring for example. The OPERA measurement systems can handle digital and analog audio interfaces integrated in the system. Refer to chapter 4 for details on the data acquisition using the sound board.

5.5.3 Algorithm Parameters

Figure 5.5 shows the property page with the settings that are specific to the PEAQ algorithm.

The **Listening level** is shown for informational purpose only. It is set fixed to 92dB SPL. This value is used to scale the binary representation of the audio data as they exist in the PC memory/harddisk to the real world in dB sound pressure level. Please refer to the ITU-R Rec. BS.1387 for more details.

The **Version** radio buttons allows switching between the PEAQ Basic and PEAQ Advanced. PEAQ Basic allows performing online measurements in realtime, while PEAQ Advanced gives a slightly higher correlation with subjective tests – at the cost of higher processing requirements. Online measurements are possible with PEAQ Basic only.

Several parameters which are helpful for online monitoring can be set in the **Result Logging** section. If **Result Logging Active** is checked, the system will reset all averaged results after the period given by the **Log Interval** parameter. If at this time the ODG is worse than indicated by **Log if ODG <=** and a **Log File** is given, the ODG will also be dumped into the log file. The resulting log file will look similar like the result files created in the batch mode.

With the parameters used in the example dialog, PEAQ Basic will be used and in the case of online measurements, the ODG will be reset every 6s. If the ODG is worse than –2 at that moment, the ODG will be printed into the file c:\temp\LogFile.txt.

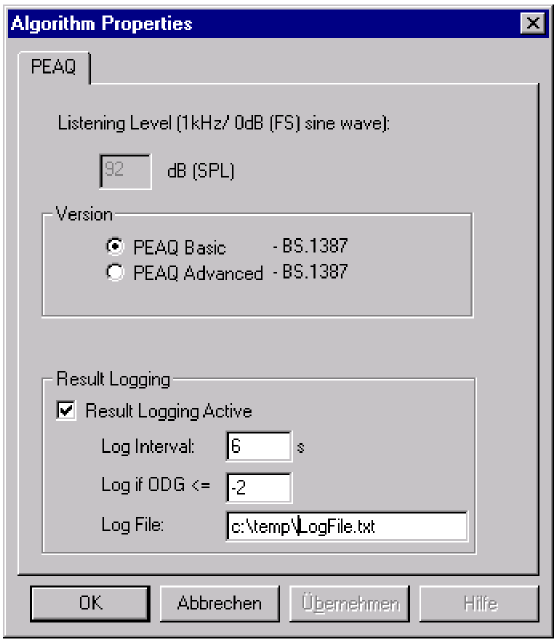


Figure 5.5: PEAQ algorithm specific properties

5.5.4 Diagram Types, PEAQ Basic

Chapter 4 showed how to select a measurement algorithm and how to start a measurement. Once the measurement is performed the results will be displayed. There are ten diagram types available for the PEAQ algorithm that will be described in this section (see **Figure 5.6**).

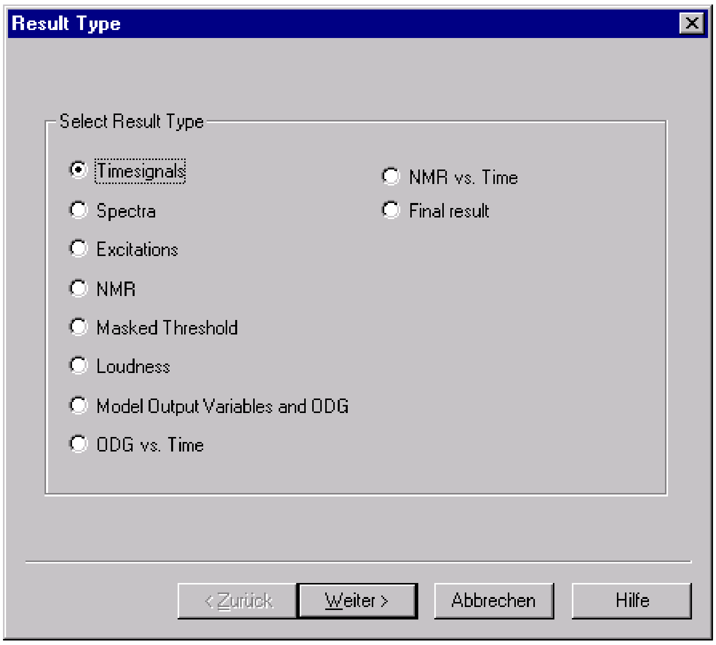


Figure 5.6: Result Type Window

Display of the Measurement Settings

First of all the information about the current measurement settings that are displayed to the right side of each diagram needs to be described (see **Figure 5.7**). This looks quite similar to the display of the Scope algorithm explained in

Chapter 4. However, there are some different parameters. **Table 5.3** explains the shown settings and their meaning.

Displayed Values	Interpretation
Time:	The time when the measurement has been finished.
F Smpl:	Sample rate of the input
BW Limit:	Highest frequency component taken into account
Level:	The current setting of the listening level
Tracking:	Status of the delay tracking function (on or off)
DC Filter:	Status of the DC filter (on or off)
Delay:	Delay in ms(first from top) as well as in samples (second from top)
Status:	Reliability of the automatic delay compensation (0..100%, Fixed = fixed delay set).
Atten:	Level difference between reference and test signal (dB)
Rel Time:	Current point of time in the measurement

Table 5.3: Interpretation of the displayed values

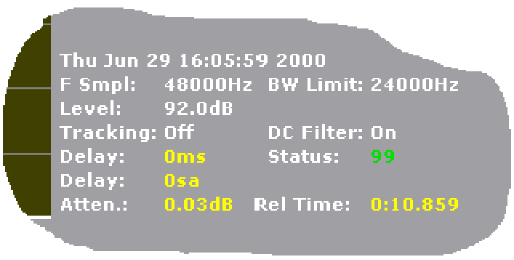


Figure 5.7: Display of the current measurement settings of the PEAQ algorithm

Timesignals

To choose this diagram type highlight the radio button next to **Timesignals**, and press **Next**. This leads to the next wizard step, the **Signal Select** dialog.

The "**Signal Select**" dialog (see **Figure 5.8**) defines for which channels and input signals the results in one diagram will be visible. Modify the selection by clicking with the left mouse button on any of the option buttons. This will add or remove the check mark in the button. A checked button means the results for the selected signal will be included in the diagram. In **Figure 5.8** the results for the left channel of the reference and the test signal were selected. Each signal will be drawn in a different colour. The assignment of the colours in the field to the right of the diagram panel will be displayed.

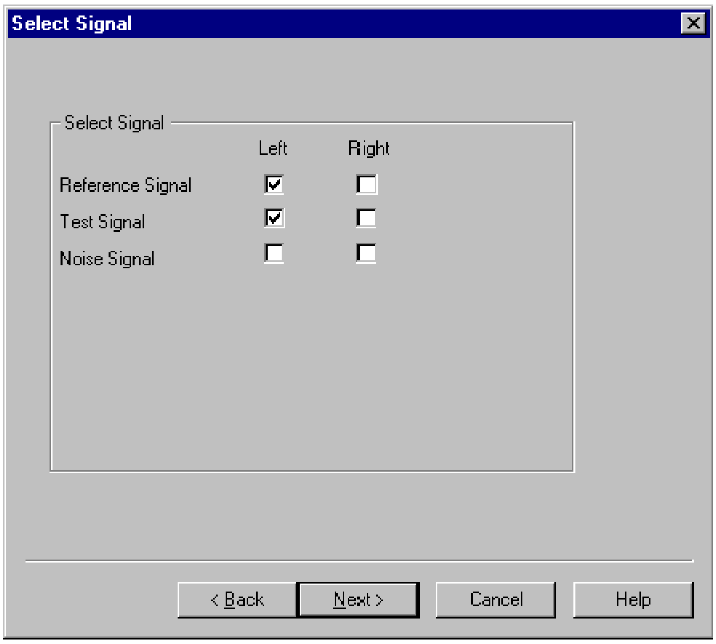


Figure 5.8: Select Signal Window

Pressing **Next** again leads to the next step, the **Result Style** dialog (see **Figure 5.9**). Here, select the way data is shown on the screen. Usually this is identical to selecting the units of the diagram axes. For the time signals choose between a binary, linear representation in which the input signals are always scaled to **[-32768 ... +32767]**, or a **dB FS** (full scale) scale.

Note:

Independently from the input data format, samples are always converted to **16bit/sample**. This means that 8bit/sample data are multiplied by 256 before they are processed any further.

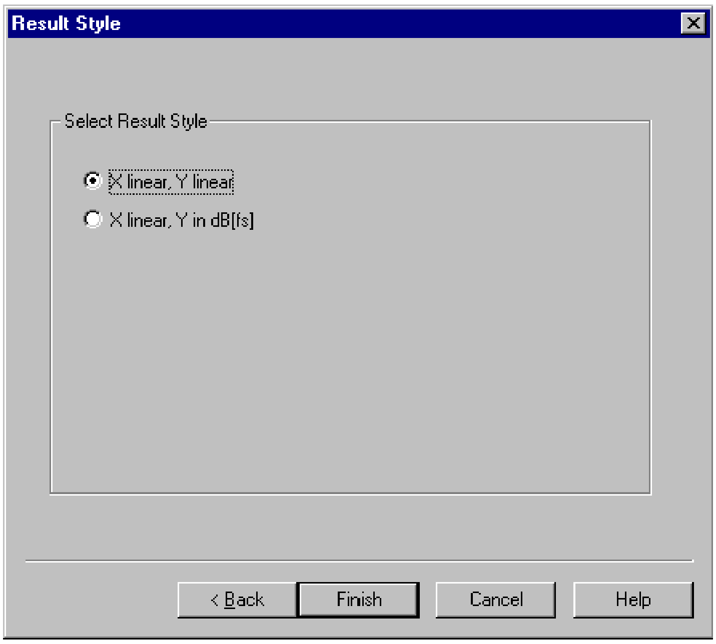


Figure 5.9: Result Style Window

After this last step click on **Finish** and the selected diagram will appear in the diagram pane as shown in **Figure 5.10**. An excerpt of the time signal of one frame will be shown.

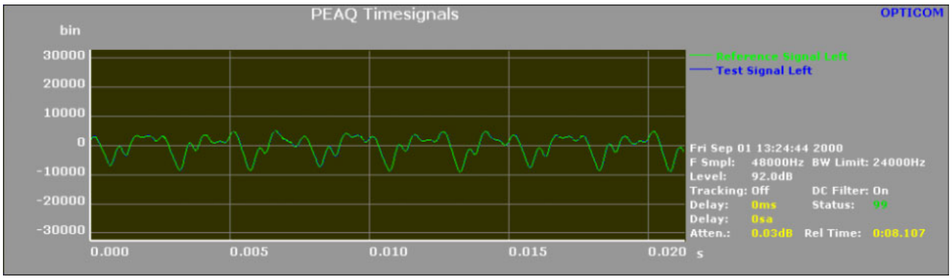


Figure 5.10: Time signals diagram

Spectra

Figure 5.11 shows the signal select wizard step for the spectra diagrams. Select the spectra of the reference signal, the test signal or the difference between the reference spectrum and the test spectrum (Noise).

Available result styles for the spectra (see **Figure 5.12**) are a linear frequency scale, a Bark scale or a ¼ Bark scale (as outlined in BS.1387). The Y-axis is always scaled in dB SPL. The Y-axis is depending on the setting of the listening level as set by the algorithm properties dialog. The spectral resolution is depending on the sample rate of the input signals. Usually a 2048 point FFT is used with a Hann window to compute these data (as required by BS.1387).

The resulting diagram will look similar to the one shown in **Figure 5.13**.

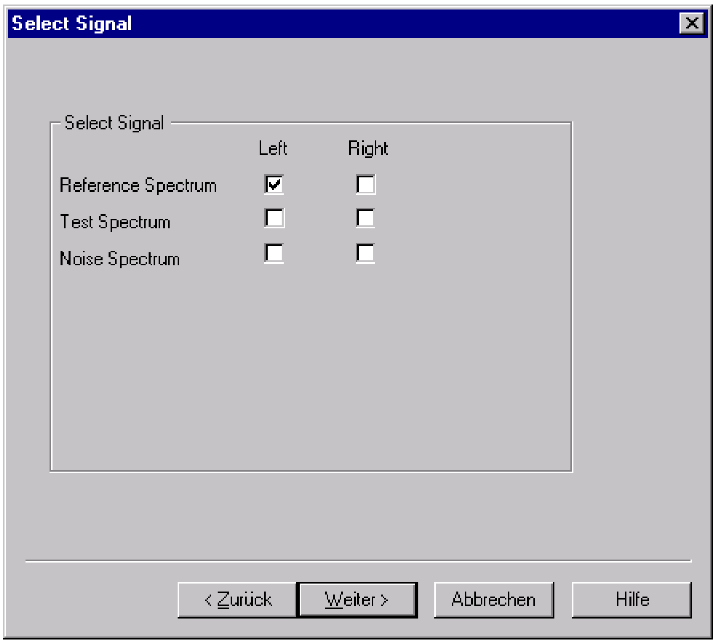


Figure 5.11: Select Signal Dialog

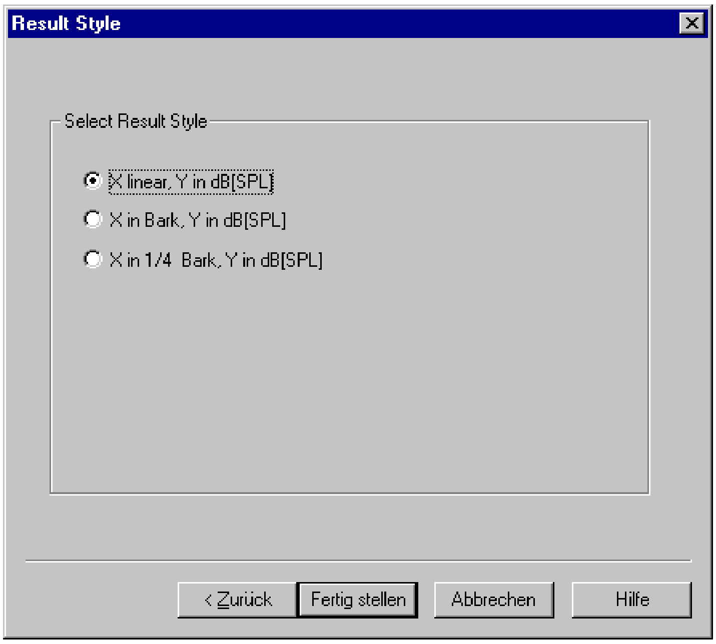


Figure 5.12: Result Style Dialog

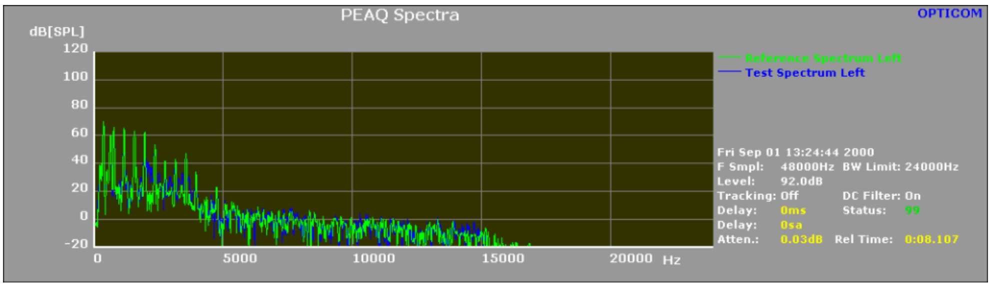


Figure 5.13: The Spectra diagram

Excitation
The excitation diagram displays the internal representation (see Section 2.2) of a signal.

The Bark scale is used for the horizontal axis, the vertical axis is scaled in dB SPL. The curves are shown on a frame by frame basis, without any averaging. As shown in Figure 5.14, the reference signal excitation, the test signal excitation and the difference between both excitations can be selected. The difference is calculated by subtraction in the linear domain and displayed in dB SPL.

Click on “Next” to go to the dialog shown in Figure 5.15. Here choose between different scalings of the X-axis. Currently available are scalings in Herz as well as in Bark.

Figure 5.16 finally shows the resulting view of the Excitation diagram.

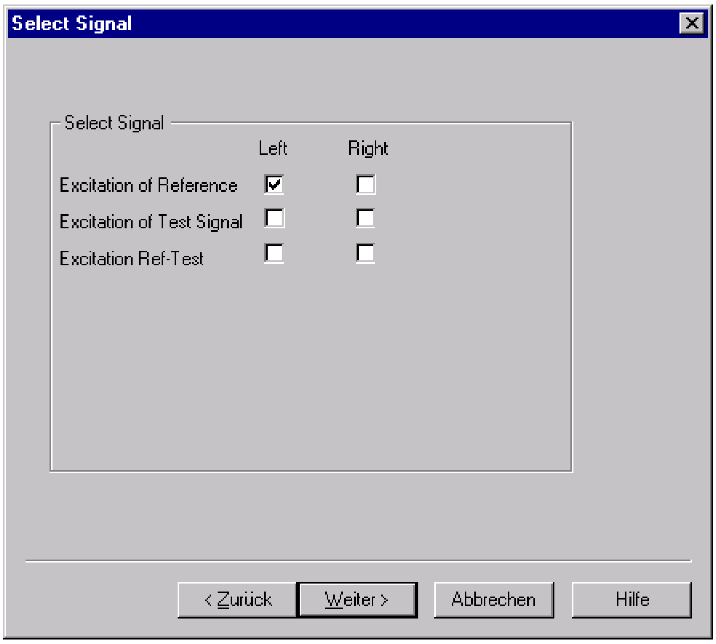


Figure 5.14: Select Signal Window for the excitation diagram type

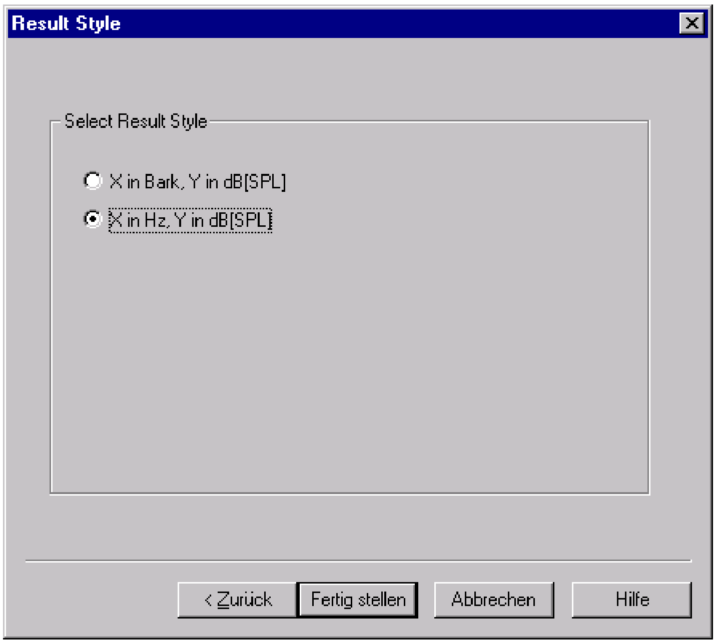


Figure 5.15: Excitation result styles

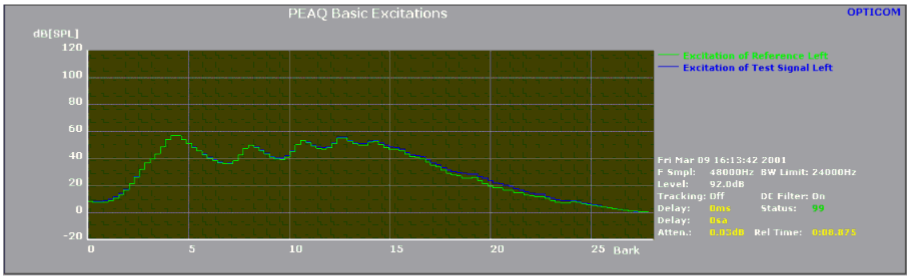


Figure 5.16: The Excitations diagram

NMR

The measurement scheme NMR (Noise-to-Mask Ratio) [BRAN87] evaluates the level-difference between the masked threshold (the maximum level of a not audible error) and the actual noise (error) signal. NMR is defined as the ratio between the error signal to the masked threshold. The masked threshold is estimated for each band of the Bark scale.

Negative values of the NMR provide an estimation of the existing safety margin. Positive values signify an estimation of the audible noise energy. **Figure 5.17** shows the select signal window for the NMR diagram type. The resulting diagram is shown in **Figure 5.18**.

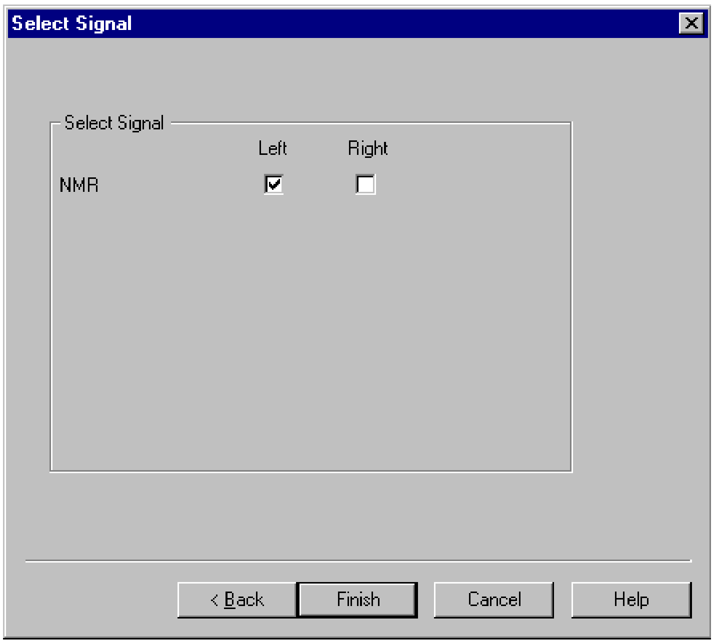


Figure 5.17: Select Signal window of the NMR diagram type

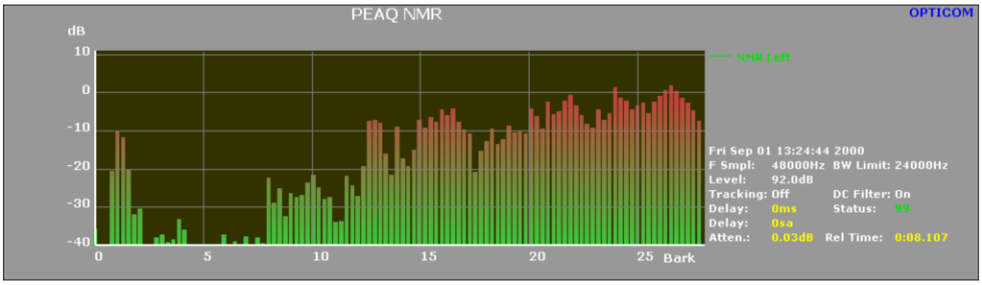


Figure 5.18: The NMR diagram

Masked Threshold

A signal that is clearly audible can be completely inaudible in the presence of another signal, the masker. This effect is called masking and the masked signal is called maskee. There are two situations that have to be distinguished - simultaneous masking and temporal masking. In case of simultaneous masking the masker and the maskee are present at the same time and are quasi-stationary. In case of temporal masking the masker and the maskee are present at different times [ITUR1387]. Both situations are taken into consideration in the masked threshold diagram.

As **Figure 5.19** shows, choose between the masked threshold, the reference spectrum and the noise spectrum (spectrum reference – spectrum test) of the right and left channel. The horizontal axis of the diagram is a Bark scale, the vertical axis is scaled in dB[SPL] (see **Figure 5.20**).

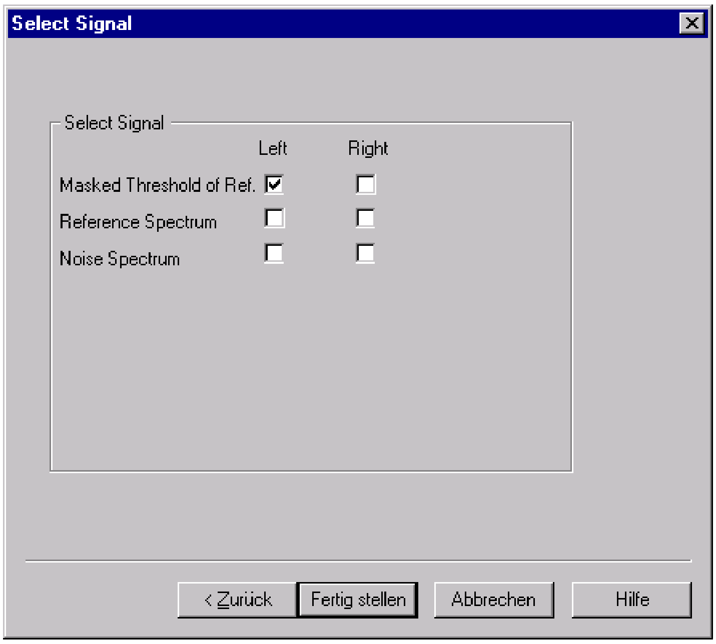


Figure 5.19: Signal Select window of the Masked Threshold

Figure 5.20: The Masked Threshold diagram

Loudness

The perceived loudness of audio signals depends on their frequency, their duration and their sound pressure level. Due to auto-masking the loudness of a complex signal is less than the sum of the loudness of all its components. In the context of audio quality measurement, the loudness of the unwanted distortion added to the reference signal, the noise loudness, is reduced by the partial masking caused by the reference signal [ITUR1387].

A choice can be made between the loudness of the reference signal and the loudness of the test signal (see **Figure 5.21**). As **Figure 5.22**Figure 5.22 shows, the scaling of the vertical axis is Sone, the horizontal axis may be scaled in Bark or in Herz. The resulting view looks as **Figure 5.23**

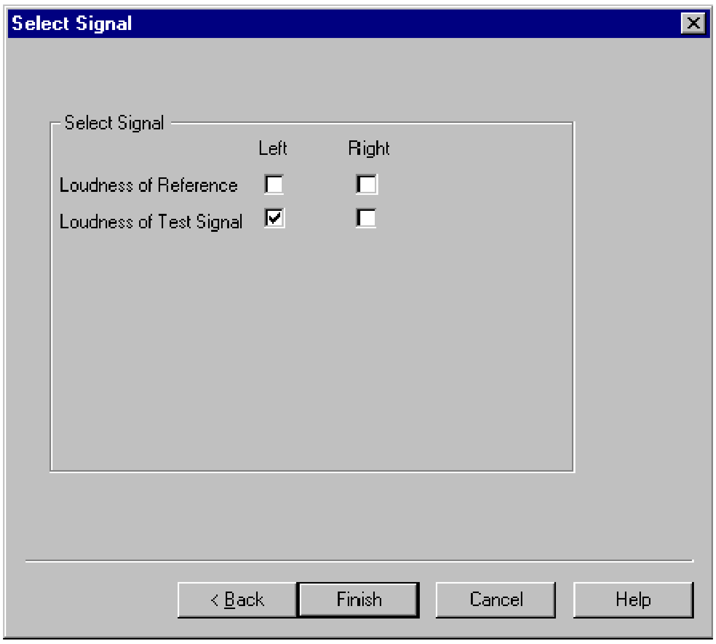


Figure 5.21: Select Signal window of the loudness diagram type

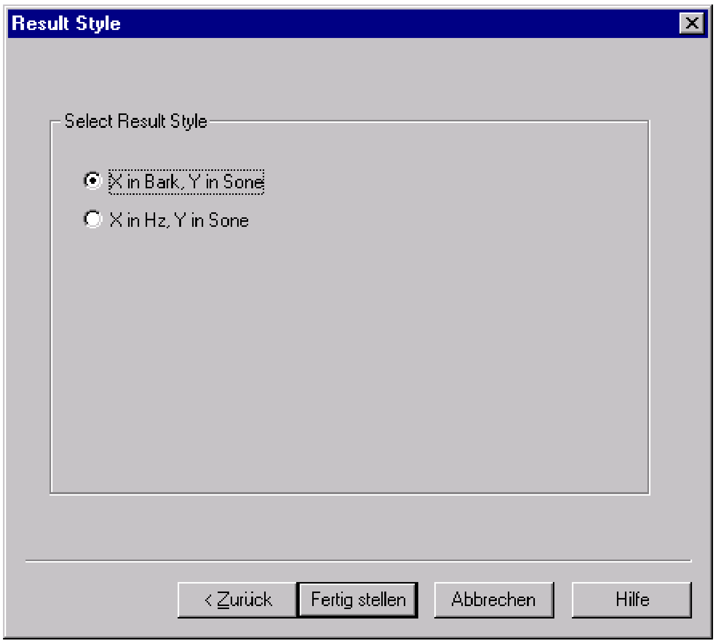


Figure 5.22: The Loudness result styles

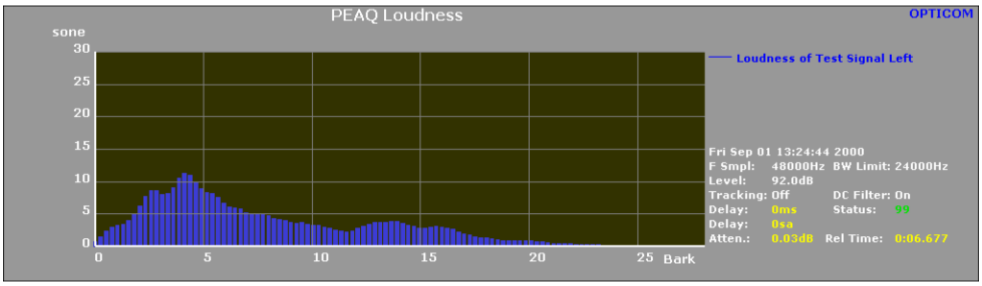


Figure 5.23: The Loudness diagram

Model Output Variables and ODG

The screen shot in **Figure 5.24** shows the Model Output Variables (MOVs) as they are defined by BS.1387. For a detailed explanation of these please refer to the ITU-R recommendation in the Appendix of this manual. The results are shown framewise and are averaged since the beginning of the measurement. This diagram contains several bars, each having a different meaning, scaling and unit. The unit - if available - as well as the current value are shown on top of the bars. The percent scale on the left side of the diagram is for orientation merely. The first 11 bars (AvgBwRef .. NLoudB) represent the MOVs according to BS.1387.

Table 5.4 may help relating the OPERA™ names to the according names of the ITU recommendation.

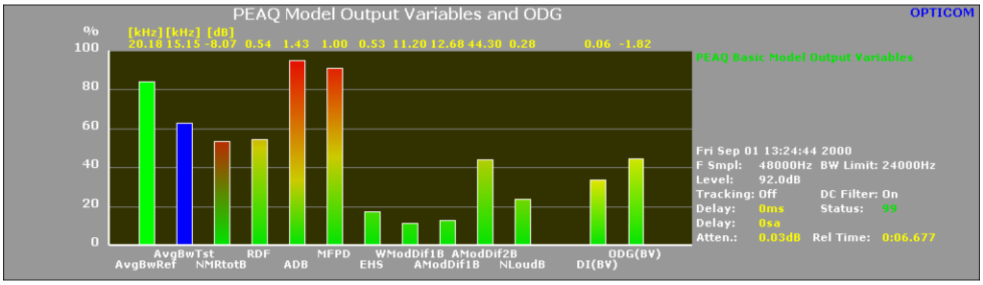


Figure 5.24: PEAQ Basic Model Output Variables (MOVs) and the ODG

OPERA™ name	BS.1387 name
AvgBwRef	$BandwidthRef_B$
AvgBwTst	$BandwidthTest_B$
NMRtotB	Total NMR_B
ADB	ADB_B
MFPD	$MFPD_B$
EHS	EHS_B
RDF	$RelDistFrames_B$
WModDif1B	$WinModDiff_B$
AModDif1B	$AvgModDiff1_B$
AModDif2B	$AvgModDiff2_B$
NLoudB	$RmsNoiseLoud_B$

Table 5.4: PEAQ Model Output Variables with respect to BS.1387 definitions

Objective Difference Grade (ODG)

The last two bars in the diagram shown in **Figure 5.24** are the Distortion Index (DI) and the final Objective Difference Grade (ODG). The "BV" in the brackets indicates that this value is a result of the Basic Version of the PEAQ algorithm. The ODG is the output value from the objective measurement method that corresponds to the SDG (see Section 2.1) in the subjective domain. The resolution of the ODG is limited to one decimal. However, be cautious and do not generally expect that a difference between any pair of ODGs of a tenth of a grade is significant. The same remark is valid when looking at results from a subjective listening test. As the right diagram shown in **Figure 5.25**, the ODG can also show positive values. Such values can occur since OPERA™ uses the cognitive model to map the MOVs to the results of subjective listening tests. In the case of subjective listening tests, the SDG can assume a positive value, when a test person has incorrectly assigned the reference and test signal.

Distortion Index (DI)

The Distortion Index (DI) has the same meaning as the ODG. However, DI and ODG can only be compared quantitatively, but not qualitatively. **Figure 5.25** shows two curves that represent the relation between the quality and the DI value (left diagram) and the relation between the quality and the ODG value (right diagram). As the left diagram demonstrates, the DI is characterized by a saturation that is less than the saturation of the ODG curve. Furthermore, the range of values is different. As a general rule, use the ODG as the quality measure for ODG values greater than approx. –3.6. The ODG correlates very well with subjective assessments in this range. When the ODG value is less than –3.6 use the DI.

> **Note:**
>
> Never compare the ODG value of one measurement with the DI value of another.

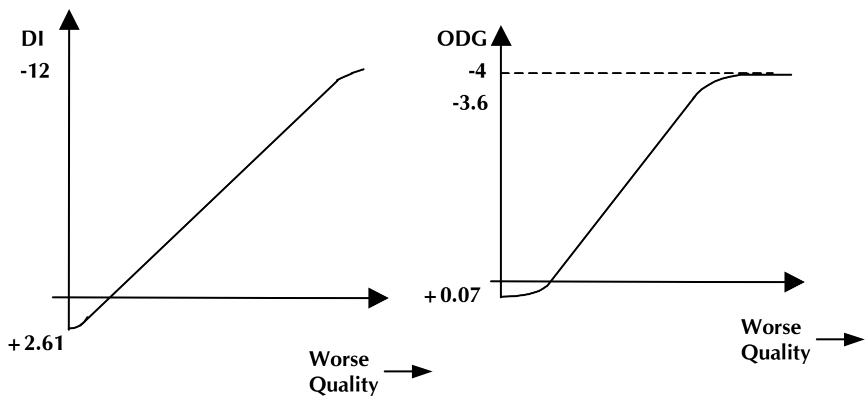


Figure 5.25: Comparison of the DI and the ODG. Left diagram DI, right diagram ODG

ODG vs. Time

The ODG value in **Figure 5.25** an averaged value. Choose the diagram type "ODG vs. Time" to see the ODG value vs. time. The resulting diagram is shown in **Figure 5.26**. The pink vertical line in the diagram indicates the current position of the time slider inside the signal.

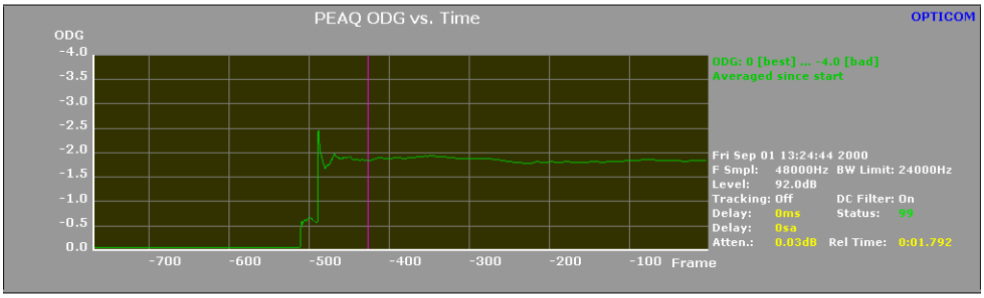


Figure 5.26: The ODG vs. Time diagram

NMR vs. Time

To see the run of the NMR value vs. time choose the diagram type "NMR vs. Time". Select between the NMR value vs. time of the right and / or the left channel (see **Figure 5.27**). The resulting diagram will look similar to the one shown in **Figure 5.28**. As in the ODG vs. Time diagram, the pink line indicates the current position of the time slider.

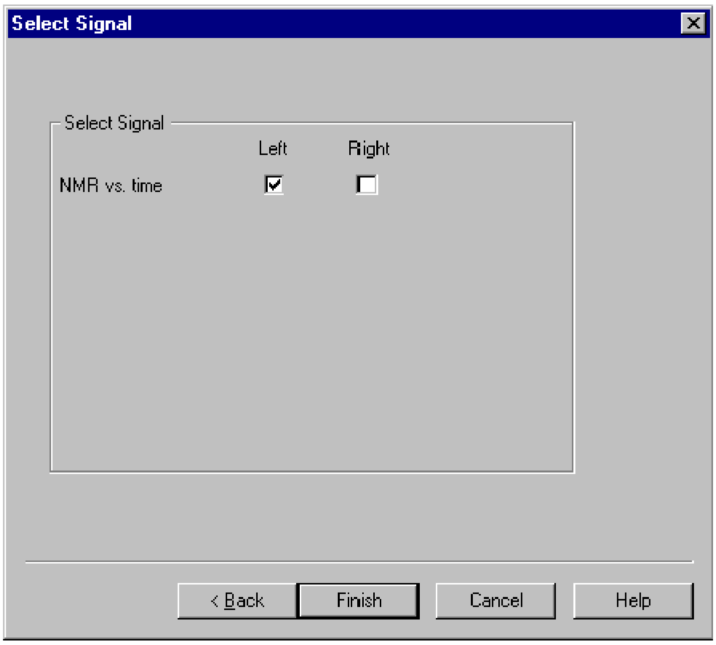


Figure 5.27: The Select Signal dialog of NMR vs. Time

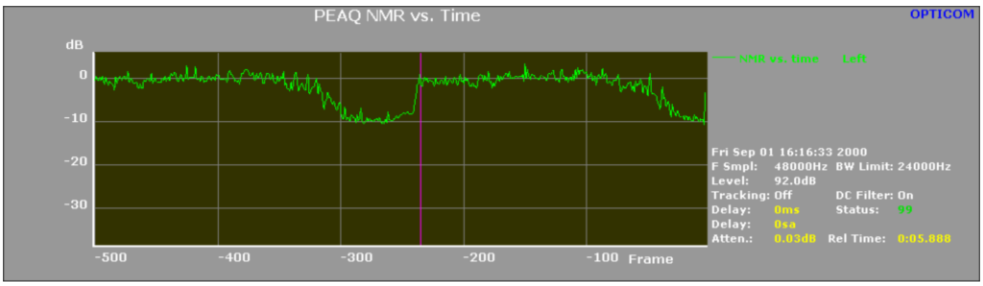


Figure 5.28: The NMR vs. Time diagram

PEAQ Basic Final Results

Figure 5.29 shows the final results diagram for the Basic version of PEAQ. This screen is a summary of the results obtained for the entire measurement sequence, from the start point to the end point. The criterion for the start and the end point is given by BS.1387. The start and stop point detection ensures that measurement results are not falsified by silent periods at the beginning or at the end of the files. The values shown in this diagram will not vary while moving through the history buffer. These are also the values that should be reported as the result of a measurement. Please note that the "Model Output Variables and ODG" diagram may contain slightly different values for the ODG and DI than the "Final Result" diagram, since only the latter takes the stop point of the measurement into account, while the first one is measuring until it encounters the end of any of the two input streams.

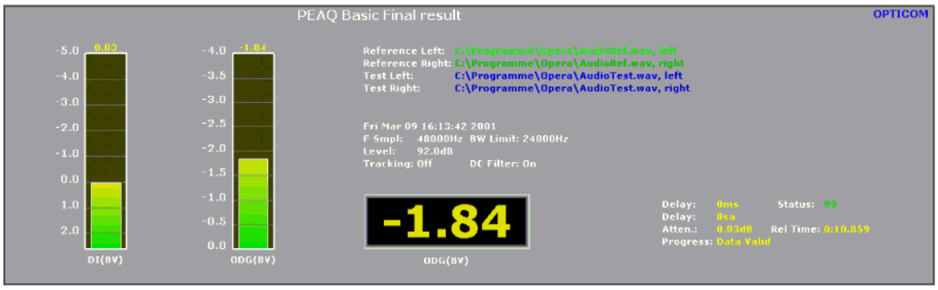


Figure 5.29: PEAQ Basic, final results

5.5.5 Diagram Types, PEAQ Advanced

Figure 5.30 shows the various diagram types available for the advanced version of PEAQ. Most of the diagrams look exactly like the according diagrams of the basic version, with the exception of the scaling of the time axis. The following paragraphs will describe only those diagrams that differ significantly from those of the basic version.

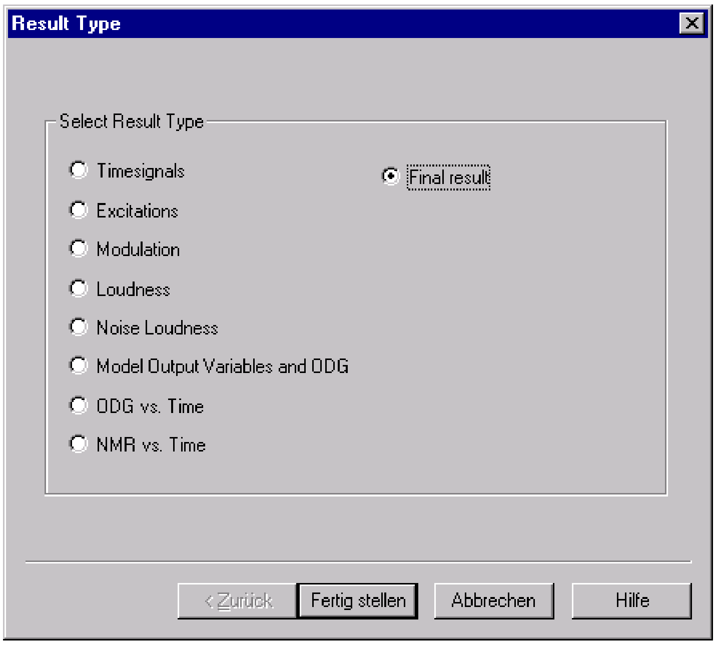


Figure 5.30: PEAQ Advanced, diagram types

Modulation

The Modulation diagram shows the modulation of the reference and the test signal over a bark scale. The Modulation is a number without units.

Noise Loudness

The noise loudness shown here is not comparable to the Zwicker loudness available in the basic version. Here see the "partial loudness of additive distortions in the presence of the masking reference signal", as described in BS.1387. For details please look at the ITU recommendation in the Annex of this manual.

Model Output Variables and ODG

The screen shot in **Figure 5.31** shows the Model Output Variables (MOVs) as they are defined by BS.1387. For a detailed explanation refer to the ITU-R recommendation in the Appendix of this manual. The results are shown framewise and are averaged since the beginning of the measurement. This diagram contains several bars, each having a different meaning, scaling and unit. The unit - if available - as well as the current value are shown on top of the bars. The percent scale on the left side of the diagram is for orientation merely . The first 5 bars (RModDif .. EHS) represent the MOVs according to BS.1387.

Table 5.5 may help relating the OPERA™ names to the according names of the ITU recommendation.

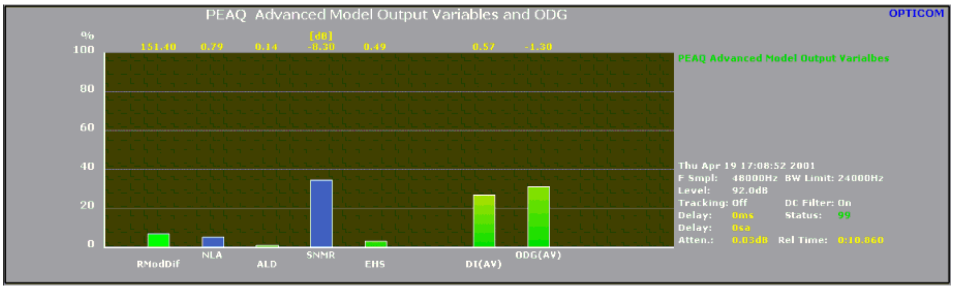


Figure 5.31: PEAQ Advanced Model Output Variables (MOVs) and the ODG

OPERA™ name	BS.1387 name
RModDif	$RmsModDiff_A$
NLA	$RmsNoiseLoudAsym_A$
ALD	$AvgLinDist_A$
SNMR	Segmental NMR_B
EHS	EHS_B

Table 5.5: PEAQ Model Output Variables with respect to BS.1387 definitions

Objective Difference Grade (ODG) and DI

The last two bars in the diagram shown in **Figure 5.31** are the Distortion Index (DI) and the final Objective Difference Grade (ODG). The "AV" in the brackets indicates that this value is a result of the Advanced Version of the PEAQ algorithm.

Note:

Never compare the ODG value of one measurement with the DI value of another.

5.5.6 Command Line Arguments

The current PEAQ implementation provides several algorithm specific command line parameters to:

- Set the version of the algorithm (basic, advanced)
- Set the listening level
- Set the logging of the results

These parameters essentially follow the settings of the algorithm parameter dialog and are listed with a short comment on their usage in the following:

Keyword	Add. Parameter	Comment
Version	0 = Basic; 1 = Advanced	Select the Version of the Algorithm
Level	Listening Level of a 1kHz 0dBfs sine tone	Set the listening level according to BS.1387
LogActive		Switch logging on
LogODG	float	Logging if ODG < = float
LogInterval	duration	Logging intervals in s
LogFileName	FileName	Name of the logfile

5.6 Example Measurement Setups

To an inexperienced user of the OPERA™ measurement system, the following examples might be useful to assist in the first measurements. The first example demonstrates a typical real time measurement, Example 2 deals with a file based measurement and finally you will get to know a typical application of measurements from a batch file with Example 3.

5.6.1 Example 1: Online Monitoring

Assuming we have the reference and the test signal needing to be assessed in a digital format, e.g. 48 kHz sampling rate, 16 bit resolution and stereo signals. We connect the reference signal with the **AES-EBU Input 1** and the test signal with the **AES-EBU Input 2**.

To properly configure your interfaces refer to Chapter 4, where this process is explained in detail.

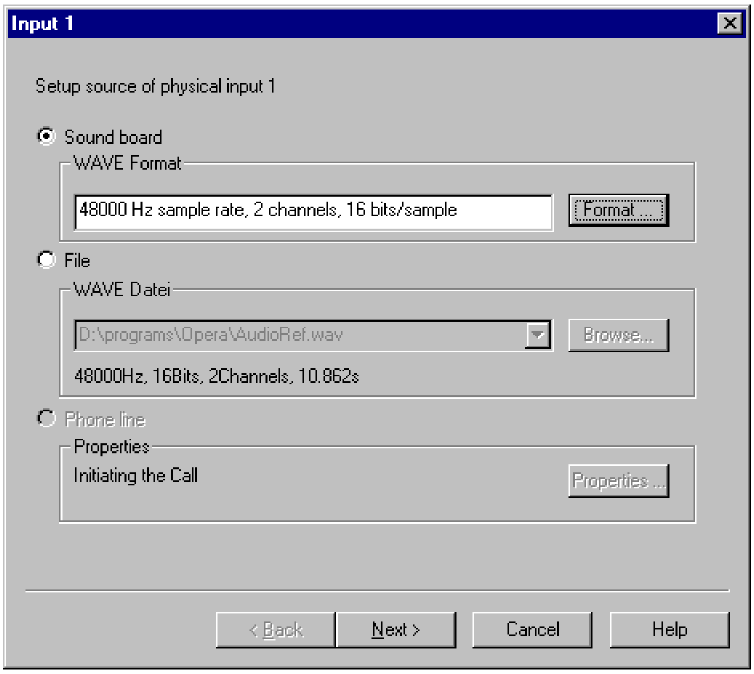


Figure 5.32: The Input 1 dialog step of the Measurement Setup Wizard

Then we choose from the OPERA™ framework the menu option **Measurement|Algorithm Parameters ...** the PEAQ algorithm and use the "Algorithm Properties" button to select the basic version of PEAQ. After doing so, click on the **OK** button and start the measurement.

In the first and second steps of the Measurement Setup Wizard, select the radio button **Sound Board** and – if necessary - change the format settings to the ones shown in **Figure 5.32**, by using the "Format" button. On systems equipped with LynxONE boards, the format settings are also used to switch between analog and digital inputs.

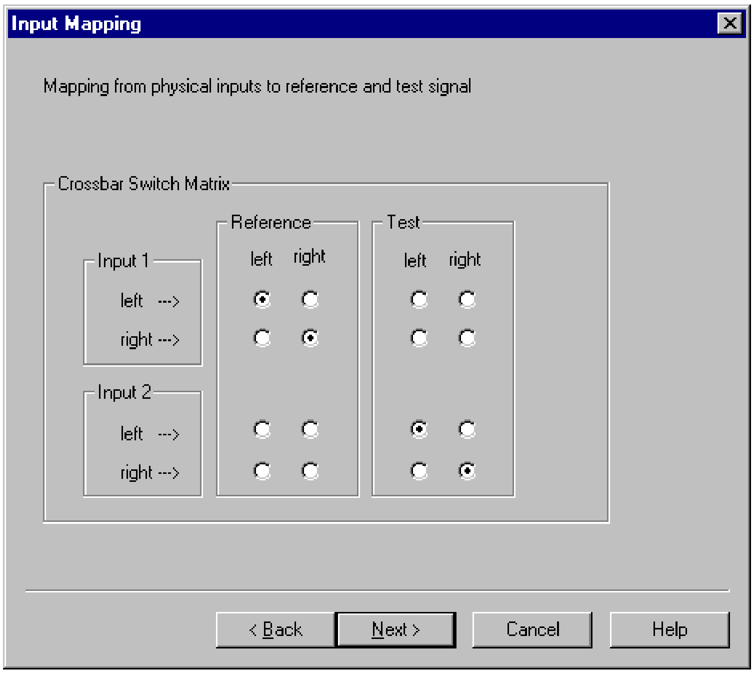


Figure 5.33: Input Mapping Wizard step

Figure 5.33 shows the correct settings of the **Input Mapping** dialog. Before actually observing the results of our measurement, it is necessary to choose several options for the preprocessing of the signals. As **Figure 5.35** shows, the **Automatic Delay Compensation** in **Normal** mode on the **Left** channels was selected. Please recall: the Normal and the Snap Mode are not enabled when a file-based measurement is performed.

In addition to this we switch on the functions **Static Gain Compensation** and **Remove DC From Signals**. Now, start the measurement by clicking on the **Finish** button.

When choosing the diagram type **Timesignals** for the upper diagram and **Model Output Variables and ODG**, the framework will look as shown in **Figure 5.36**, of course depending on the signals that are used for the measurement.

The more time elapses the more the result values in the lower diagram will stabilize. When pressing the **Reset Averaged Values** button shown in **Figure 5.34**, observe how the result values are re-established.

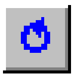


Figure 5.34: Reset Averaged Values toolbar button

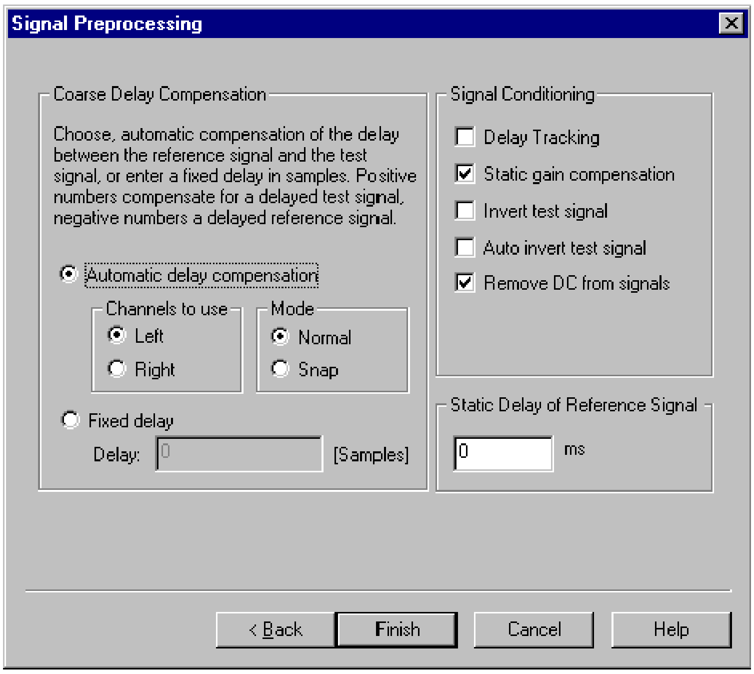


Figure 5.35: The Signal Preprocessing dialog of the Measurement Setup Wizard

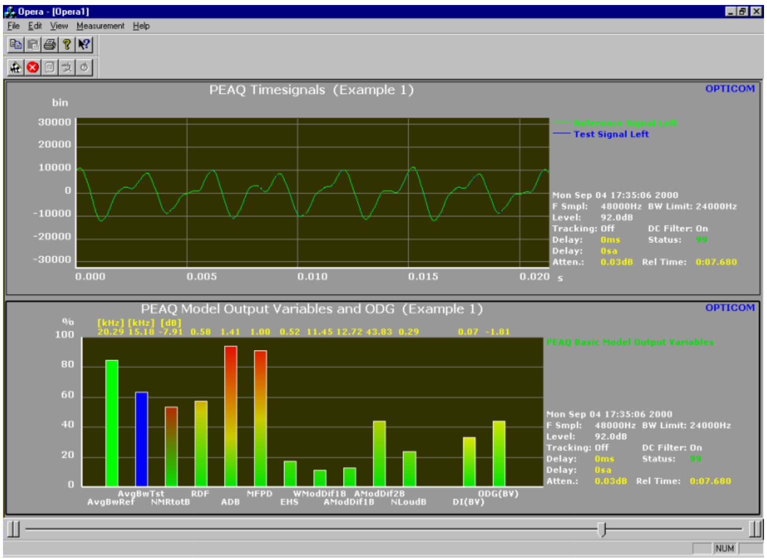


Figure 5.36: View on a real time measurement result

As described in Chapter 4, the delay between the reference and the test signal is checked permanently in **Normal** Mode. In some measurement cases notice that the displayed value for the detected delay in the field to the right of the diagram panes changes. When stopping the permanent delay detection while keeping a certain delay value, choose the **Freeze Delay** option by pressing the toolbar button shown in **Figure 5.37** or the corresponding menu option from the menu topic **Measurement**. After doing so, the status display to the right of the diagrams will indicate this mode by displaying "Frozen".

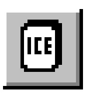


Figure 5.37: Freeze Delay toolbar button

The **Snap Again** toolbar button to the right of the **Freeze Delay** button is disabled at the moment since our measurement is running in the **Normal** mode of the Automatic Delay Compensation. At any time the running measurement can be discontinued and a different measurement settings like the **Snap** mode can be selected, for example, before starting a new measurement.

5.6.2 Example 2: Stand Alone Testing

In this example an analysis will be performed of a .wav file, which could be a decoded mp3 signal, for instance.

First of all, enter a name for our measurement. Select from **Measurement|Name Measurement...** and enter the name "Example 2". Now select from the menu option **Measurement|Algorithm Parameters...** the algorithm **PEAQ**. Take notice of the following warning message and click on the **Continue** button.

Now commence the measurement. Press the **Start toolbar button** shown in **Figure 5.38**.

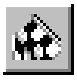


Figure 5.38: Toolbar button for starting the measurement

Now the Measurement Setup Wizard starts and the window to select the first input signal is displayed (see **Figure 5.39**). Select the **radio button File** and choose "AudioRef.wav" as your first signal. Underneath the text edit box information about the signal is displayed: The sampling rate, the bit resolution, the number of channels and the total time length.

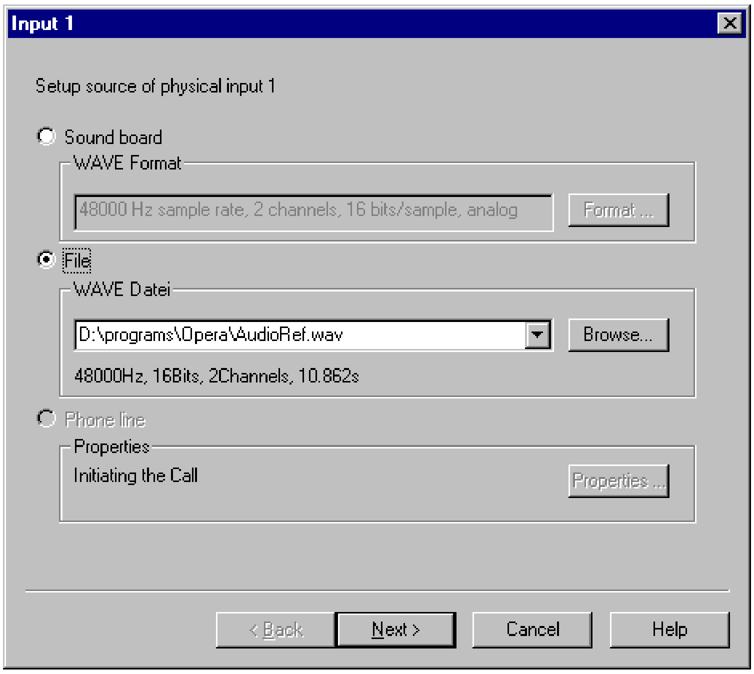


Figure 5.39: Input 1 window of the Measurement Setup Wizard

In the next window of the wizard chose the file for the second input, "AudioTest.wav". After this determine that Input 1 shall be used as the reference signal and the signal of Input 2 as the test signal. This is done in the mapping window shown in **Figure 5.40**.

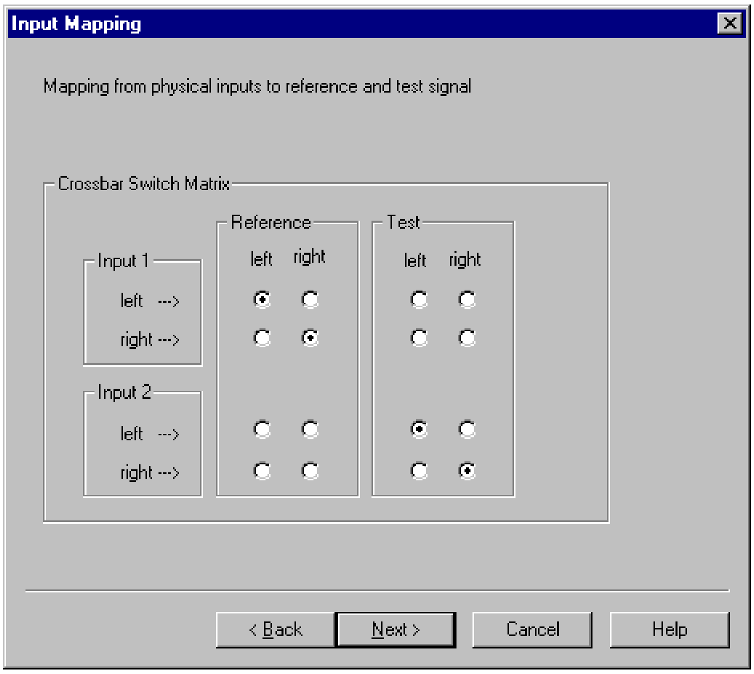


Figure 5.40: Input Mapping window of the Measurement Setup Wizard

The last window of the Measurement Setup Wizard before the actual measurement starts, is the Signal Preprocessing Window shown in **Figure 5.41**. If there is a delay between the reference and the test signal, it should be compensated automatically. Therefore choose the **automatic delay compensation** function. To compensate any difference in the static gain between both signals choose the option **Static gain compensation** and, finally, the option **Remove DC from signals**.

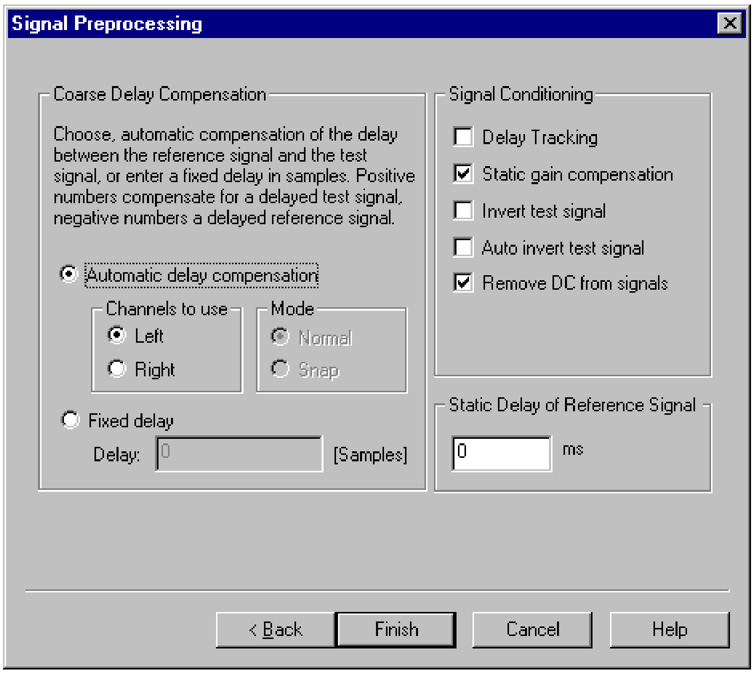


Figure 5.41: Signal Preprocessing window of the Measurement Setup Wizard

After pressing the **Finish** button, the delay is computed, which might take some seconds. Finally one can choose which result values to display. By a right mouse button click in the upper diagram choose the type **Model Output Variables and ODG**, for the lower diagram, select the diagram type **ODG vs. Time** that displays the averaged ODG since the start of the measurement. The resulting view of the Main Window is shown in **Figure 5.42**. The upper diagram shows the averaged ODG. When compared with the SDG scale depicted in Section 2.1, observe that the ODG value of –1.80 in this example corresponds to slightly annoying impairments.

The display of the current measurement settings to the right of each diagram panel contains the parameter "**Status**" which is a measure of the reliability of the delay compensation. For measurements with the PEAQ algorithm this value should not fall below approx. 80%. When the status value is less it could be useful to check what the time signals of the reference and the test signals look like. Change to the diagram type **Timesignals** to check the gain and the delay compensation.

When interested in more detailed information about your measurement, you might want to check the MOVs. Please refer to the ITU-R recommendation BS.1387 in the Appendix of this manual for detailed explanation.

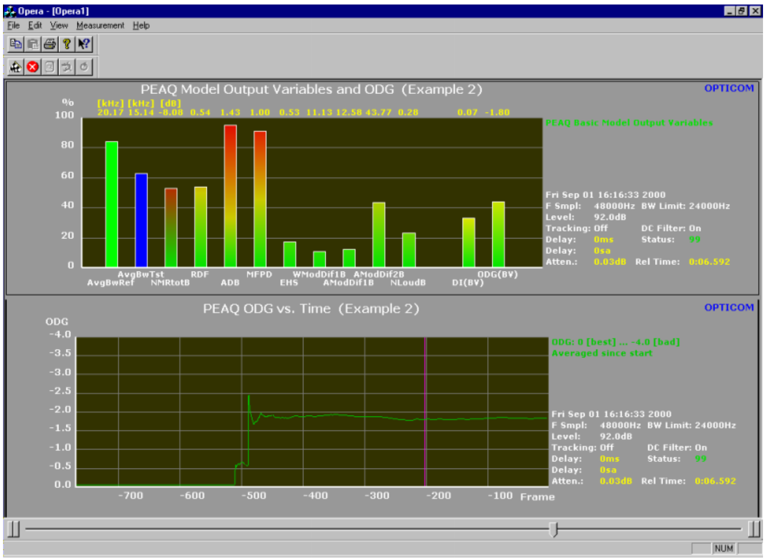


Figure 5.42: Main Window with the view of the measurement results

5.6.3 Example 3: Measurements From a Batch File

For this example use the batch files contained on the **PEAQ Sample CD** which were received along with your OPERA™ system. On your PEAQ Sample CD you find three batch files in the root:

- ConfPeaqBasic.bat
- ConfPeaqAdvanced.bat
- RunPeaqBasic.bat
- RunPeaqAdvanced.bat

In the following we will replace Basic and Advanced by XXX. These batch files work together and perform several PEAQ measurements with different test and reference signals. **ConfPeaqXXX.bat** calls **RunPeaqXXX.bat** several times, each time measuring different wave files contained on the PEAQ Sample CD.

The wave files used in the batch file **ConfPeaqXXX.bat** are taken from the **DB3** database that was used in the ITU-R BS.1387 standard to verify the PEAQ algorithm. **DB3** contains several reference and test files and their corresponding SDG values from the subjective listening tests.

A result file **ConformanceReferenz.txt** can be found in the root of your PEAQ Sample CD. This file contains the result values of the PEAQ Basic measurements for the DB3 database used by the ITU-R to verify PEAQ. The contained values are tab-separated and can thus be imported into any spreadsheet analysis program. Use this result file to verify that the delivered OPERA™ system is working correctly.

The following will explain the mentioned batch files and explain how to save result values in your own result file.

The following lines are the contents of the file **RunPeaqBasic.bat**. Please notice that the commands after the keyword **Opera -Exec** have to be written in one line in the batch file. Here, the line has to be wrapped in the absence of space.

RunPeaqBasic.bat

```
@echo on
rem batch file to compute PEAQ for a pair of files from the
rem   ITU DB3
rem This batch assumes, that OPERA is installed on drive C !!
rem Parameters:
rem
rem  RunPeaqBasic   <OperaPath> <File1> <File2><Outputfile>
rem
rem         OperaPath:  Drive and path of the OPERA
rem                     installation directory
rem
rem         File1:   File that contains the reference signal
rem
rem         File2:   File that contains the test signal.
rem
rem         Outputfile: Results are stored in this file.
rem                     If it exists already results are
rem                     appended to it, otherwise it will be
rem                     newly created.
rem
rem
echo ***************************************************
echo ***** RunPeaqBasic  V1.0  (c) OPTICOM, 1999   ***
echo ***************************************************
echo.

pushd
c:
cd %1

Opera -Exec -Algorithm Name=PEAQ -Input Inp=0 File=%2 Inp=1
File=%3 -Mux InpRefLeft=0 ChannelRefLeft=0 InpRefRight=0
ChannelRefRight=1 InpTestLeft=1 ChannelTestLeft=0
InpTestRight=1 ChannelTestRight=1 -Delay FixedDelay Delay 0 -
Out %4 -Append

e:
popd
```

In this file all the settings for the measurement are defined. The path and filename of the reference, the test file and of the result file are passed to the file by parameters.

Use pushd and popd in a batch program to return to the directory where the batch program was started. See the MS Windows help for details.

`RunPeaqBasic.bat` is called from the file **`ConfPeaqBasic.bat`** where all the files from the DB3 are entered. See the following lines that show the content of **`ConfPeaqBasic.bat`**. Note that the parameters that follow the command "call" must be written into one line. Again, those lines had to be wrapped in the absence of space in this manual.

ConfPeaqBasic.bat

```
@echo off
rem batch file to compute PEAQ for the DB3. OPERA must be
rem installed on drive C!
rem Parameters:
rem
rem   ConfPeaqBasic   <OperaPath> <DB3Path> <Outputfile>
rem
rem   OperaPath:  Drive and path of the OPERA installation
rem               directory
rem
rem   DB3Path:     Drive and path of the DB3 test items
rem
rem   Outputfile: Results are stored in this file. If it
rem               exists already results are appended to it,
rem               otherwise it will be newly created.
rem
rem
echo ****************************************************
echo ***** ConfPeaqBasic  V1.0  (c) OPTICOM, 1999   ***
echo ****************************************************
echo.
@echo on
call RunPeaqBasic %1 %2\Conformance-BS.1387\arefsna.wav
%2\Conformance-BS.1387\acodsna.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\breftri.wav
%2\Conformance-BS.1387\bcodtri.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\crefsax.wav
%2\Conformance-BS.1387\ccodsax.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\drefryc.wav
%2\Conformance-BS.1387\dcodryc.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\erefsmg.wav
%2\Conformance-BS.1387\ecodsmg.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\frefsb1.wav
%2\Conformance-BS.1387\fcodsb1.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\freftri.wav
%2\Conformance-BS.1387\fcodtr1.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\freftri.wav
%2\Conformance-BS.1387\fcodtr2.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\freftri.wav
%2\Conformance-BS.1387\fcodtr3.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\grefcla.wav
%2\Conformance-BS.1387\gcodcla.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\hrefryc.wav
%2\Conformance-BS.1387\hcodryc.wav %3
```

```
call RunPeaqBasic %1 %2\Conformance-BS.1387\hrefstr.wav
%2\Conformance-BS.1387\hcodstr.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\irefsna.wav
%2\Conformance-BS.1387\icodsna.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\krefsme.wav
%2\Conformance-BS.1387\kcodsme.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\lrefhrp.wav
%2\Conformance-BS.1387\lcodhrp.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\lrefpip.wav
%2\Conformance-BS.1387\lcodpip.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\mrefcla.wav
%2\Conformance-BS.1387\mcodcla.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\nrefsfe.wav
%2\Conformance-BS.1387\ncodsfe.wav %3

call RunPeaqBasic %1 %2\Conformance-BS.1387\srefclv.wav
%2\Conformance-BS.1387\scodclv.wav %3
```

5.6.4 More Examples ...

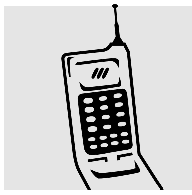
More example set-ups, especially some more exotic applications can be found in our paper "OPERA Application Notes" which is attached to this manual. We recommend to carefully look at the paper since some of the problems discussed there may well be transferable to your own application.

Chapter 6

6 TELEPHONY BAND VOICE QUALITY TESTING

Applying the Perceptual Speech Quality Measure (PSQM) and others to Voice Signals in a Telecommunication Environment.

Chapter 4 dealt with the general operation of the OPERA™ framework. This chapter is about all the specific concerns in assessing telephony band voice quality. In addition, the fundamentals of the corresponding measurement methods will be described. Finally, some measurement examples at the end of this chapter will help with your first measurement applications.



6.1 What To Know About Testing Telephony Band Voice Quality

Subjective quality assessment of speech codecs is one of the key technologies in designing digital telecommunication networks. Recommendation P.830 defined subjective testing methodologies for speech codecs. Since subjective quality assessment is time-consuming and expensive, it was therefore desirable to develop an objective quality assessment methodology to estimate the subjective quality of speech codecs with less subjective testing.

In the past, the most widely-used objective speech quality measure was the Signal-to-Noise Ratio (SNR = S/N). However, it was pointed out that the SNR does not adequately predict subjective quality for modern network components. This is especially true for recent low bit-rate codecs.

Within the telecommunication sector of the ITU, in 1996 study group 12 finalized recommendation P.861 [ITUT861] for the objective analysis of speech codecs. After a wide-ranging comparison of proposed methods, the group opted for the PSQM algorithm. PSQM correlated up to 98 percent with the scores of subjective listening tests. This high correlation was excellent in 1996 and still is extraordinarily good, however, as soon as PSQM is applied to signals that are out of the scope of the recommendation, the correlation will usually drop down significantly. This is mainly due to the fact that the coding and network technology has dramatically changed since the time PSQM was developed. One of the key technologies that demand different measures is Voice over IP (VoIP). Although the PSQM implementation in OPERA circumvents the major weakness of PSQM by extending the algorithm with a time alignment algorithm that can handle varying delays as they show up on packet oriented networks. This solution will, nevertheless, never be the ideal solution for measurements on such networks, if the implementation needs to conform to the P.861 recommendation.

Due to the widely extended demand for measurement algorithms suitable for packet oriented transmission in real networks, the ITU started developing a new recommendation with a widely extended scope. The outcome was recommendation P.862 (PESQ) [ITUT862][BEER02a][BEER02b], which is in place now since 2000. Although the basic structure of PESQ is very similar to PSQM, many details in all parts of the algorithm have been improved. The two major advances are PESQ including a very good time alignment algorithm that can handle varying delays and that the final result is a MOS score. For PSQM the function which mapped the PSQM score to the MOS scale was proprietary and so frequently leading to totally different results between two implementations. The following chapter explains PSQM in more detail. PSQM is chosen here as an example for all modern perception based measurement methods. Both algorithms will be explained in even more details in the according chapters where their implementation and usage are described.

6.2 Reference Files for Voice Quality Testing and Echo Measurements

OPERA Systems are delivered with a complete set of Test files. All WAVE files are stored in the folder c:\programme\opera\wavefiles. Besides the reference files used for performing life tests, there is a pair of reference and test file for each algorithm which are used for demonstrating file based measurements. These files are called AlgorithmRef.wav and AlgorithmTest.wav, where Algorithm must be replaced by PSQM, PESQ or Echo. These files must not be used as reference files for real measurements. For this purpose the files according to the following table are supplied.

Filename	**Description**
DefaultRefFileEnglish.wav	English language, male and female speaker, mu-law.
DefaultRefFileEnglish-a.wav	English language, male and female speaker, a-law.
DefaultRefFile.wav	Same as DefaultRefFileEnglish.wav
DefaultRefFileGerman.wav	German language, male and female speaker, mu-law.
DefaultRefFileGerman-a.wav	German language, male and female speaker, a-law.

DefaultRefFileMixed.wav	Mix of different languages. The contents was selected following linguistic criteria, male and female speaker, mu-law.
DefaultRefFileMixed-a.wav	Same as DefaultRefFileMixed.wav but a-law

For measurements on analog lines it does not matter if a-law or mu-law coded files are used, since they will be converted analog in any case. For ISDN, E1 or T1 interfaces however the coding must be chosen conforming to the network requirements. All reference files conform to P.800. Instead of the delivered files, any other speech file may be used as well.

6.3 PSQM as an Example for Perception Based Measurement Algorithms

The objective of PSQM is to mimic the sound perception of subjects in real-life situations [BEER94]. PSQM simulates experiments in which subjects judge the quality of speech codecs. This is done by comparing a coded signal (characterized as output y[t] in **Figure 6.1**) to a source signal (input x[t]). For this reason, experimental parameters – subject properties and listening conditions – have to be taken into account. These parameters are the listening level, the weighting on silent intervals, environment noise in the receiving side, characteristics of the hearing threshold and finally the sending and receiving characteristics of the handset.

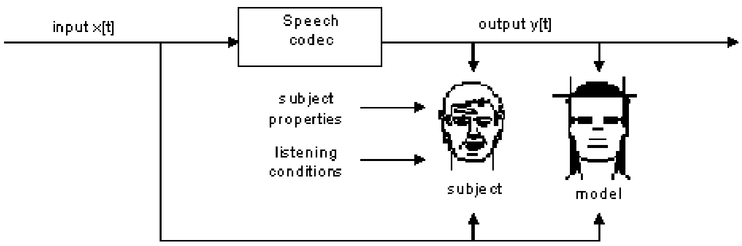


Figure 6.1: Overview of the basic philosophy used in the development of the PSQM algorithm [ITUT861]

To the extent that PSQM is a faithful representation of human perception and judgement processes, signals with inaudible differences between input and output will receive the same PSQM score. In particular, if the input and the output are identical, PSQM will predict perfect quality irrespective of the quality of the input signal.

Within PSQM, the physical signals constituting the source and coded speech are mapped onto psycho-physical representations that match the internal representations of the speech signals (the representations inside our heads) as closely as possible.

As depicted in **Figure 6.2**, the quality of the coded speech is judged on the basis of differences in the internal representation. This difference is used for the calculation of the noise disturbance as a function of time and frequency. In PSQM, the average noise disturbance is directly related to the quality of coded speech.

Besides perceptual modelling, the PSQM method also uses cognitive modelling in order to achieve high correlation between subjective and objective measurements [ITUT861]. The result is the estimated quality of the received signal.

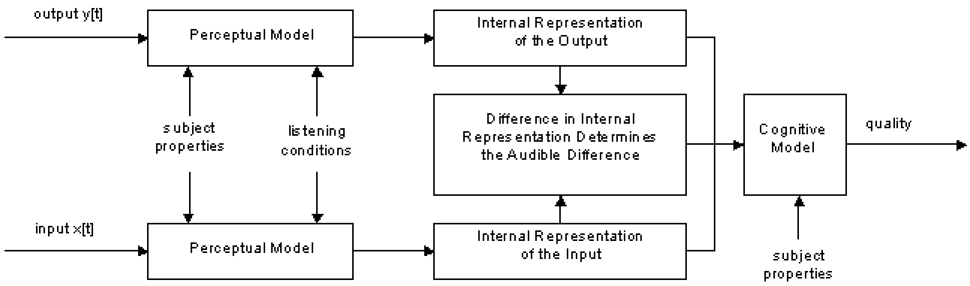


Figure 6.2: Block diagram of the basic model of the PSQM algorithm [ITUT861]

6.4 PSQM or PESQ, which one shall I use?

Since there are two speech quality algorithms available as ITU recommendations now, the user has to decide which one to use. This decision however can be based on some very simple rules:

- Whenever possible use PESQ. It is significantly better than PSQM.

- If PESQ cannot be used for any reason, use PSQM, but take care of varying delays.

- The only reasons why the use of PSQM could be required are, to perform online measurements, to compare your results to older measurements or simply that a PESQ license for your OPERA system has not yet been ordered.

- When using PSQM for any of these reasons, look carefully at the time alignment. Although this is not a real issue with conventional PSTNs, it is a significant problem on packet based networks. In these networks usually varying delays are observed. To compensate for these varying delays, switch OPERA's delay tracking option on.

Note:

Never compare:

- MOS values obtained from PESQ measurements to those obtained from PSQM measurements.

- MOS values obtained from PSQM implementations from different manufacturers. They may and usually do differ significantly (of course not between different OPERA systems....).

6.5 PSQM Measurement

6.5.1 Fundamentals of the PSQM Measurement Algorithm

The algorithm to calculate the perceptual speech quality measure (PSQM) was introduced by Beerends in 1993 [BEER94]. This development by KPN Research represents an adapted version of the more general perceptual audio quality measure (PAQM) [BEER92], optimized for telephony speech signals. This is due to the observation that the psycho-acoustic effects known from masking experiments differ significantly, when comparing the perception of speech and music signals. One reason might be that the human brain possibly recalls the reference sound of familiar voices more accurately from the daily life experience, compared to music sounds. Up to now, no single homogeneous approach has been presented that would allow for high correlation with both, speech, and music signals without adapting algorithm parameters [BEER95].

Figure 6.3 depicts a detailed block diagram to calculate PSQM. In the first step, the time domain representations of both input signals, x and y are transformed to the frequency domain. This transformation is accomplished by selecting blocks of the input samples that are input to an FFT. A Hann window is applied. The (linear) frequency scale is transformed to a pitch scale ("frequency warping"). The pitch modelling is also often referred to as "Bark transformation". Both, the reference, and the test signal are then filtered with the transfer characteristics of the receiving device (e.g. handset, loudspeaker, or headphones). A "Hoth noise" signal is added to simulate the background noise present in a typical office environment. The objective is to take into account the masking effects of real world environment noise, to properly model a masked threshold. The subsequent process of "intensity warping" leads to a representation of a compressed loudness as a function of pitch and time. By subtracting the two signal representations, an estimate of the audible error is derived. The difference signal is - of course - still a function of pitch and time.

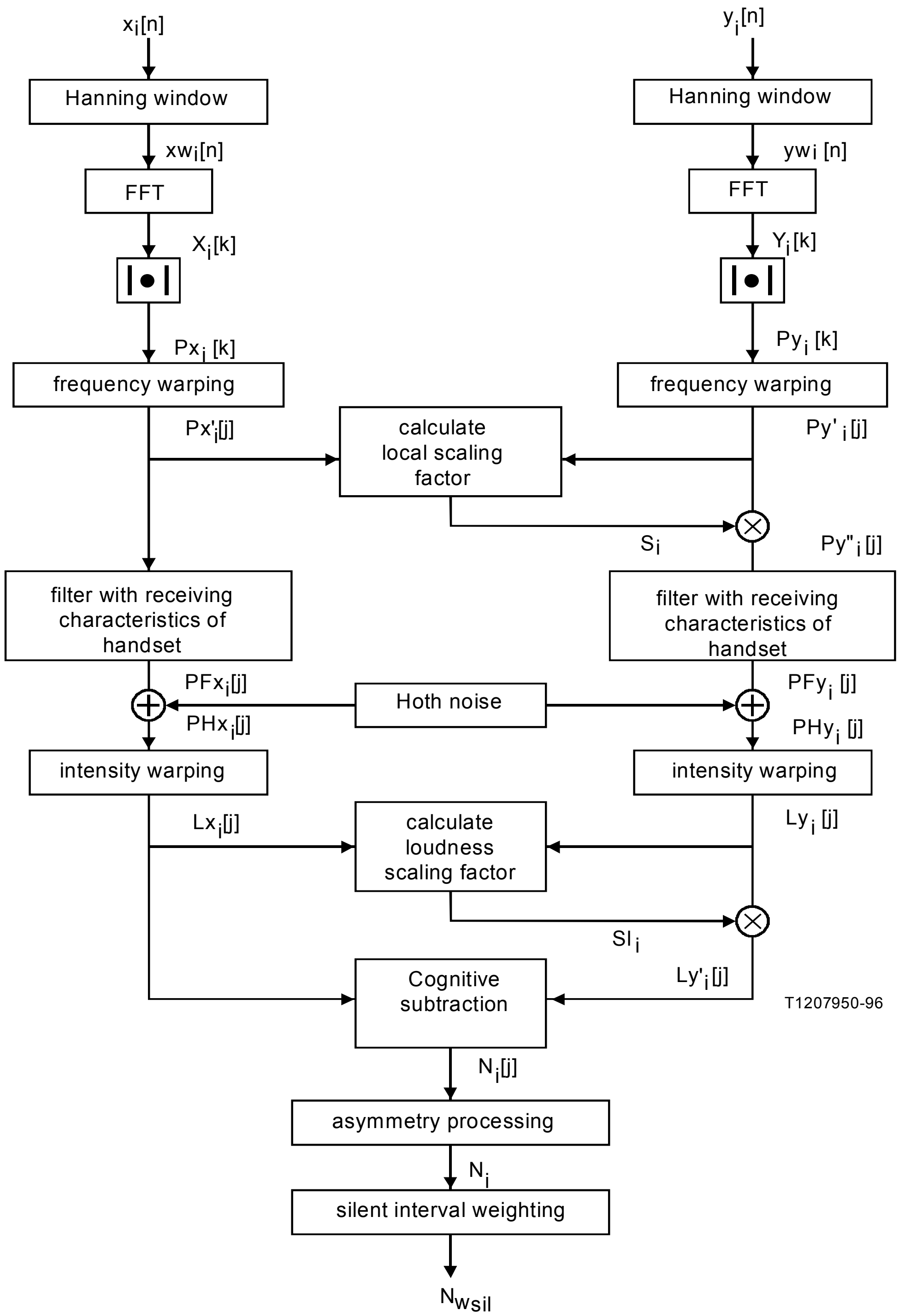


Figure 6.3: Block diagram of the PSQM Algorithm according to [ITUT861]

The following blocks are intended to represent the cognitive part of the modelling. The "asymmetry processing" should take into account that distortions, which were introduced by the device under test, are more easily perceived than signal components that were left out by the codec. Finally, the "silent interval weighting" will differ between silent and speech active intervals over the time. It is believed that this parameter allows a fitting of the cognitive processing to cultural differences. It was shown that almost identical subjective tests carried out at several locations in the world, and comprising different languages have led to different results, for instance in Europe, and Asia. It was concluded that the difference results from language differences, and the accompanied cultural differences. For example, a noisy floor may be more annoying if there are more silent intervals during a telephone conversation.

The PSQM algorithm was one out of several proposals that had been brought before study group 12 of ITU-T in 1995 for the purpose of international verification. Further proposals were the EPR Algorithm ("Expert Pattern Recognition"), which consisted of measures of the "LPC Cepstrum Function", "Information Index", and the "Coherence Function" (CHF). In a test series conducted by the Japanese phone corporation NTT, including listening tests in Japan and Italy, the highest correlation was achieved with PSQM results, when compared to the subjective tests. Consequently, PSQM was recommended by the ITU-T in 1996 for the objective quality measurement of telephone band speech codecs. Since then, PSQM has been used intensively for R&D as well as field applications in networks.

6.5.2 Signal Acquisition

PSQM can be used for online as well as for offline measurements. As **Figure 6.4**, shows, two kinds of signal sources be used. All OPERA™ versions can assess files. The version including a voice board can issue test calls to acquire the data, but the actual evaluation is also performed offline. For the data acquisition use **OptiCall™** which is described in chapter 4.2. Online measurements with phone lines are currently not supported. However, if there is an audio interface option included in your OPERA™ system – in addition to or instead of a POTS telephony board – evaluations of audio sources (e.g. VoIP terminals) with PSQM are available.

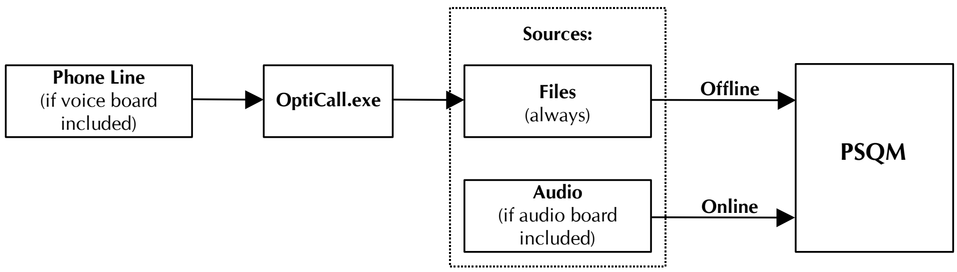


Figure 6.4: Kinds of signal sources for the PSQM measurement

File Based Version

When using the file based version of the Telecom Version of OPERA™ (OPR-100-xxx-x), no telephony interfaces are provided. Instead of this, use audio files as input. Supported file formats are **WAVE** files containing either plain PCM and **a-law** or **μ-law**. The PSQM algorithm is defined for sampling rates of **8 kHz** and **16 kHz**.

As described in Chapter 4, select the files you want to assess in the first two steps of the measurement set-up wizard.

Version with Telephony and/or Audio Interfaces

OPERA™ systems equipped with audio interfaces may be used for online measurements in real time too. Use our OptiCall program for the data acquisition if the system is equipped with telephone interfaces. OptiCall may also be used together with the audio interfaces. Please refer to chapter 4 for details on the data acquisition.

6.5.3 PSQM Algorithm Properties

PSQM always simulates a listening test. The following will reference that simulated test as a **virtual** listening test. To obtain results that highly correlate with those results that would have been obtained from subjects in a real listening test, PSQM must know some parameters of that virtual listening experiment. The following parameters must therefore be entered in the algorithm properties dialog (see also Section 4.4.2):

The listening condition, indicating if the virtual experiment uses loudspeakers, headphones or typical telephone handsets for listening.

The level of the background masking noise that was present during the virtual experiment "**Hoth noise**". Real life will always have background noise that produces masking effects. Even in silent environments this noise is in most cases higher than 30 dBA. This effect is modelled by adding background Hoth noise to the reference as well as to the test signal.

The level at which the signal is played to the subject in the virtual listening test.

The upper frequency, representing the upper frequency limit of the measurement.

Figure 6.5 shows the settings as they are recommended by P.861 for signals with an average active speech level of -26dBov (this is e.g. the setting for the NTT speech database).

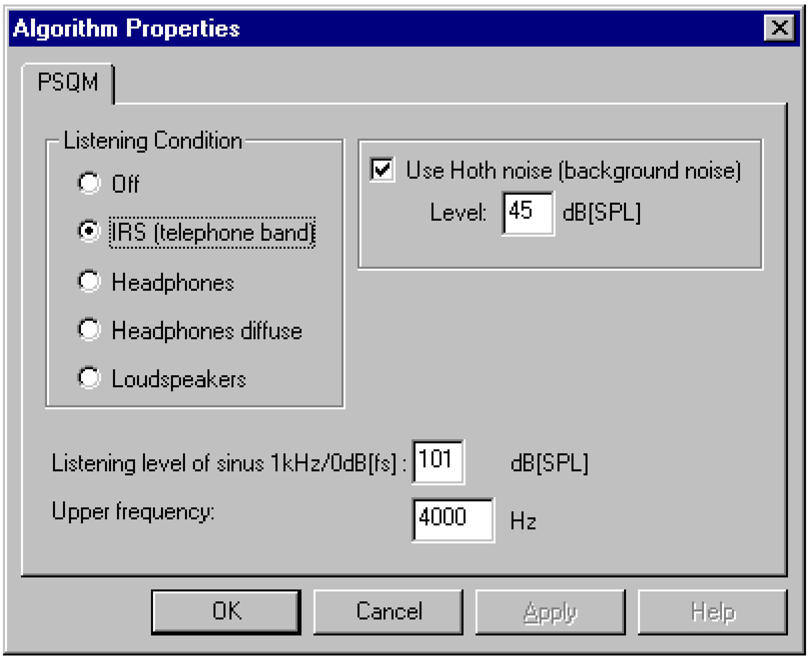


Figure 6.5: PSQM Algorithm properties as recommended by P.861

6.5.4 Diagram Types

In Chapter 4 the Section "Getting to Know the OPERA™ Framework" shows how to select a measurement algorithm and how to start a measurement. Once the measurement is performed, results should be displayed. For the PSQM algorithm, there are seven diagram types (see **Figure 6.6**) available that will be described in this section. If not otherwise mentioned, all diagrams show either framewise values (Timesignals, Spectra, Excitation), values averaged since the start of the measurement (Raw PSQM, MOS etc, MOS vs. time), or overall values (Final Result). The values shown in the "Final Result" diagram are the only ones, that take the start *and* the stop point into account, as defined by P.861. All other averaged values ignore the stop point. The start and the stop point exclude leading and trailing silent periods from the measurement.

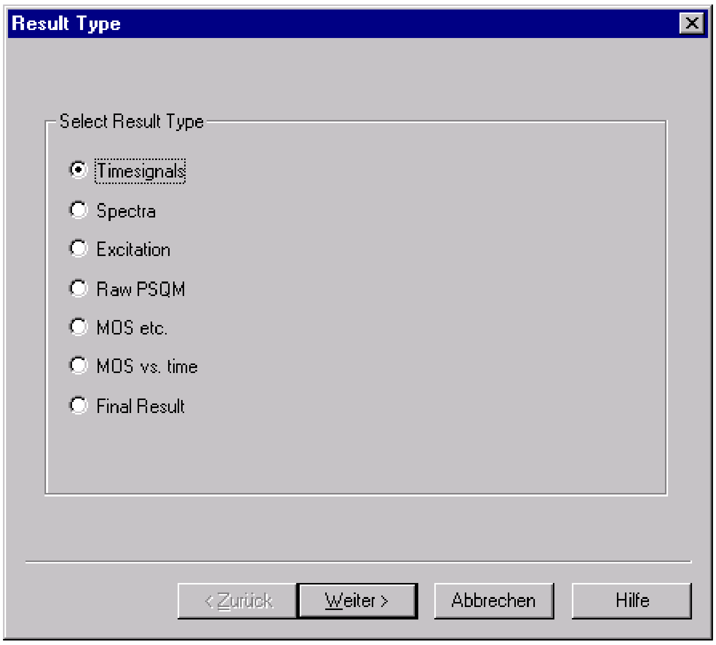


Figure 6.6: PSQM Result Type Window

Display of the Measurement settings

The information about the current measurement settings that are displayed on the right side of each diagram is described first(see **Figure 6.7**). The meaning of these values is as shown in **Table 6.1**:

Displayed Values	Interpretation
Time:	The time when the measurement has been finished
Hoth Noise	Setting of the level of the background masking noise
F Smpl:	Sample rate of input signals
Filter	Setting of the listening condition
BW Limit:	Highest frequency component taken into account
Level:	The current setting of the listening level
Tracking:	Status of the delay tracking function (on or off)
DC Filter:	Status of the DC filter (on or off)
Delay:	Delay in ms (first from top) as well as in samples (second from top)
Status:	Reliability of the automatic delay compensation (0..100%, Fixed=fixed delay set).
Atten:	Level difference between reference and test signal (dB)
Rel Time:	Current point of time in the measurement

Table 6.1: Interpretation of the displayed values

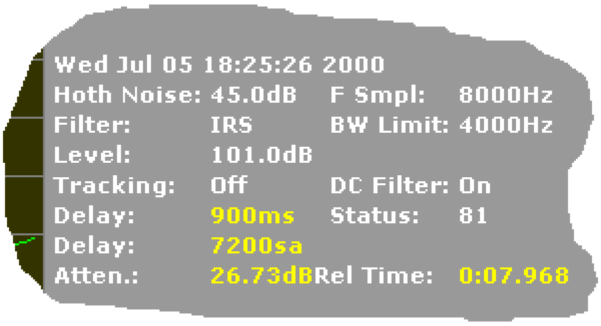


Figure 6.7: Display of the current measurement settings of the PSQM algorithm

Timesignals

To choose this diagram type highlight the radio button next to **Timesignals**, and press **Next**. This leads to the next wizard step, the **Signal Select** dialog.

With the **Select Signal** dialog (see **Figure 6.8**) a number of channels and input signals is defined that will be shown together in one diagram. Modify the selection by clicking with the left mouse button on any of the option buttons. This will add or remove the check mark in the button. A checked button means that the results for the selected signal will be drawn in the diagram. In **Figure 6.8** the results for the left channel of the reference and the test signal were selected. Each signal will be drawn in a different colour. Observe the assignment of the colours in the field to the right of the diagram panel.

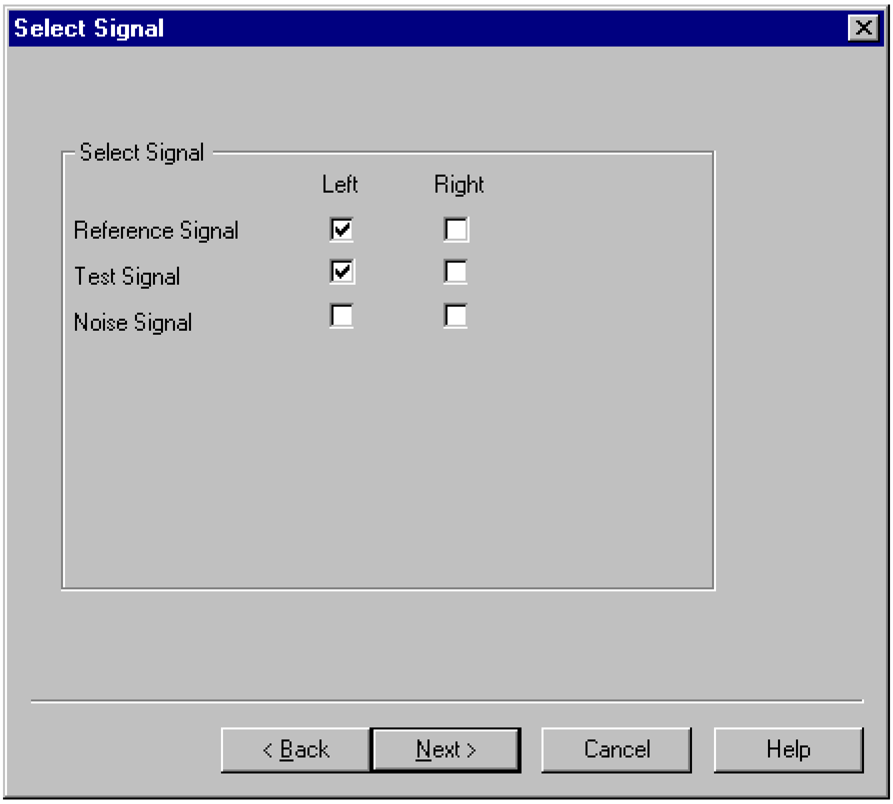


Figure 6.8: Select Signal Window

Pressing **Next** again leads to the next step, the **Result Style** dialog (see **Figure 6.9**. Here select the way data is shown on the screen. Usually this selection is identical to selecting the units of the diagram axes. For the time signals choose between a binary, linear representation in which the input signals are always scaled to **[-32768 ... +32767]**, or a **dB FS** (full scale, = dB_{ov}) scale.

Note:

Independent of the input data format, samples are always converted to **16bit/sample**. This means that 8bit/sample data are multiplied by 256 before they are processed any further.

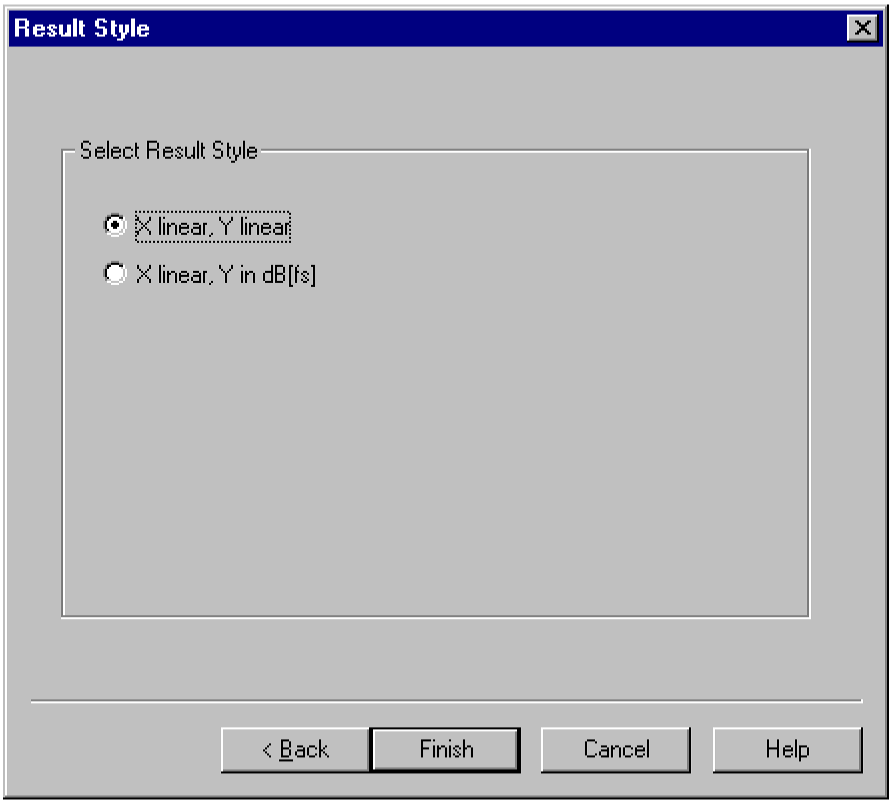


Figure 6.9: Result Style Window

After this last step click on **Finish** and the selected diagram will appear in the diagram pane as shown in **Figure 6.10**. Observe an excerpt of the time signal of one frame.

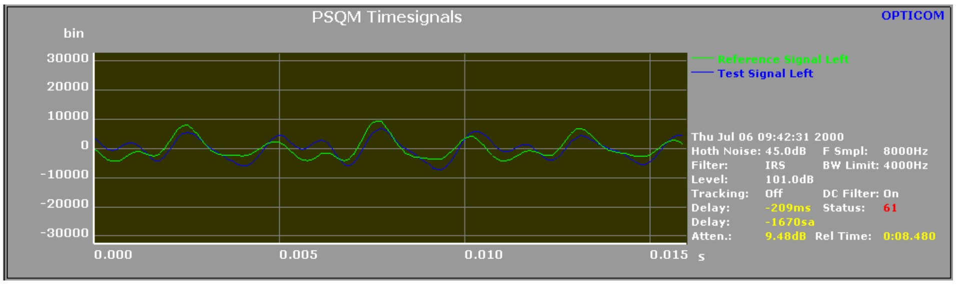


Figure 6.10: Time signals diagram

Spectra

Figure 6.11 shows the signal select wizard step for the spectra diagrams. Select the spectra of the reference signal, the test signal or the difference between the reference spectrum and the test spectrum.

Available result styles for the spectra (see **Figure 6.12**) are a linear frequency scale or a Bark scale (as outlined in P.861). The Y-axis is always scaled in dB SPL. Note that the Y-axis is depending on the setting of the listening level as set by the algorithm properties dialog. The spectral resolution is dependent on the sample rate of the input signals. At an input sample rate of 8kHz, a 256 point FFT is used with a Hann window and 50% overlap to compute these data (as required by P.861).

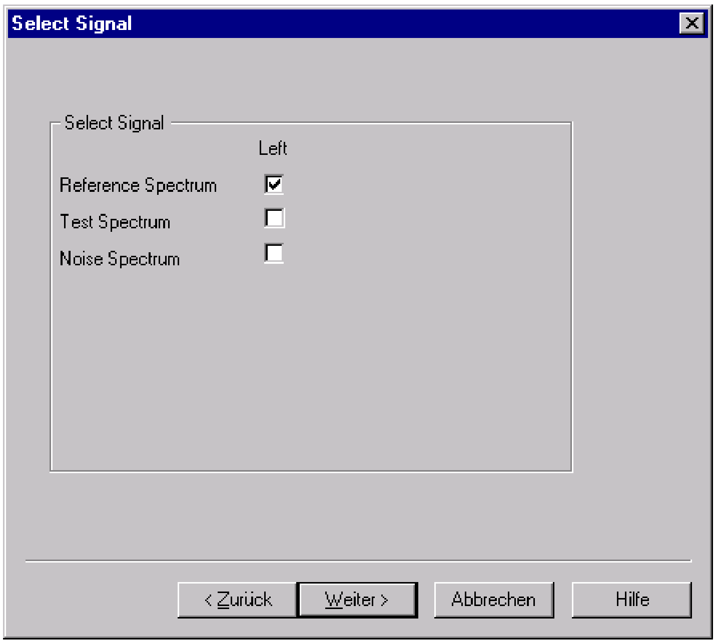


Figure 6.11: Select Signal Dialog

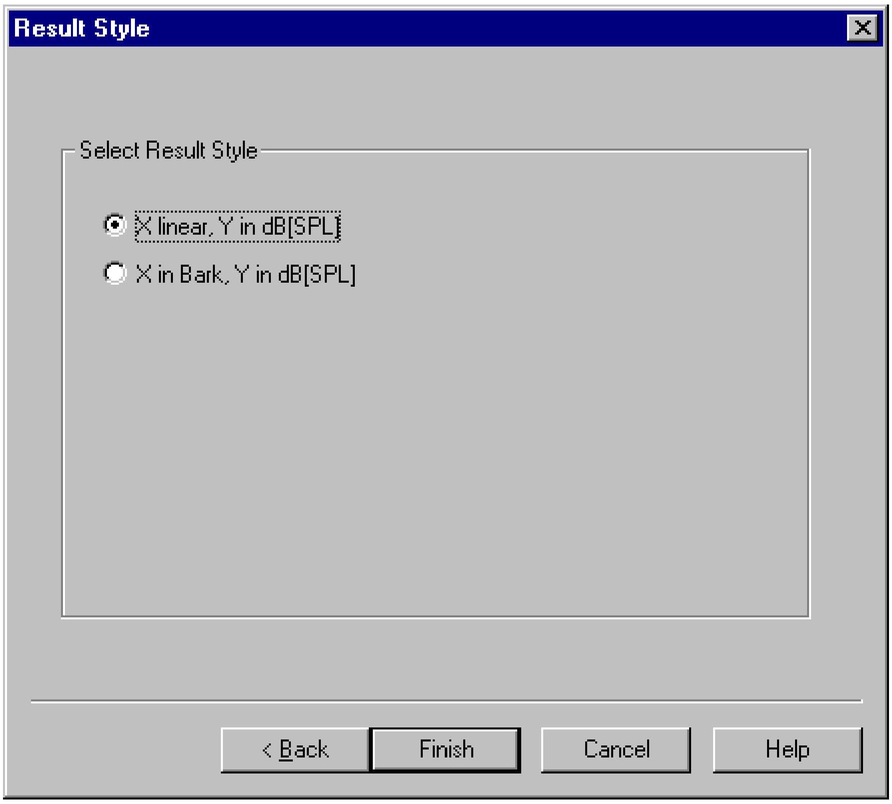


Figure 6.12: Result Style Dialog

The resulting diagram is as shown in **Figure 6.13**.

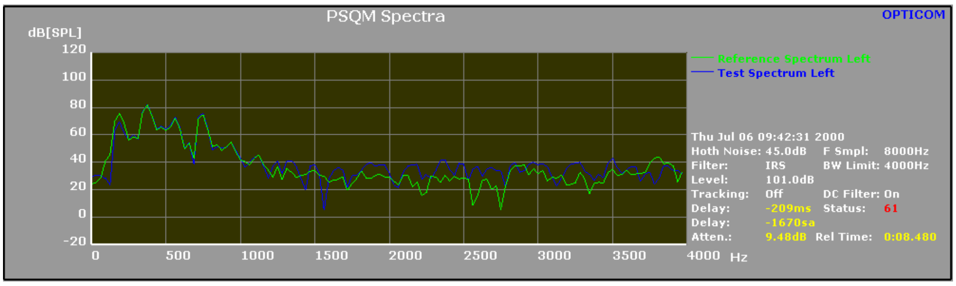


Figure 6.13: The Spectra diagram

Excitation

The excitation diagram displays the internal representation (see Section 2.2) of a signal. The Bark scale is used for the horizontal axis, the vertical axis is scaled in dB SPL. The curves are shown on a frame by frame basis, without any averaging.

As shown in **Figure 6.14**, the reference signal excitation, the test signal excitation and the difference between both excitations can be selected. The difference is calculated by subtraction in the linear domain, then displayed in dB SPL.

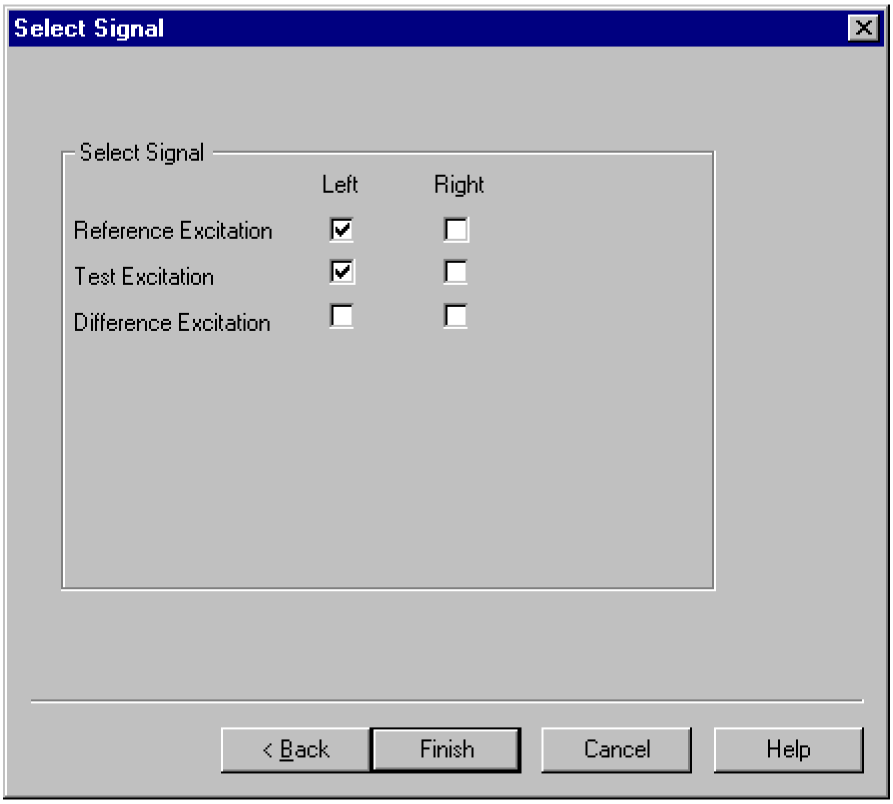


Figure 6.14: Select Signal Window for the excitation diagram type

Figure 6.15 shows the resulting view off the Excitation diagram.

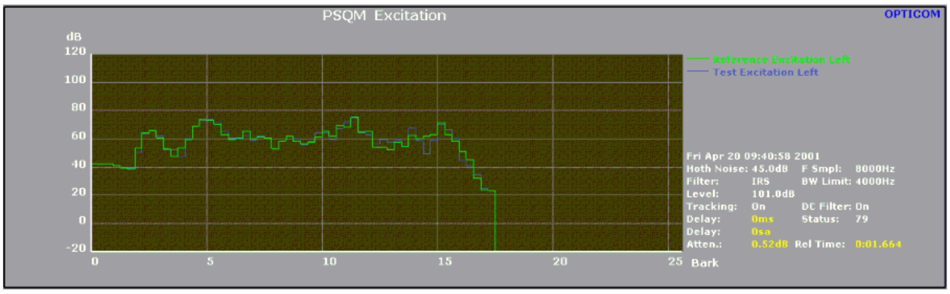


Figure 6.15: The Excitation diagram

Raw PSQM

Figure 6.16 represents the diagram of the Raw PSQM. There are several values shown, the description of the result values is given in **Table 6.2** below.

The PSQM value indicates the degree of subjective quality degradation as a result of speech coding. For this reason, when an estimation of subjective quality on a specific scale is not necessary, e.g. in optimizing parameters of a codec or in simply comparing the performance of codecs, the PSQM value itself is quite useful.

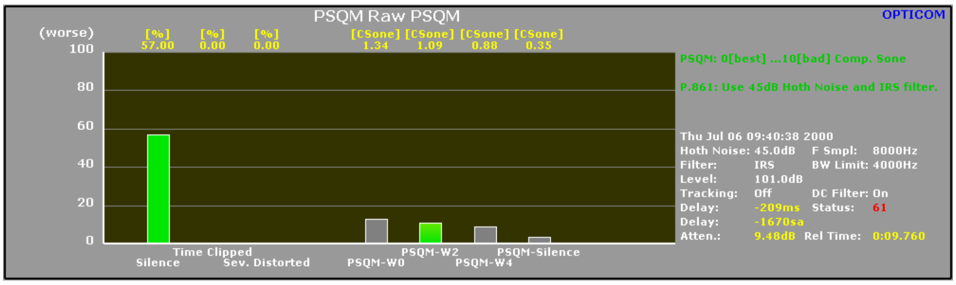


Figure 6.16: The Raw PSQM diagram

In PSQM, the silent intervals are taken into account using a weighting factor that depends on the context of subjective experiments, i.e. the portion of silence intervals varies from one culture group to the other. As **Figure 6.16**, three PSQM values are displayed that use different weighting factors. For European languages to take the PSQM-W2 value into account is recommended.

Model Output Variable (MOV)	Interpretation
Silence	The percentage of silent intervals during a measurement
Sev. Distorted	The percentage of severely distorted frames during measurement
Time Clipped	The percentage of time clipped frames during measurement
PSQM-W0	PSQM according to P.861, silence weight = 0.0
PSQM-W2	PSQM according to P.861, silence weight = 0.2
PSQM-W4	PSQM according to P.861, silence weight = 0.4
PSQM Silence	PSQM value of the silence intervals

Table 6.2: MOVs used by the OPERA™ PSQM version, and their interpretation

MOS etc.

In subjective assessment of the performance of codecs, the ACR method using the Listening Quality scale specified in Recommendation P.800 is often used, giving subjective quality in terms of MOS (see chapter 2). The OMOS value represents the Objective Mean Opinion Score.

Model Output Variables

Table 6.3 presents the interpretations of the bars in the diagram shown in **Figure 6.17** (from left to right). The OMOS+ is the PSQM+ result, mapped to the MOS scale. PSQM+ is a further development of PSQM, optimized for severe impairments as they occur in the case of VoIP and realistic GSM connections. OMOS-I is the PSQM result mapped to a full five point MOS scale, ranging from 1 to 5. This represents an individual, ideal listener. The average listener will usually not exploit the full range of this scale.

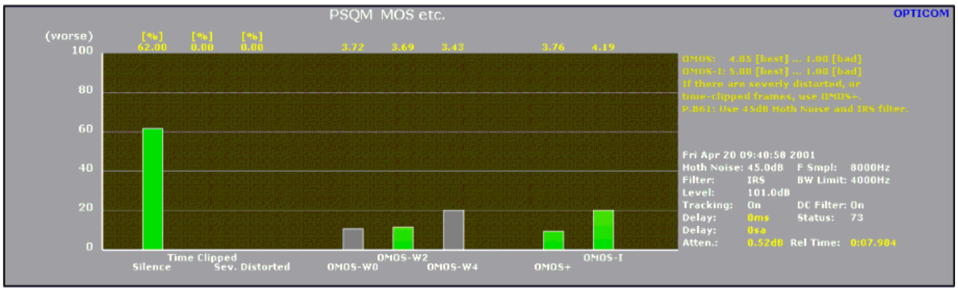


Figure 6.17: The diagram type MOS etc.

Model Output Variable (MOV)	Interpretation
Silence	The percentage of silent intervals during a measurement
Sev. Distorted	The percentage of severely distorted frames during measurement
Time Clipped	The percentage of time clipped frames during measurement
OMOS-W0	MOS according to P.861, silence weight = 0.0
OMOS-W2	MOS according to P.861, silence weight = 0.2
OMOS-W4	MOS according to P.861, silence weight = 0.4
OMOS+	MOS according to PSQM+
OMOS-I	MOS of the individual listener

Table 6.3: Results calculated by the OPERA™ PSQM version, and their interpretation

Table 6.4 shows the simple interpretation of the MOS (P. 801):

The ITU-T P.801 MOS scale

MOS Value	Interpretation
5	= Excellent
4	= Good
3	= Fair
2	= Poor
1	= Bad

Table 6.4: The opinion scale according to P.801

Since subjective tests show an average between **4.05** and 4.5 for transparent quality (some listeners always hear some distortions...), the MOS scaling of PSQM ranges from 1.0 to 4.05 (and not up to 5.0). The only exception from this is the OMOS-I. Since this value represents the behaviour of an individual, ideal listener, it covers the full range of the ITU scale (1..5).

During a measurement the result values are averaged between the start sample and the end of either the reference, or the test file. The stop point as defined by

P.861 is not taken into account. To know the result at the stop point, Refer to the "Final Result" diagram.

ITU P.861 provisionally recommends a silence weight of 0.2 to be used for the MOS calculation. A value of 0.4 is sometimes found in listening only experiments. Concerning a conversational quality prediction, a silence weight of 0.0 is recommended.

Note:

An interpretation of MOS values using different playback level settings is dangerous. If the playback level is decreased, the noise disturbance will decrease, thus leading to an increase in MOS. There is no experimental data available to verify this level dependency of the MOS value.

MOS vs. Time

This diagram type (see **Figure 6.18**) shows the PSQM MOS over all frames in the history buffer. This figure is not an averaged value and no weighting factor is used for this result value. When scrolling through the buffer, a pink cursor indicates the current position within the diagram.

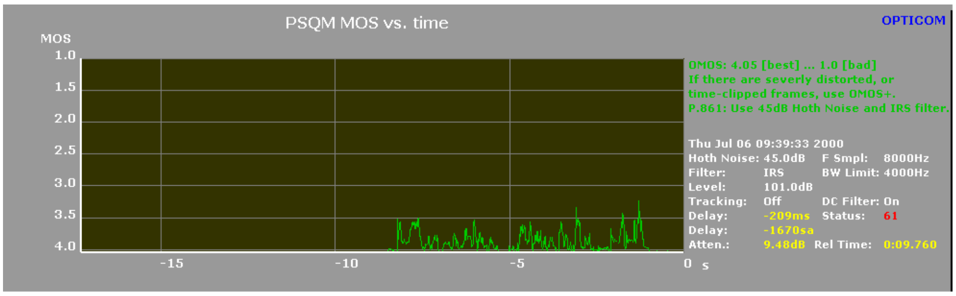


Figure 6.18: The MOS vs. Time diagram type

Final Results

This window (**Figure 6.19**) shows the most important results at the end of the measurement. The end is defined by the stop point according to P.861, which means that trailing silence before the end of the input files/streams is excluded. The content of this diagram remains constant even when scrolling back through the history buffer. To scroll back in the buffer to find the moment of the stop point, just remember the "Rel. Time" field of this diagram and scroll back through the buffer until the "Rel. Time" field of any of the other diagrams matches this value.

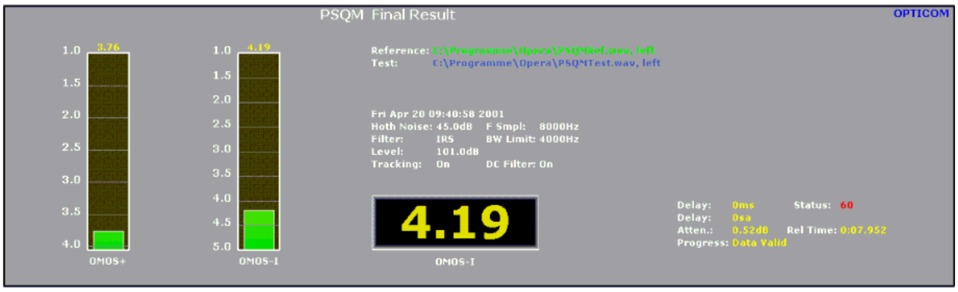


Figure 6.19: The PSQM Final Result diagram type

6.5.5 Command Line Arguments

In addition to the command line arguments described in Chapter 4 specific commands will be described in this section.

PSQM currently interprets the following algorithm specific command line switches. If no switches are specified, the default settings for correct measurements according to P.861 will be chosen (45dB Hoth noise, 4kHz upper limit, IRS filer, 101dBSPL listening level).

These commands are to be used together with the **-Algorithm Name=PSQM** switch:

Keyword	Add. Parameters	Comment
HothNoise		Use background masking noise
HothNoiseLevel	Level of Hoth noise in dBSPL	Level of background masking noise
ListeningLevel	Level of a 1kHz 0dBov sine tone in dBSPL	Listening level acc. to P.861
UpperFreq	Frequency in Hz	specify upper frequency limit for measuring
Flat		Listening condition: Flat frequency response
IRS		Listening condition: IRS telephone) filtering
Headphones		Listening condition: Headphones
HeadphonesDiff		Listening condition: Headphones (diffuse field)
Speakers		Listening condition: Loudspeakers

Table 6.5: PSQM specific command line parameters.

6.5.6 Common Mistakes

When all of your measurement results show a MOS of 5.0, while at the same time clearly audible distortions exist, please check the following:

There are clearly Audible Distortions, but PSQM scores around 5.0

- Are the correct files used?
- Are the listening level and the upper frequency limit set up properly (check under Algorithm Properties)?
- If "Delay Tracking" is enabled, it will discard all frames, for which it could not detect a reliable delay. In extreme cases this may result in almost all invalid frames and the default score of 5.0.

I always measure a MOS near 1.0

If you always measure a MOS of 1.0, the most frequent reason is – despite the trivial solution of mixed up files – that either the Delay Compensation algorithm is not set to "Auto", or that the delay can not be compensated by the system due to its length or variation. Check the delay status. If it is very low (<40) then examine the time signals and verify that the waveforms of the reference and the test signal do indeed match. Assuming the delay is simply too long, you can try setting a static delay offset. Supposing it varies too much, the scope of P.861 is exceeded in any case and measurements should be performed with PESQ.

6.6 PESQ Measurement and VAD Measurement

When PSQM was standardized as P.861, the scope of the standard at that time was state-of-the-art codecs, used primarily for mobile transmission, like GSM. VoIP was not yet a topic. The requirements for measurement equipment have changed dramatically since then. As a consequence, the ITU set up a working group to revise the P.861 standard to cope with the new demands arising from modern networks like VoIP. Within these networks, the measurement algorithm has to deal with much higher distortions than with GSM codecs, but perhaps the most eminent factor is that the delay between the reference and the test signal is not constant any longer.

A first approach to overcome these problems was the development of PSQM+. It handled the larger distortions well, as they are caused by e.g. burst errors, but still had significant problems with the compensation of the varying delay. A Delay Tracking Feature has been added by OPTICOM in it's OPERA™ system that implemented an easy way to solve the varying delay issue in most cases, without loosing the option of realtime operation. Although this feature failed for some signals, it was the only available method to date to achieve reliable results for the speech quality of VoIP networks.

With the new ITU standard P.862 (PESQ) this problem is finally eliminated. PESQ combines the excellent psycho-acoustic and cognitive model of PSQM+ with a time alignment algorithm that handles varying delays perfectly. PESQ is absolutely not designed for streaming applications, which is it's only drawback. This is also why it cannot fully replace PSQM+. With PSQM and PESQ there are now two standards that cover the entire problem of measuring speech quality. **Figure 6.20** gives an overview of the structure of the PESQ algorithm and shows the new blocks that have been added to the PSQM algorithm.

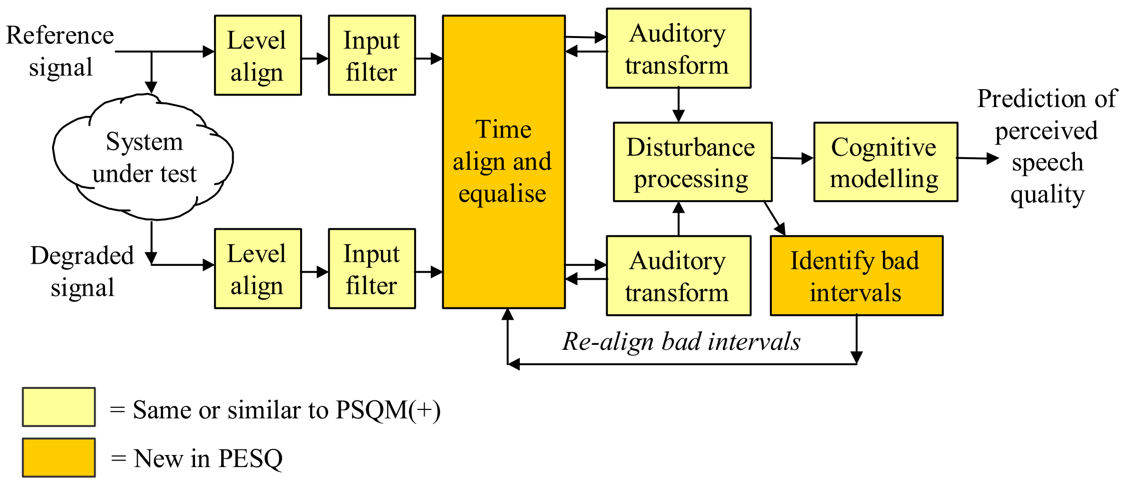


Figure 6.20: The structure of the PESQ algorithm

6.6.1 Advantage of using PESQ instead of PSQM

One of the major advantages of PESQ over PSQM(+) is, that it contains a real good time alignment algorithm, which is capable of handling varying delays. With PSQM, such time alignment was missing in the standard, and it became the responsibility of the implementers to resolve this issue. As experience showed, only very few PSQM implementations came with a time alignment algorithm that was well suited for static delays on real networks, and even fewer measurement systems were capable of handling varying delays, as they appear on e.g. packet based networks. One of the best algorithms was probably the one used in OPERA, which worked quite well for most situations, but as a result of its design for realtime applications, it also failed from time to time. As the outcome of the wrong time alignment, two parts of the reference and the test signal were compared that did not match and consequently did sound different. This sonic difference unquestionably led to a PSQM score that was too pessimistic and simply wrong. With PESQ, this shortcoming is finally eliminated and the user will obtain realistic results for the device under test. There is no danger any longer, that the tested system is downgraded, only because of a deficiency of the measurement algorithm.

6.6.2 Explanation of the Measured Parameters

This paragraph explains the basics of the parameters which are measured by the PESQ implementation on OPERA, as well as how these parameters are defined.

MOS

The most eminent result of PESQ is the MOS. It directly expresses the voice quality. The PESQ MOS as defined by the ITU recommendation P.862 ranges from 1.0 (worst) up to 4.5 (best). This may surprise at first glance since the ITU scale ranges up to 5.0, but the explanation is simple: PESQ simulates a listening test and is optimized to reproduce the average result of all listeners (remember, MOS stands for Mean Opinion Score). Statistics however prove that the best average result one can generally expect from a listening test is not 5.0, instead it is app. 4.5. It appears the subjects are always hearing distortions, even if there is no degradation at all of the signal available. OPERA can determine the MOS for the entire reference and test signal, for active speech parts of the signals only and for the silent parts of the signals. In the two latter cases, active speech is detected by using the VAD, which is part of the PESQ time alignment. Knowing the individual MOS scores is especially useful for optimising e.g. comfort noise generation or noise reduction systems.

6.1.1.1 P.800 MOS and PESQ-LQ

Listening tests are very difficult to repeat and will never give identical results. Moreover it is generally required to apply at least a linear transformation to the results of one test if they shall match the results of a second test (with identical test material, but performed at a different place or at a different time). The same holds for the correspondence between a listening tests and the PESQ MOS. If highest correlation is required, a linear mapping of the PESQ MOS to the scale which was actually used by the test subjects must be applied. The PESQ MOS according to P.862 was derived by optimizing a third order polynome to give highest correlation on a very large set of data. Although this is generally the best approach, it is of course possible to achieve higher correlations on a smaller set of data by applying a second polynome. One such approach is the PESQ-LQ value. It uses the following formula to transform the PESQ score (x) into the PESQ-LQ value (y):

$$Y = \begin{cases} 1.0, x \leq 1.7 \\ -0.157268\ x3 + 1.386609\ x2 - 2.504699\ x + 2.023345, x > 1.7 \end{cases}$$

This mapping was submitted to the ITU-T SG12 with the intention of extending P.862 by an annex or appendix. SG12 however clearly rejected this proposal, and we, OPTICOM are fully in line with this rejection for the following reasons:

- The PESQ MOS has the best overall performance. If a user requires the mapping of the PESQ score to another listening test, he has to perform his own mapping in any case. PESQ-LQ will be as wrong as any other parameter in this case.

- Having a second MOS-like parameter is confusing.

- Applying a second third order polynome to the already third order mapped PESQ MOS doubles the mathematical degree of freedom. This will increase the correlation on the data which were used for the parameter fitting ("training"), but it also increases the risk of complete failure on other data.

Similar is the situation with a MOS mapped to the full P.800 scale. Although in OPERA we use a linear mapping in this case only, we do not recommend using this value.

Both parameters are supplied in OPERA, due to the sole reason, that customers wanted to see them. In our opinion it is scientifically wrong to use them, and we do not recommend it. However we do recommend application of a linear mapping between the PESQ MOS and subjective results if a direct comparison to a specific listening test is required. This will compensate for differences in the MOS scales used by the listeners and by PESQ. The actual scale varies slightly between listening tests, which means in test 1 File a may be graded as a 3.2, while in test 2 it may score 3.4, while PESQ will always give the same result.

G.107 Rating, R Factor

The ETSI e-model as defined in ITU-T G.107 [ITUT107] is a planning tool that assigns a certain equipment impairment factor *I*e to each piece of equipment in the transmission chain. These *I*e values are then summed up and combined with several other parameters to give the final R factor or R rating. This R Rating is an estimate of the quality that can be expected if the network is realised the way it is planned. Although the e-model is an excellent planning tool, it can never replace measurements on the final network, since it has to make some very wide ranging assumptions. R ranges from 0 for terrible quality up to 100 for "users are very satisfied". Values below 50 are generally interpreted as "nearly all users are dissatisfied". Of course there is a well defined relation between R and the MOS score. To allow for the comparison between the estimates from the network planning phase and the QoS of the live network, OPERA provides the R factor as well. It is directly derived from the overall MOS, as it is calculated by PESQ [MÖLL02]. Please note, that the R value presented here is derived directly from the PESQ MOS. It takes neither delay, nor echo or attenuation into account and is in fact more corresponding to the G.107 I_e value than to the R factor (which is a conversational measure, rather than a listening quality index).

Delay and Delay Jitter (Latency)

As soon as a signal is processed by any piece of equipment, it will be slightly delayed. The resulting **delay** is also frequently called **latency**. During the transmission of a speech signal these delays may add up and become intolerable. Excessive delays predominantly influence the efforts required for a conversation. The longer the delay , the more discipline is required from both parties involved in a conversation. Delays larger than app. 300ms are generally unacceptable. While the delay for the old POTS is usually in the range of few milliseconds, it is typically around 150ms for VoIP systems, in some instances much longer. Extreme delays up to more than a second can be observed on satellite links.

In packet-based-networks the signal delay is not constant, a so-called **Delay Jitter** is present. In the context here delay and jitter are both referring to the speech signals. They are generally not directly related to the jitter of the IP packets. Delay jitter in the speech signal can have various causes. The most frequent however is the dynamic adaptation of the jitter buffers built into modern VoIP equipment. The purpose of these buffers is to assemble a continuous voice stream out of the RTP (=speech) packets which arrive in bursts with non-deterministic timing. The longer these buffers are, the more packet jitter they may compensate for, but the latency of the speech signal is also increased. On the other hand, if the jitter buffer is shorter, the latency is shorter as well, but the risk of packet loss is significantly higher. The optimum length of the jitter buffer is depending on the network itself and the load on the network. To optimize the latency of VoIP equipment, adaptive algorithms are used to automatically adjust the size of the jitter buffer to what ever is required by the network. These adaptations cause delay changes in the voice stream and usually happen during silent periods. Often however, adaptation during silence is not possible and audible distortions are the consequence.

OPERA supports the assessment of this behaviour by providing various delay parameters. The minimum, maximum and average delay in ms are provided to give an overview on the performance of the system under test. A more detailed analysis graphs with the delay vs. time as well as a histogram of the actually occurring delays are available. The delay histogram shows the probability of each individual delay value, this is much more meaningful than looking just at the average values. For a system where 90% of the time the delay is 80ms and 10%of the time the delay is 1000ms, the average delay will be 172ms. This is as misleading as looking at the extreme values only. While the first value appears quite acceptable, the second will give the impression, that the QoS is totally unacceptable. When looking at the delay histogram instead, the PDF (Probability Density Function) will show that most of the time the delay was excellent, just sometimes it was unusual. Then, further analysing the delay vs. time, you may find the reason for the excessive delays. The Delay Jitter is also shown as a separate value. It is defined as the maximum and minimum deviation of the delay from the average delay in ms.

All the delay measurements are derived from the PESQ time alignment algorithm. If a delay change occurs during silence, it is impossible to determine the exact position of the delay change within the silent interval. PESQ usually sets the delay change right into the middle of the silent period.

Waveforms

Within OPERA the Waveforms of the signals before, as well as after the time alignment can be shown. The signals are always directly derived from the raw data and no filtering is applied. The waveform before the time alignment is the raw signal as it is read from the WAVE file. The time aligned test signal, though, is already processed by PESQ, which sometimes leads to unusual looking situations at first glance. According to the P.862 standard, the time alignment algorithm will repeat parts of the test signal during short dropouts. Although this is very rare, the time alignment may also fail completely under some circumstances, e.g. if you have a reference sequence consisting of the phrases a-b-a and the test signal contains only b-a. In this case the time alignment algorithm may mess up the phrases. When in doubt looking at both the aligned as well as the unaligned waveforms will be helpful.

Attenuation and Gain

All analog equipment in particular introduces attenuation into the speech signal. A high attenuation generally leads to a worse perception of speech. PESQ does not weight this as a degradation, since it has no absolute reference level available. Also, in real world systems, a low speech level on the electrical side does not mean that the signal sounds very quiet, since the transducers used, have a significant impact on the final loudness. For PESQ it is therefore generally impossible to weigh the attenuation in terms of a perceived distortion. The value of the attenuation however is important for optimizing the overall system design. As delay, as well the attenuation/gain of modern telecom equipment is not constant anymore, almost every mobile phone and VoIP terminal has a built in AGC (Automatic Gain Control) or ALC (Automatic Level Control). These mechanisms both target the same problem. They amplify or attenuate the input signal before the transmission to compensate for very loud or very quiet talker, and in this way keep the signal level in the optimum operating range for the transmission. The gain adaptation happens constantly, with a reasonable high time constant. Of course it is of major interest to not only know the static behaviour of the transmission system, but also it's dynamic characteristics. For this purpose our PESQ implementation provides both, the overall attenuation in dB, as well as the gain/attenuation variation over time. Both are derived from the time aligned and IRS filtered input signals. Out of band energy below 300Hz is disregarded. The attenuation is calculated as the ratio between the reference and the test signal energy. The gain variation is updated only when one of the two signals exceeds the threshold in quiet.

Level Measurement

Closely related to the attenuation is the measurement of the signal levels. Here it is of special interest to know the signal levels separately for the active speech parts of the signals as well as for the silent parts. Naturally it is important to know these parameters for the reference as well as for the test signal, otherwise it is not possible to see the influence of the device under test. In OPERA all three parameters, the total level, the level of the active speech part and the level of the silent parts (background noise) are shown for the reference as well as for the test signal in dB_{ov}. These parameters are derived from the time aligned and IRS filtered input signals. Out of band energy below 300Hz is not taken into account. These values are exceptionally useful for the assessment of comfort noise generators.

Loudness Measurement

The loudness is a more psycho-acoustic view of the signal levels. It expresses not only how much energy is contained in a signal, but also how loud it is perceived by the listener. Of course PESQ can not know the characteristics of the telephone used, since it operates on the electrical level only, but it assumes a fixed relation between the binary signal level in the input files, and the loudness. It blocks the out of band energy, filters the input signals and calculates the overall energy. It is then assumed that this overall energy – still in dB_{ov} – corresponds to $79dB_{SPL}$ at the ear reference point according to P.830. To allow for the assessment of equipment exhibiting noise substitution, PESQ provides the loudness of the entire reference signal, the test signal and the silent intervals of the test signal separately. The actual loudness calculation is performed by the PESQ perceptual model as defined in P.862, taking into account the above relation between dB_{ov} and dB_{SPL}.

VAD Parameters, Front End Clipping (FEC) and Hold Over Time (HOT)

Most VoIP systems use VAD (Voice Activity Detection) to save bandwidth. If the VAD indicates active speech, then the encoder transmits packets containing speech. If the VAD decides on silence at the input, the encoder simply informs the decoder of the characteristics of the noise at the encoders input and this noise is then substituted by the decoder. This requires significantly less bandwidth, than transmitting the entire speech signal. The two most common problems of VADs are because they must meet severe realtime constraints. Once the VADs decide on active speech, they cannot tell the encoder a few milliseconds later that in fact their decision was wrong and that the encoder had to transmit the signals differently. It is imply too late and the signal is already transmitted. This false or slow detection of active speech and silence is characterised by two parameters, Front End Clipping (FEC) and Hold Over Time (HOT). The Hold Over Time is frequently also called Hang Over Time. Both parameters are expressed in ms and specify the time between the actual start of the active speech sequence until the VAD decides on speech (FEC), respectively the time after active speech ended and the VAD decided on silence (HOT). Opera gives the average HOT and FEC as well as HOT and FEC on a per utterance basis.

Measuring FEC and HOT in real networks is far not a trivial problem. Currently there are several methods available to assess these parameters. Some of these will be explained in the following. All methods have in common that they are not perfect and that they may fail under some circumstances. Especially if other network effects, like varying delay due to jitter buffer adjustments or packet loss occur at the same time as FEC or HOT. Under such circumstances it is almost impossible to determine the correct result, since the situation is ambiguous. Detection of FEC and HOT is much easier if special test signals are used in stead of real voice, however the situation for the device under test is also less realistic. Of course, artificial signals with algorithms that were designed to work for real voice can always be used, but not vice versa.

Noise Burst and low Level Sine Tone

The first method uses a noise burst, together with a low level sine tone ("dyer tone"). At the beginning of the test sequence both signals are overlaid. The noise is switched off after a well defined time and the sine tone continues. Since the duration of the noise burst and the frequency of the sine tone are known, it can be assumed that the difference between the duration of the received noise pulse minus the duration of the original pulse is the duration of the Front End

Clipping. HOT can be assessed by a frequency analysis of the received signal. As long as the reference sine tone is detected in the received signal, the VAD is open. Once it is closed, the decoder should generate comfort noise. The major problems with this algorithm are:

- The special test sequence used by this method is almost ideal for a VAD device, and the switching points between voice and silence are much easier to detect than with real speech.
- Inaccurate measurement of FEC if a delay variation or drop out occurs during the noise pulse.
- Transmission of a pure sine tone may be critical on some networks.

Correlation of the spectra

Another method is to correlate the spectra of the reference signal and the received test signal. This allows for a very accurate detection of HOT. However, the temporal resolution is limited to the FFT window size. A shorter window allows for a higher resolution, but the correlation results will become less reliable. This method can work with real speech signals.

Ideal VAD plus Time Alignment

This method is used in OPERA. It takes the time aligned signals from PESQ, realigns them if required and calculates an almost ideal VAD on both signals. It is much easier for the measurement algorithm to calculate a VAD than for the codec, since no realtime criteria have to be met. Likewise, the measurement algorithm has no restrictions in terms of latency, which means that it can analyse the entire signal before deciding which parts are active speech and which parts are silence. This method can also be used for real voice.

Drop outs

Dropouts are parts of the test signal, where the signal contains little or no energy, while there is energy in the reference signal. Dropouts are mostly caused by packet loss in IP networks, or severe RF problems in mobile networks. Using OPERA you can identify dropouts in the signal and analyse them graphically. Dropouts are shown in the same diagram as the VAD parameters.

6.6.3 Using PESQ

In general usage of the PESQ algorithm is exactly the same as using the PSQM algorithm. However there are a few significant differences:

- PESQ has no algorithm specific settings
- Since the ITU recommendation P.862 / PESQ is much more specific than PSQM was, and includes a really good time alignment algorithm, most of the settings for the signal preprocessing are disabled. OPERA uses exactly the parameters that are defined by the standard.

- Due to the complex and iterative time alignment algorithm, the data are all processed in one frame. There is no history through which the user can scroll for an analysis of the signals. The input data are processed as if they were exactly one large frame.

6.6.4 Diagram Types

Figure 6.21 shows the result diagrams available for PESQ. Our PESQ implementation offers much more then just voice quality testing. A partial side effect is a detailed analysis of VAD behaviour, jitter buffer adaptation or AGC/ALC tests.

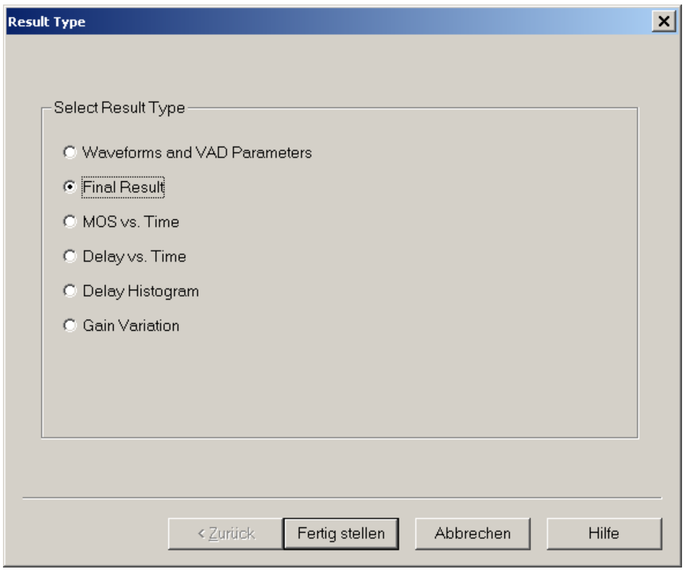


Figure 6.21: Result types available for PESQ

Waveforms and VAD Parameters

The "Waveforms and VAD Parameters" display of PESQ as shown in **Figure 6.22** differs significantly from the Timesignal displays of the other algorithms, since it shows the entire waveforms in one diagram instead of the framewise displays of the other algorithms. Within this diagram you can plot the following graphs:

- Waveform of the reference signal
- Waveform of the test signal
- Front End Clipping region as a red shaded area
- Hold Over Time region as a yellow shaded area
- Drop outs as an orange shaded area

All waveforms can be shown before or after the time and level alignment and in dBov as well as linear.

Additional information is shown on the right side of the diagram. This information includes the time and date of the measurement, as well as general information on the input data. The delay shown is the average delay in milliseconds as well as in samples. This delay is the average for the entire measurement period.

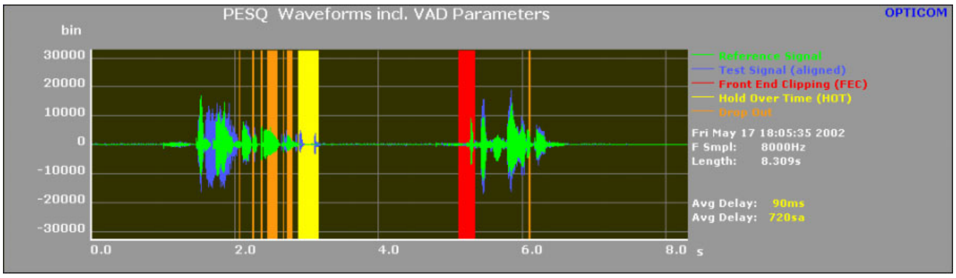


Figure 6.22: Waveform and VAD Parameters diagram. During this measurement severe distortions were detected (Drop outs and Front End Clipping)

Final Results

The screen below shows important results of the PESQ algorithm. It chiefly shows the MOS score, level measurement results and some additional information on the delay variation. The PESQ MOS as defined by the ITU recommendation P.862 ranges from 1.0 (worst) up to 4.5 (best). Below the PESQ MOS the same value is given mapped to:

- The full P.800 scale (1..5)
- To the PESQ-LQ scale

This view presents the final MOS score for the entire files, as well as the MOS score for Speech and silence separately. The resulting R factor according the e model is also shown here. Other values given are the minimum, average and maximum delay in milliseconds, as well as the delay jitter in milliseconds.

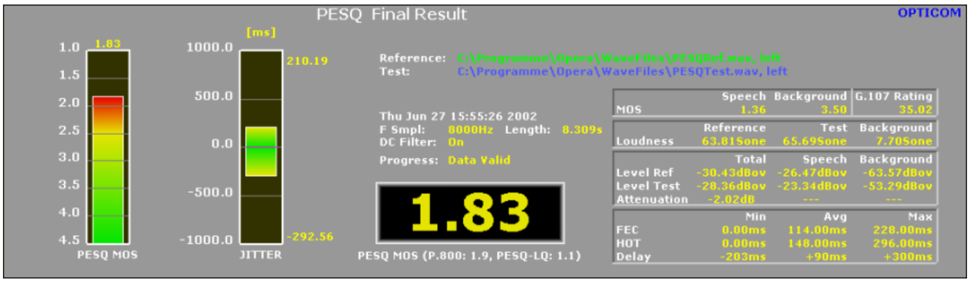


Figure 6.23: PESQ Final Result diagram

Delay Jitter

Using PESQ on OPERA, you can even analyze the behaviour of adaptive jitter buffers. Of course, OPERA can not look into the gateways, but the result of the jitter buffer adaptation can be observed as a delay jitter of the audio signal. The length of the jitter buffer adds linear to the delay of the speech signal. This means that a delay jump of 100ms is directly related to a jitter buffer adaptation of this amount (assuming that all the other latencies in the network are constant). Also you may observe that adaptations occur during active speech, which results in a worse MOS value. Jitter measurement may give you valuable information on how to optimize your network.

The delay jitter as measured by OPERA is defined as the maximum and minimum deviation of the delay from the average delay in ms as is shown in the final results window. An excerpt of this window can be seen below in **Figure 6.24**.

For a more detailed analysis of delay variations refer to the Delay vs. Time Diagram and the Delay Histogram.

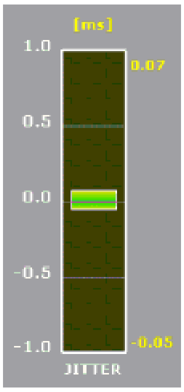


Figure 6.24: PESQ Delay Jitter display.

Levels

Information on the measured signal levels is also given in the Final results window. In the tables on the right side of the diagram the signal levels in dB_{ov} as well as the loudness in Sone are found.

MOS vs. Time

Figure 6.25 shows the MOS vs. Time diagram. This diagram indicates the perceived voice quality as measured by PESQ on a frame by frame basis. Use this diagram to analyze sequences that have spurious audible distortions. Search the peak in the MOS vs. Time diagram and then analyse the signals around this time stamp using the other provided diagrams.

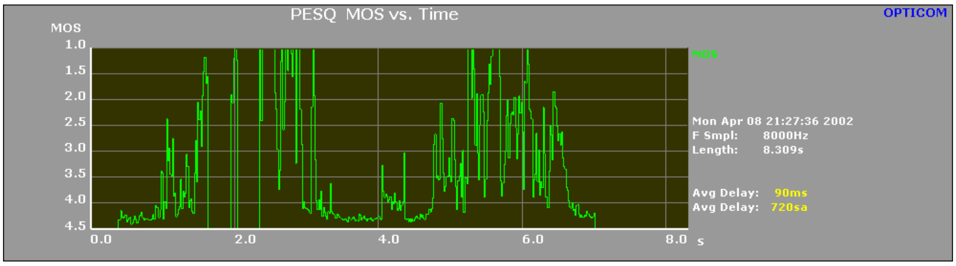


Figure 6.25: MOS vs. Time diagram.

Delay vs. Time

Figure 6.26 represents the Delay vs. Time diagram. It shows the time on the x-axis and the delay in ms on the y-axis. Especially with packet networks, significant variations of the delay during a call can be observed. This diagram is generally very useful for the analysis of jitter buffer adaptation algorithms, which are the main source for varying delays on IP networks. Additional information is shown on the right side of the diagram. This information includes the time and date of the measurement, as well as general information on the input data. The delay shown is the average delay in milliseconds as well as in samples. This average delay can also be plotted as a graph into the diagram. It is important to note that the average delay shown as a yellow line in **Figure 6.26** is a pure virtual number. It is the average delay over the entire measurement period. Since the delay in real systems usually varies in discrete steps only, this average delay probably never really occurs.

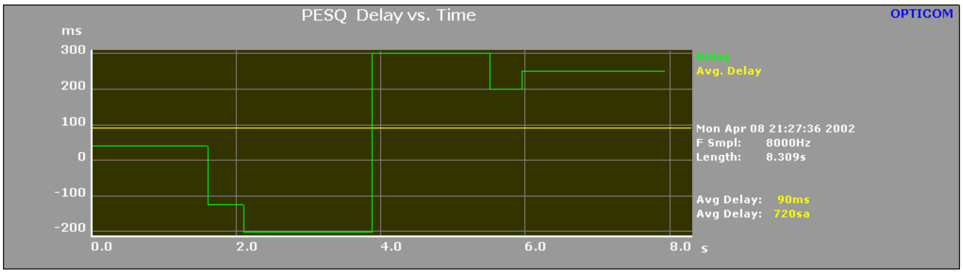


Figure 6.26: Delay vs. Time diagram

Delay Histogram

Figure 6.27 shows another view of the delay measurement. The delay histogram tells you the probability of each delay that actually occurred as the Probability Distribution Function (PDF). On the y-axis you see the probability in percent for each delay, whereas on the x-axis you find the delay in ms. The integral of this function is always 100%. Although the diagram looks like a bar chart, it is in fact the probability density function of the delay. The bar chart effect is due to the discrete delay values occurring in the measurement and the resolution of the algorithm. This diagram together with the Delay vs. Time diagram defines the full dynamic characteristic of the delay of your system under test.

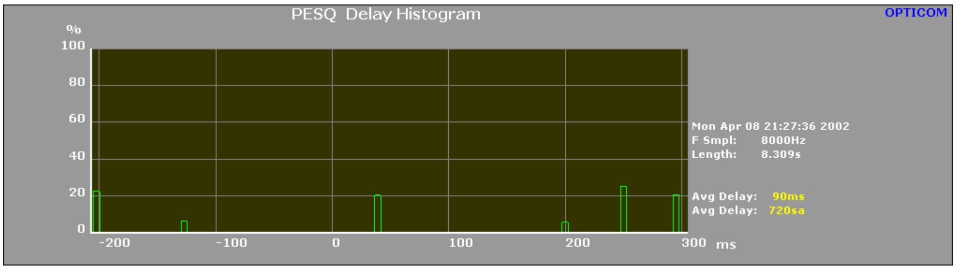


Figure 6.27: PESQ Delay Histogram

Gain Variation

The behaviour of e.g. AGC devices can best be measured using the Gain Variation diagram as shown in **Figure 6.28**. The y-axis of this diagram indicates the variable part of the gain in dB, which must be seen relative to the overall attenuation as shown in the Final Results diagram. The x-axis is the time in ms. Please note, that both signals must exceed the threshold in quiet by app. 7dB. A red line will be plotted for the invalid periods. All frames that do not meet this criterion will be set to 0dB.

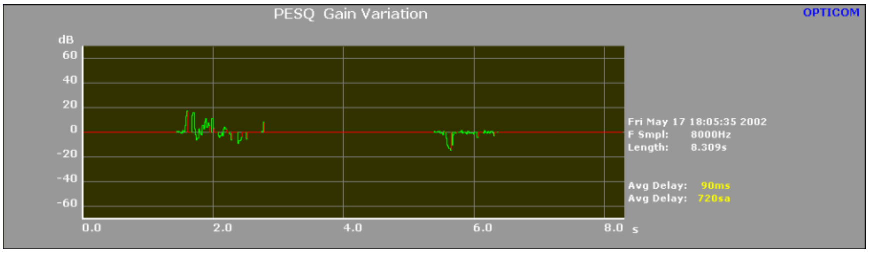


Figure 6.28: PESQ Gain Variation diagram

6.6.5 Command Line Arguments

PESQ currently requires no algorithm specific command line switches. To use PESQ from the commandline, OPERA must be started with the **-AlgorithmName=PESQ** switch.

6.7 Echo Measurement

6.7.1 Fundamentals of the Echo Measurement Algorithm

About Echo

In the Public Switched Telephonic Network (PSTN) the transit network is built with using four wires (see **Figure 6.29**). Two wires are used to transmit voice in one direction, and the two other wires are used for the opposite direction. In contrast, the subscriber network is built with using only two wires to transmit voice in both directions in a full-duplex connection to be cost effective.

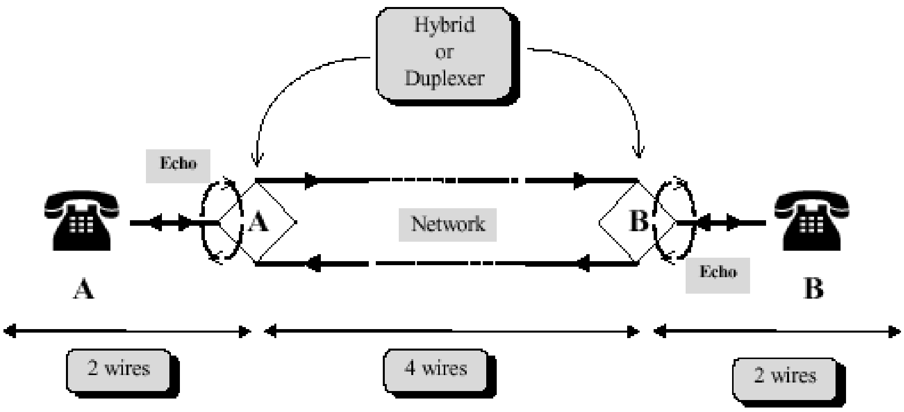


Figure 6.29: PSTN Transit Network [NMS99]

Because the length of the subscriber's line varies from 0 to 9 km (5.5 miles), the impedance of the line never matches the reference impedance of the hybrid. This non-balanced impedance generates echo on the hybrid.

When telephone set A is connected to telephone set B through the telephone network, telephone A will receive a near-end echo from hybrid A and an far-end echo from hybrid B.

If the length of the subscriber A line is up to 9 km (5.5 miles), delay will be less than 1 ms. Because the hybrid adds some distortion and multiple reflection, in general the width of the echo is up to 8 ms.

The delay of the far-end echo will depend on the call type (for example, local call, national call, international call). If the network is a full four-wire network only the far-end hybrid will generate the echo. The delay of this echo depends on the group delay of all devices used by the network.

If the network is still using two-wire devices to carry communications, additional echo will be generated [NMS99].

In VoIP networks there are many additional potential sources of echoes and because of the typically much longer network latencies, the echo delays will also be significantly longer. Multiple echoes are a very common problem in VoIP networks as well. With mobile networks the situation is even worse. Here the acoustical path between the speaker and the microphone of the mobile device is another potential source of echoes, especially with modern, very small phones.

The algorithm to calculate the echo on a telephone line uses real speech as the stimulus for the measurement. Consequently, the values obtained by this algorithm may not be compared to an Echo Return Loss ("ERL") measured according to ITU-T G.122.

On the other hand, the results achieved when an echo is measured with a speech signal may differ from those measured with a sine tone as a test stimulus. For most applications the echo of a real conversation is more interesting than the echo of a sine tone. The algorithm used to determine the echo is based on the long term correlation between the two signals. The values require some time after the start of the measurement until a steady state is reached.

6.7.2 Interpretation of Echo Parameters

Echoes shorter than app. 5ms are also known as **side tone** and not perceived as disturbing, they are even wanted and required to perceive a conversation as natural.

6.7.3 Signal Acquisition

Figure 6.30 clearly shows there are two kinds of sources that signals can be obtained from. In all OPERA™ versions files can be assessed. When working with the OPERA system version including a POTS telephony board or the audio interface option, OPTICOM's signal acquisition software **OptiCall™** can be used which is described in detail in chapter 4. Online measurements with phone lines are currently not supported.

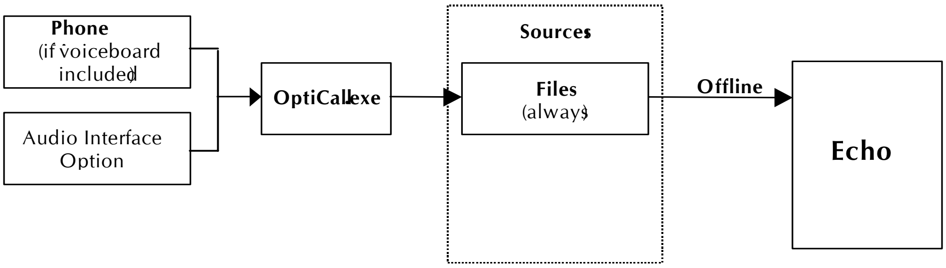


Figure 6.30: Kinds of signal sources for the Echo measurement

6.7.4 Echo Algorithm Properties

Figure 6.31 shows the algorithm properties of the echo algorithm. There is only one value, the maximum delay of the expected echo. This value may range from 10ms up to 1000ms. Choosing a lower value will lead to a faster calculation of the echo delay.

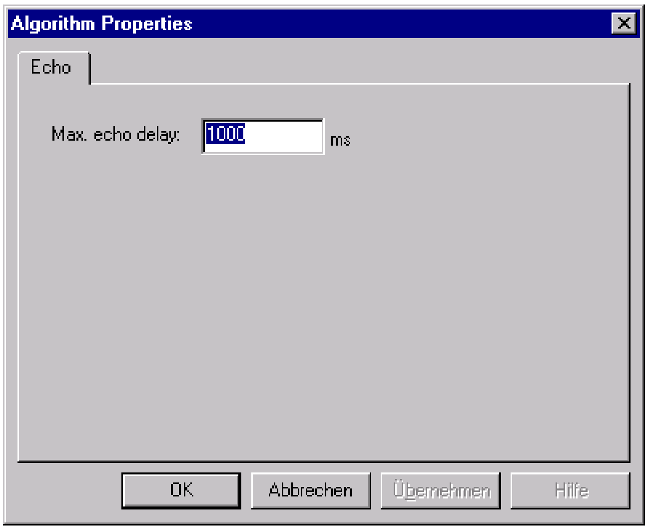


Figure 6.31: Echo algorithm properties

6.7.5 Specific Settings for the Echo Measurement

When performing echo measurements, the only settings available are "Invert test signals" and "Remove DC from signal". It is recommended to switch off the first and to switch on the second parameter, as shown in **Figure 6.32** All other parameters are disabled for the Echo algorithm.

Note:

Not all signal preprocessing options are available for the Echo algorithm. They must be set to fixed values in order to achieve proper measurement results and are therefore disabled.

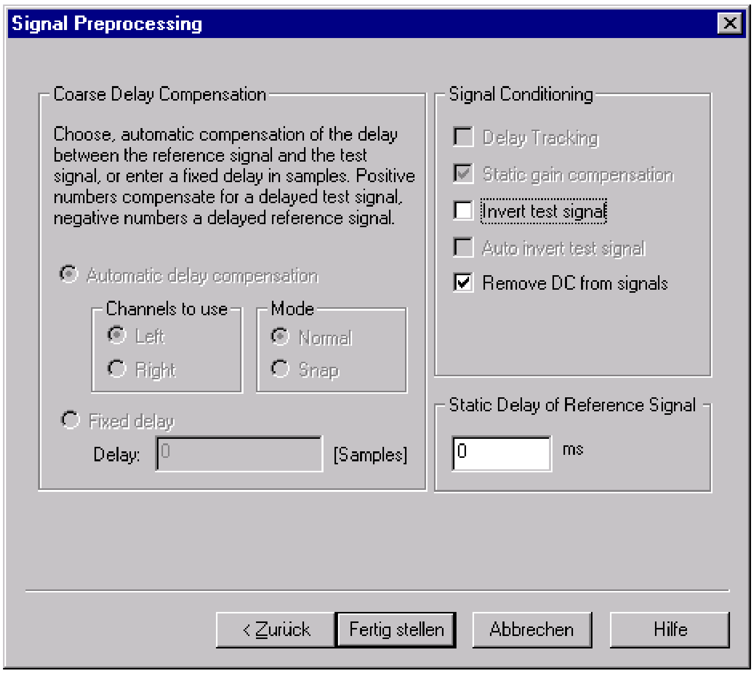


Figure 6.32: The signal preprocessing dialog

6.7.6 Diagram Types

Chapter 4 explained how to select a measurement algorithm and how to start a measurement. Once a measurement is performed results can be displayed . This section describes the three diagram types (**Figure 6.33**) available for the Echo algorithm.

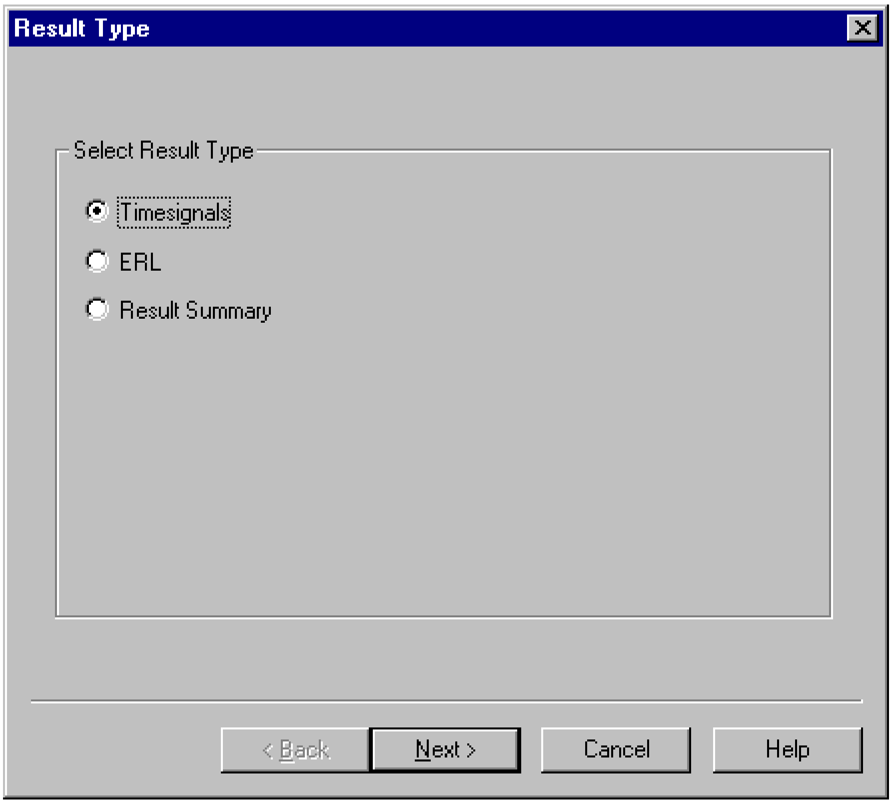


Figure 6.33: Result Type Window for the Echo Algorithm

Display of the Measurement settings

To the right of each diagram, some general information about the current measurement settings is shown in a text block (see **Figure 6.34**). The meaning of the values is as shown in **Table 6.6**.

Displayed Values	Interpretation
Time:	The time when the measurement has been finished
F Smpl:	Sample rate of input signals
Tracking:	Status of the delay tracking (on or off)
DC Filter:	Status of the DC filter (on or off)
Delay:	Delay in ms (first from top) as well as in samples (second from top)
Status:	Reliability of the automatic delay compensation (0..100%, Fixed=fixed delay set).
Atten:	Level difference between reference and test signal (dB)
Rel Time:	Current point of time in the measurement

Table 6.6: Interpretation of the displayed values

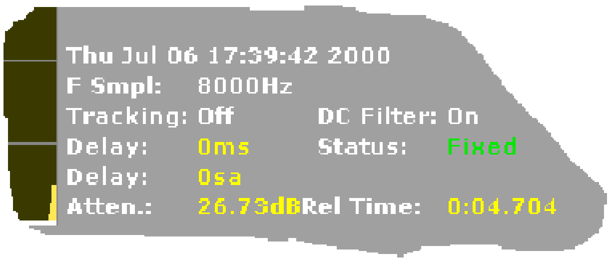


Figure 6.34: Display of the current measurement settings of the Echo algorithm

Timesignals

To choose this diagram type highlight the radio button next to **Timesignals**, and press **Next**. This leads you to the next wizard step, the **Signal Select** dialog.

The "**Signal Select**" dialog (see **Figure 6.35**) allows selection of the channels and input signals for which the results in one diagram shall be plotted. The selection may be modified by clicking with the left mouse button on any of the option buttons. This will add or remove the check mark in the button. A checked button means the results for the selected signal will be drawn in the diagram. In **Figure 6.35** the results for the left channel of the reference and the test signal were selected. Each signal will be drawn in a different colour. The assignment of the colors can be seen in the field to the right of the diagram panel.

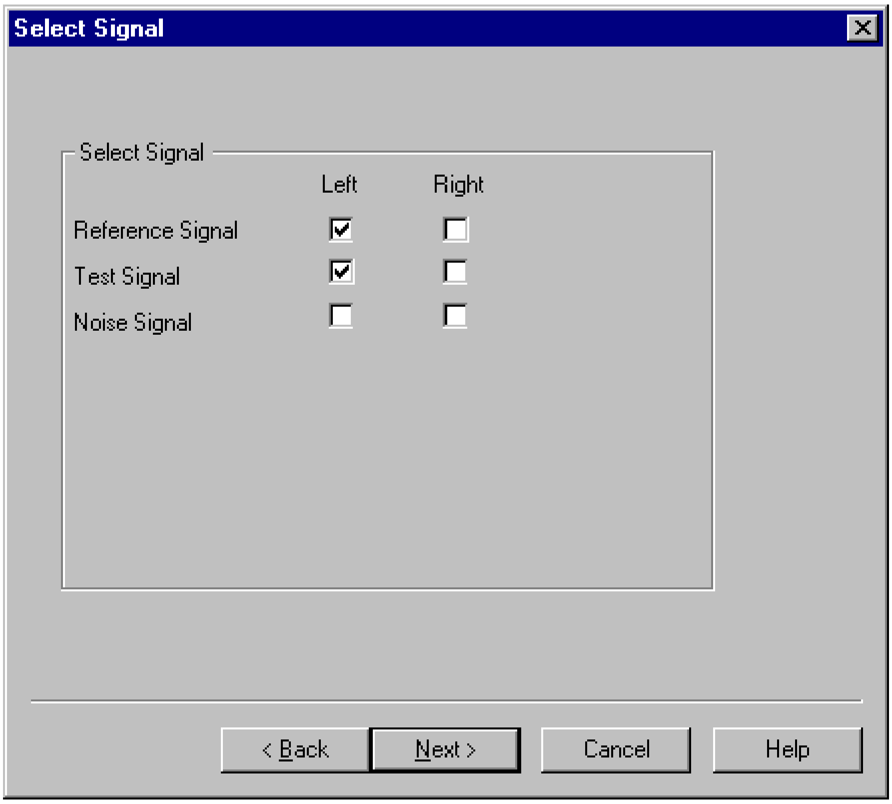


Figure 6.35: Select Signal Window

Pressing **Next** again leads to the next step, the **Result Style** dialog (see **Figure 6.36**). Here the way data is shown on the screen is selected. Usually this is identical to selecting the units of the diagram axes. For time signals choose between a binary, linear representation, in which the input signals are always scaled to **[-32768 ... +32767]**, or a **dB FS** (full scale) scale.

Note:

Independent of the input data format, samples are always converted to **16bit/sample**. This means that 8bit/sample data are multiplied by 256 before they are processed any further.

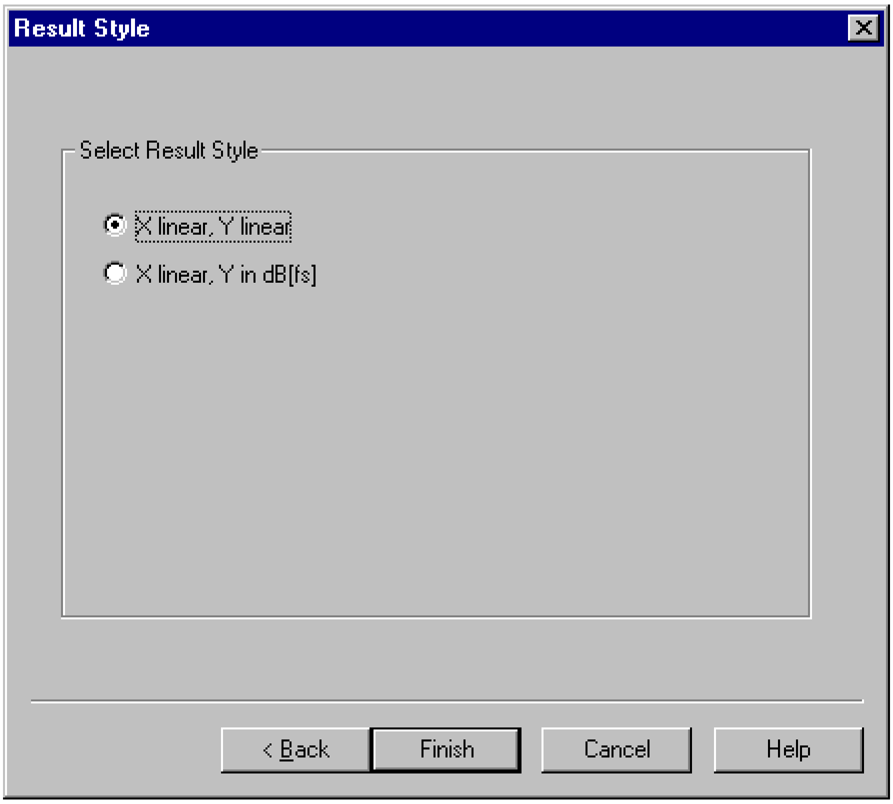


Figure 6.36: Result Style Window

After this last step you may click on **Finish** and the selected diagram will appear in the diagram pane as shown in **Figure 6.37**. An excerpt of the time signal of one frame can be seen.

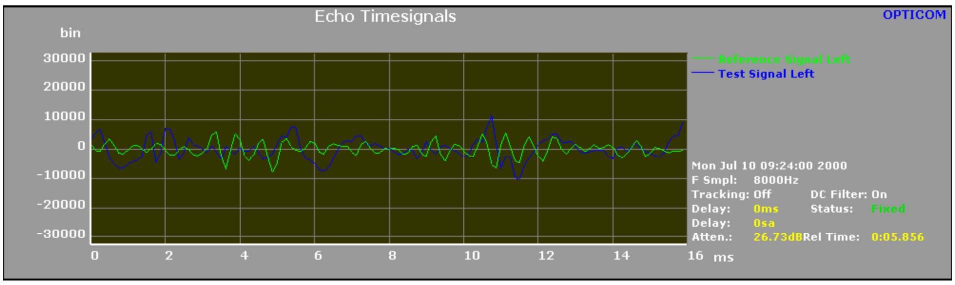


Figure 6.37: Time signals diagram

ERL (Echo Return Loss)

Figure 6.38 shows the diagram display when you choose the ERL diagram type. The horizontal axis is a time axis for the amount of the delay, the vertical axis shows the attenuation of the echo signal given in dB. Optionally you can overlay the diagram with the acceptable tolerance curve according to ITU-T Rec.G.131. Please note that the G.131 defines this curve for the one way delay, whereas OPERA measures the real echo delay, which corresponds to the two way delay of the ITU recommendation. Please be also aware, that G.131 is defined for simple artificial test signals only and does not take into account multiple echoes.

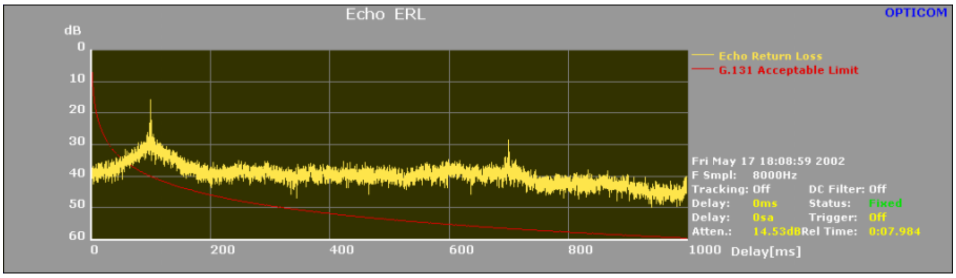


Figure 6.38: Display of the ERL diagram type

Result Summary

When choosing this diagram type, the diagram will display several result values of the echo measurement, as shown in **Figure 6.39**.

This diagram type distinguishes between the averaging over the time of one window (momentary - "mom") and the averaging over the time of the whole measurement ("peak"). See **Table 6.7** for an overview of the displayed result values.

Result Values	Interpretation
Min.Delay:	Echos with values smaller than this value will not be regarded as echo
Frame Size:	Size of one frame
ERLmom:	Attenuation of the echo in the momentary window with least attenuation
ERLmom Delay:	Delay time of the echo in the momentary window with least attenuation
ERLmom Window:	Window size
ERLpeak:	Attenuation of the echo with least attenuation. Averaged over the time of the whole measurement
ERLpeak Delay:	Delay time of the echo with least attenuation. Averaged over the time of the whole measurement
ERLpeak Window:	Window size

Table 6.7: Interpretation of the displayed result values

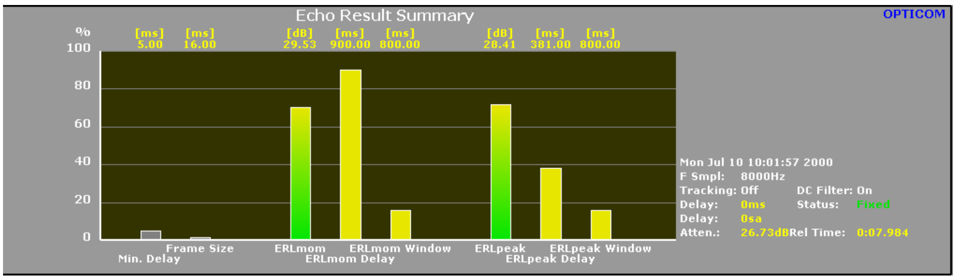


Figure 6.39: Result summary diagram type

6.7.7 Command Line Arguments

In addition to the command line arguments described in Chapter 4 there is one specific command for the Echo algorithm. This command is to be used together with the **-Algorithm Name=Echo** switch:

MaxDelay <maximum delay in integer> ;maximum delay of echo measured

6.8 Measurement Examples

For an inexperienced user of the OPERA™ measurement system, the following examples might be useful to assist in your first measurements. The first example demonstrates a typical loop measurement, while Example 2 will show a typical application of measurements from a batch file.

6.8.1 Example 1: Stand Alone Loop Measurement

6.1.1.2 Signal Acquisition With OptiCall™

After connecting the OPERA™ system to the telephone network to be measured, start the OptiCall™ program from the start menu **Start|Programs|Opera|OptiCall** (or click on the icon on the desktop). In the OptiCall dialog, select the loop mode button. Make a right mouse button click into the title bar of OptiCall and choose "Telephony Standard View" from the drop down menu. Enter the phone number that is associated to line 0. Select the file that contains a speech sample that will be sent through the network, in the field **Reference**. In this example the WAVE file that is located in the WaveFiles subfolder of the installation directory of your OPERA™ system is used, "DefaultRefFile.wav".

Finally, enter the destination directory where the test files will be saved. Also enter the root file name of your test files after the character #, e.g. "D:\test#050700. Now press the start button. The connection will be established, the speech sample will be sent and the test samples will be saved in the location we have defined above.

> **Note:**
>
> The WAVE file "PSQMRef.wav" located in the OPERA™ installation directory should **not** be used with OptiCall™. This file contains data that is 16 bit linear PCM coded. Since the POTS interfaces would convert this to G.711, there would be an additional, unwanted impairment of the test signal, caused by the system.

When the test call has terminated two test files will be stored in the specified destination directory. Assuming you that OptiCall was used in the Telephony Standard mode, "050700-Line0.wav" is the relevant file when using **PSQM**, or **PESQ**, "050700-Line1.wav" is the relevant file if an analysis of **Echo** shall be performed with OPERA™. If OptCall was used in the Expert mode, take care to choose the file recorded at the transmitting side for the Echo measurement and the file recorded at the side which was listening only for the **PSQM** or **PESQ** measurement

Using the PSQM Algorithm

At this time our measurement can be analyzed. Close OptiCall™ and bring OPERA™ into the foreground again by clicking on it. If OPERA is not yet open, choose **Start|Programs|Opera|Opera.** First use the PSQM algorithm. For this purpose, select from the menu **Masurement|Algorithm Parameters ...** the **PSQM** algorithm. To define the available properties click on the **Properties...** button. Choose the settings shown in **Figure 6.40**, i.e. the IRS (telephone band) filter characteristic, Hoth noise of 45 dB(SPL), a listening level of 101 dB(SPL) and an upper frequency of 4 kHz.

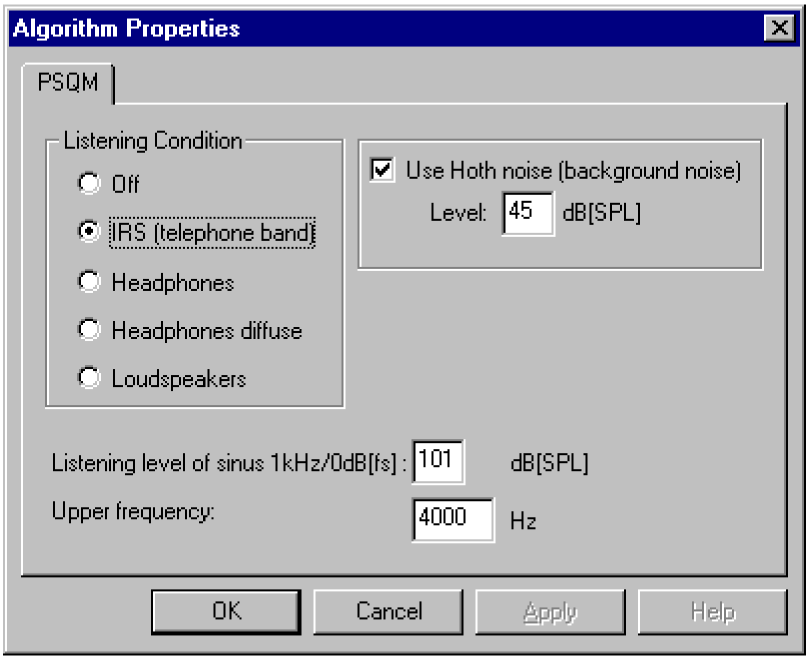


Figure 6.40: Algorithm Properties dialog for PSQM

Click on the OK button and close the Algorithm Properties dialog. Take notice of the warning message and click on the **Continue** button.

Now start the analysis by clicking on the toolbar button shown in **Figure 6.41** for example. The Measurement Setup Wizard will start.

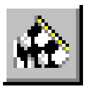


Figure 6.41: Toolbar button for starting the measurement

Choose the speech sample as the first input signal sent through your network when using the OptiCall™ program (see **Figure 6.42**). In this case "C:\Programme\Opera\DefaultRefFile.wav" Click on the **Next** button to select the second input signal.

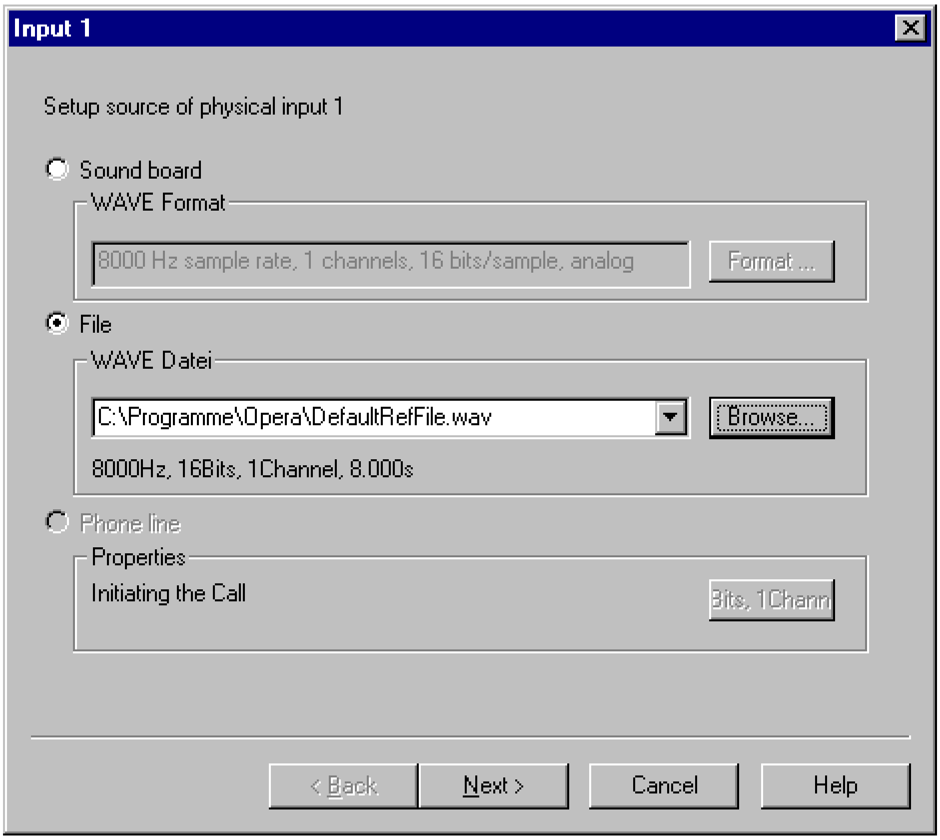


Figure 6.42: First step of the Measurement Setup Wizard

As the second input signal, select the stored file whose filename contains "-Line0", in this example "D:\test\050700-Line0.wav" (see **Figure 6.43**). When clicking on the **Next** button the Input Mapping dialog will appear. Here, the left signal of Input 1 as the left channel of the reference signal and the left signal of Input 2 as the left channel of the Test signal are defined. Only the buttons for the left channels are available since mono signals are assessed. Mono signals are always treated as "left channel only".

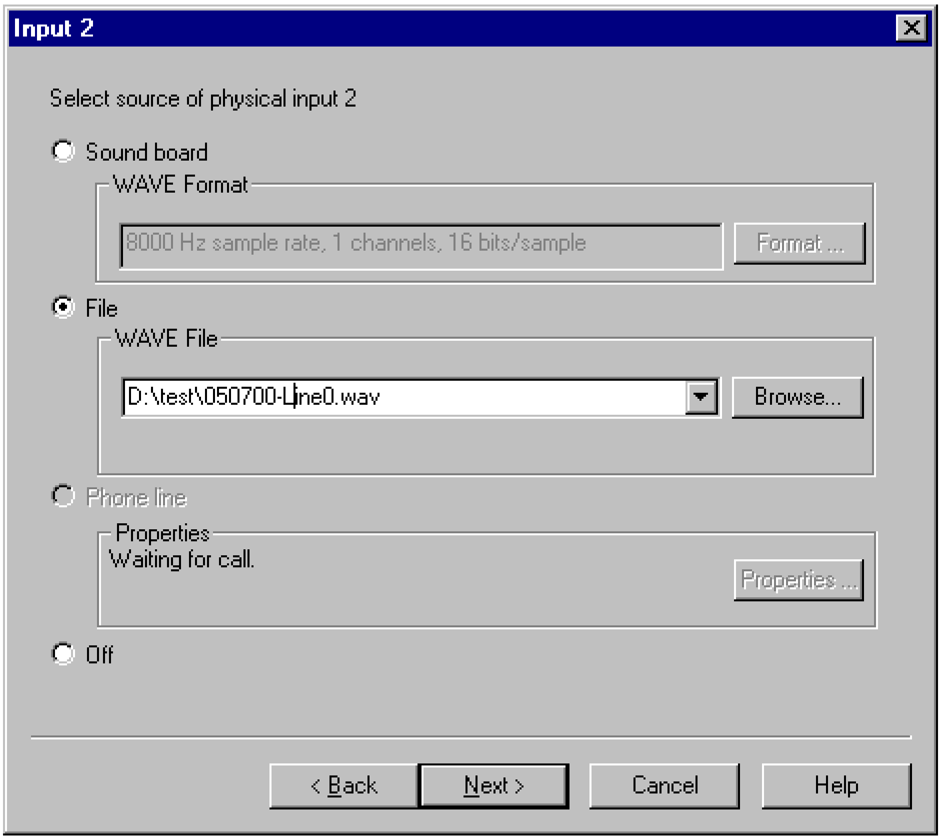


Figure 6.43: Second step of the Measurement Setup Wizard

After clicking on the Next button, we will get to the Signal Preprocessing dialog where the **Automatic Delay Compensation** function, the **Static Gain Compensation** function and the **Remove DC from Signals** option (see **Figure 6.44**) are selected. Finally click the **Finish** button to start the actual analysis.

After the computation of the delay has been finished, select the diagram types for display, for example **Final Results** for the upper diagram and **MOS vs. Time** for the lower diagram panel. Depending on the reference signal and the resulting test signal being used, the view on the measurement will look similar to the view depicted in **Figure 6.45**.

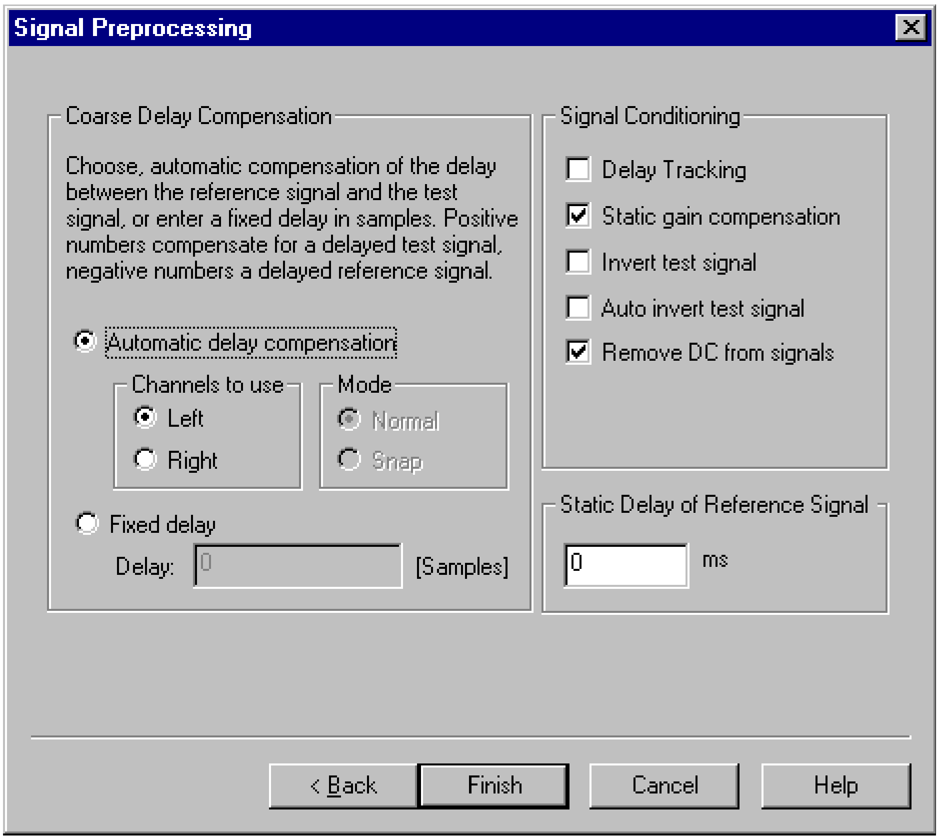


Figure 6.44: Signal Preprocessing dialog of the Measurement Setup Wizard

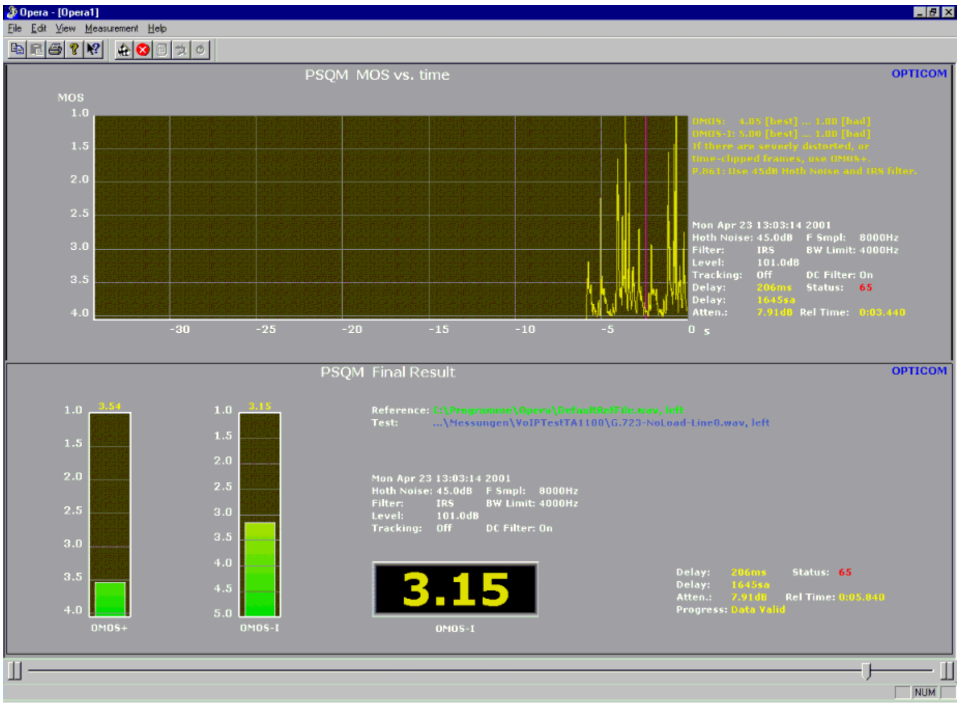


Figure 6.45: Resulting view on the measurement results

Using the Echo Algorithm

At this time an echo measurement will be performed. Therefore the **Echo** algorithm from the Menu **Measurement|Algorithm Parameters ...** is selected In the **Properties** dialog choose 1000 ms is chosen as the maximum echo delay, as shown in **Figure 6.46**. Click the **OK** button, close the Algorithm dialog and take notice of the warning message and press **Continue**.

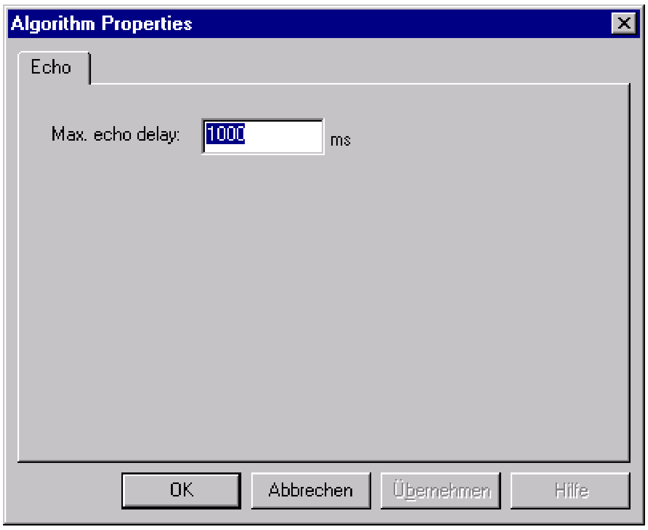


Figure 6.46: Algorithm Properties dialog for the Echo algorithm

After clicking on the **Start** toolbar button or selecting the menu option **Measurement|Start** the Measurement Setup Wizard starts and the sent speech sample file is chosen used during the OptiCall™ procedure as the first input signal (see **Figure 6.47**), in this example "DefaultRefFile.wav". As the second input signal the stored file is selected whose name ends with the string "Line1". In this example the file "**050700-Line1.wav**" is selected (see **Figure 6.48**).

When clicking on the **Next** button the Signal Mapping Dialog will come up. Here the left channel of Input 1 is assigned to the left channel of the Reference Signal and the left channel of Input 2 to the left channel of the Test Signal.

Clicking on the Next button the Signal Preprocessing Dialog is reached where the following settings are selected: Click on the **Fixed Delay** Radio Button and leave the delay setting of 0 samples. In the field "Signal Conditioning" only the **Remove DC from Signals** function is selected. No Static Delay of the Reference Signal is entered. **Figure 6.49** shows the settings proposed for this example.

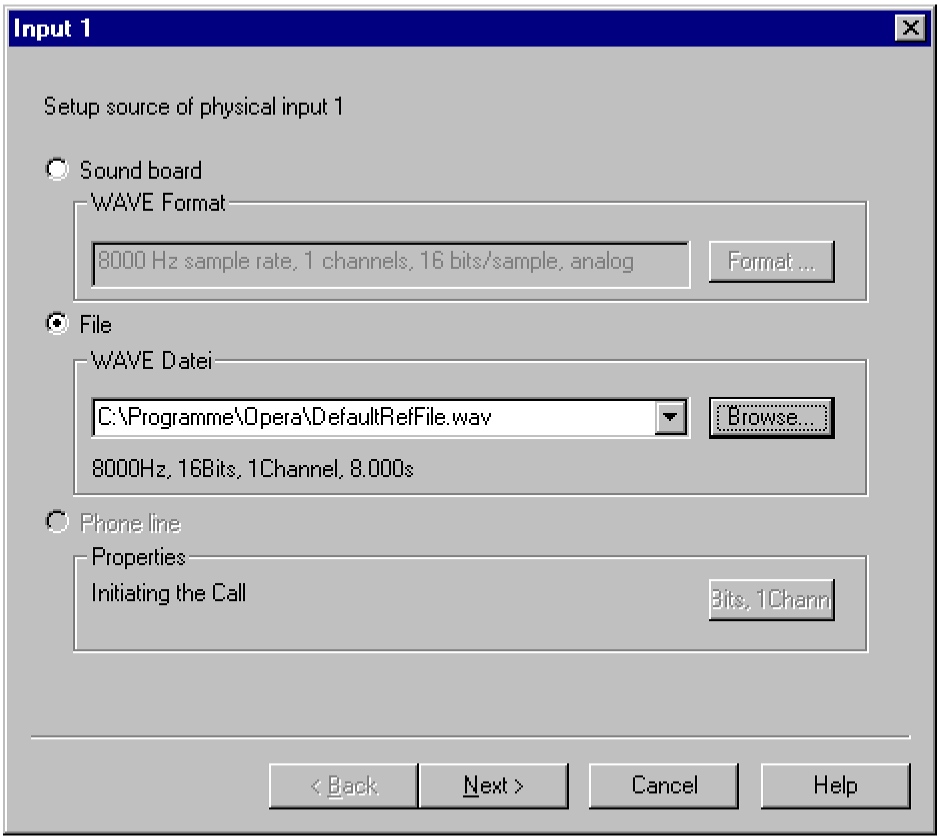


Figure 6.47: First step of the Measurement Setup Wizard

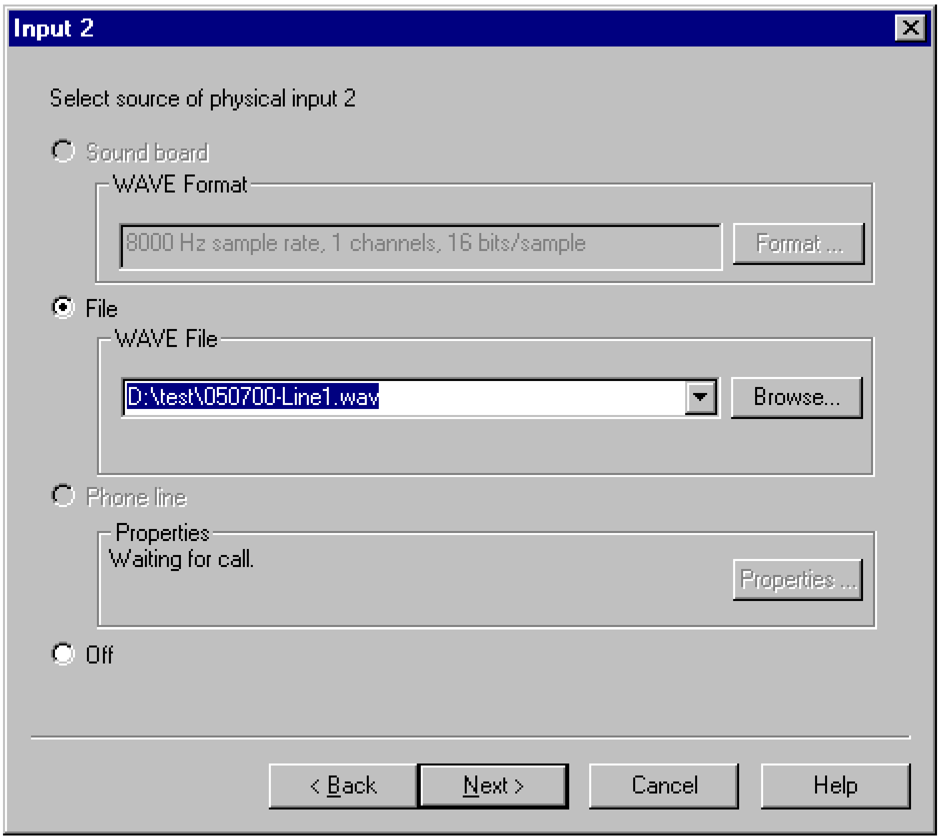


Figure 6.48: Second step of the Measurement Setup Wizard

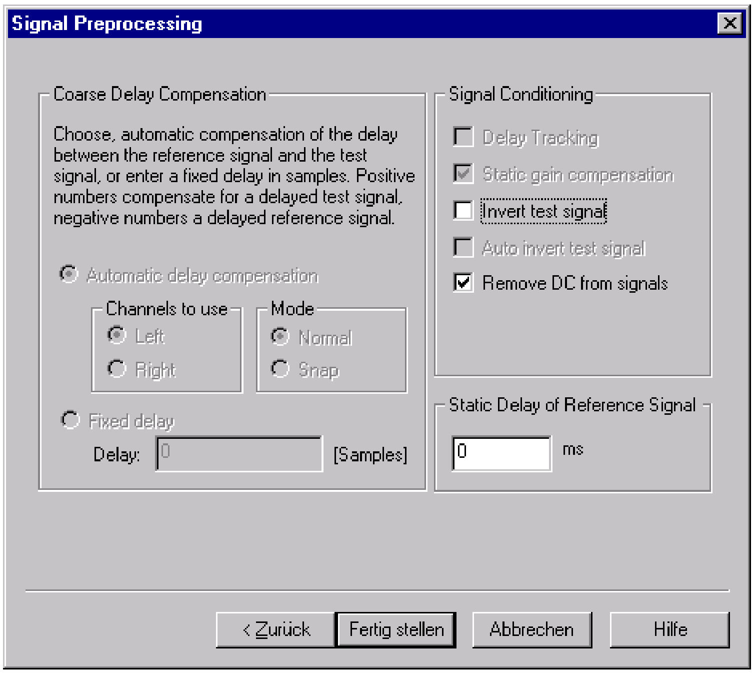


Figure 6.49: Settings in the Signal Preprocessing Dialog

When clicking on the Finish button in the Signal Preprocessing Dialog, the echo is measured. After this process has been completed, the diagram types to display can be chosen. In this example the Result Summary type for the upper diagram pane and the ERL type for the lower diagram are selected. Thus you will get a view similar to the one shown in **Figure 6.50**.

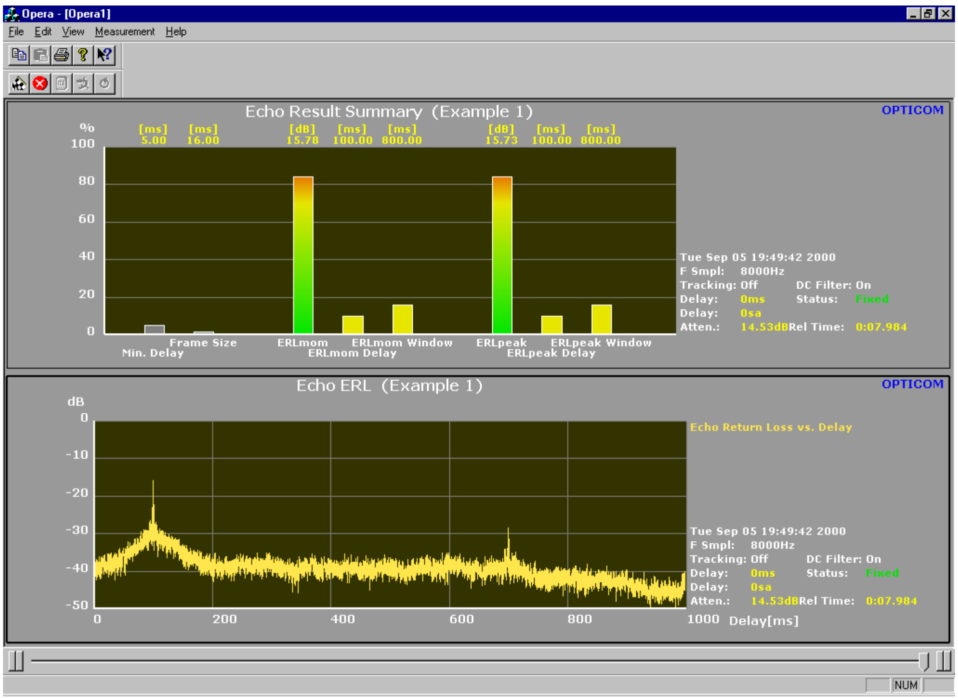


Figure 6.50: Resulting view on the Echo measurement

6.8.2 Example 2: Measurements From a Batch File

For this example the batch file **RunPSQM.bat** contained in the installation directory of your OPERA™ system is used. The proposed files for assessment in this example are located in the same directory. The names of these files are "PSQMRef.wav" and "PSQMTest.wav". The following lines show the contents of **RunPSQM.bat** . Have a look at those lines and read the explanation that will follow after the file excerpt.

RunPSQM.bat

```
@echo off
rem batch file to compute PSQM from two mono input files
rem
rem   Parameters:
rem
rem   RunPsqm   <File1> <File2> <Outputfile>
rem
rem          File1:   File that contains the reference signal
rem
rem          File2:   File that contains the test signal.
rem
rem          Outputfile: Results are stored in this file. If
rem                      it already exists, results are
rem                      appended to it, otherwise it will be
rem                      newly created.
rem
rem
echo **************************************************
echo ***** RunPSQM        V2.0  (c) OPTICOM, 2000   ***
echo **************************************************
echo.
pushd
echo *** TODO: change working dir according to where
echo     OPERA.exe is!
echo ... Processing file %1
Opera -Exec -Algorithm Name=PSQM -Input Inp=0 File=%1 Inp=1

   File=%2 -Mux InpRefLeft=0 ChannelRefLeft=0 InpTestLeft=1

   ChannelTestLeft=0 -Delay Channel=0 -Signal StaticGainOn

   DCFilterOn -Out %3 -Append
popd
echo Done!
```

To run this batch file, please open an MS-DOS Command Prompt window. Start the batch file by entering following syntax: "**RunPQSM <File1> <File2> <Outputfile>**". This batch file has three parameters, the first one is the filename and the path of the reference signal, e.g. "C:\Programme\Opera\WaveFiles\PSQMRef.wav". The second parameter is the test file, e.g. "C:\Programme\Opera\WaveFiles\PSQMTest.wav". Both of these WAVE files are located in the WaveFiles subdirectory of the OPERA™ installation folder on the harddisk.

The third and last parameter is the path with the name of the output file that will contain the result values, e.g. "C:\Temp\PSQMresult.txt".

In the next lines of the batch file OPERA™ is executed. Please note that the parameters that follow the command "opera" must be written into one line. Here, those lines had to be wrapped in the absence of space in this manual.

First, there are no PSQM algorithm properties defined. This means that standard values are used as recommended by ITU-T P.861. The recommended values are:

- Listening Condition = IRS
- Hoth noise of 45 dB(SPL)
- Listening level of 101 dB(SPL)
- Upper frequency of 4 kHz..

After the **Opera –Exec** command following definitions are made: the PSQM algorithm is selected and the input signals are defined, i.e. the first parameter of the batch file is defined as Input 1, the second parameter as Input 2.

The signals are mapped: the left channel of input 1 is defined as the left channel of the reference signal, the left channel of input 2 is defined as the left channel of the test signal.

Then some settings that correspond to the signal preprocessing dialog are made. First the Automatic delay compensation is switched on. The left channel is used for this function and it is run in normal mode. In the Command Line Parameters functionality there is only Normal Mode available, since the Snap Mode is only useful for online measurements. However, online measurements cannot be performed from a batch file.

The Static Gain Compensation function and the DC filter are switched on.

Finally, the third parameter of the batch file is defined as the output file. This file contains the result values and the current measurement settings of the PSQM measurement. The contained values will be tab-separated and can be imported into any spreadsheet analysis program.

When the batch file has run successfully, have a look at the output file you have specified. When the WAVE files have been used as proposed in this section – **PSQMRef.wav** and **PSQMTest.wav** – the measurement results will correspond with the following data:

- OMOS-W0 = 3.35
- OMOS-W2 = 3.46
- OMOS-W4 = 3.30
- OMOS+ = 3.60

6.8.3 More Examples ...

More example setups, especially some more exotic applications can be found in our paper "OPERA Application Notes" which is attached to this manual. Carefully look at the paper since some of the problems discussed there may well be transferable to your own application.

Chapter
7

7 AUTOMATION AND PROGRAMMING

7.1 General

This is primarily a summary of the previous chapters on using OPERA from the commandline. To support users with automation of their measurements we have collected all relevant scripting information in this chapter.

7.2 Performing Measurements From Batch Files

When performing measurements on a huge number of input files it is much easier to start your OPERA™ system from a batch file. OPERA™ then will process all files in a row, which will save considerable time in comparison to performing the same task manually. To support this, OPERA™ understands a number of command line parameters. These parameters may also be written into a batch file and contain comments. For information about the syntax of batch files, please refer to the corresponding help topic in your Windows help.

7.2.1 Syntax of the Command Line Parameters

To start an OPERA™ measurement with command line arguments use the following syntax:

`opera -Exec` <list of parameters>

Every measurement requires these keywords to be typed at the beginning of the line. The list of parameters comprises some of the parameters that are described in this section. All parameters have to be put on a single line for each measurement. Alternatively the list of parameters can also be placed in a configuration file. In this case start OPERA like:

`opera -Exec –Cfg` <name of the configuration file>

For a description on how to use configuration files see paragraph 7.2.3.

7.2.2 Parameters common to all Algorithms

Below the syntax of the command line parameters is described. For all parameters inside a command section (i.e. –Input or –Mux) the section identifier (e.g. "-Input") has to be typed only once, at the beginning.

Section Identifier	Option	Parameter	Description
-Algorithm			
	Name	<PSQM | PESQ | ECHO | PEAQ>	Name of the algorithm to be used
	Settings	"more parameters"	Parameters that algorithm specific. See the algorithms description for details .Note that the parameters must be enclosed in qotes!
-Input			
	Inp=0 File="File1"		File name used for input 1
	Inp=1 File="File2"		File name used for input 2
-Mux			
	InpRefLeft	<0 | 1>	Input used to form the left channel of the refernce signal
	InpRefRight	<0 | 1>	Input used to form the right channel of the reference signal
	InpTestLeft	<0 | 1>	Input used to form the left channel of the test signal
	InpTestRight	<0 | 1>	Input used to form the right channel of the test signal
	ChannelRefLeft	<0 for left | 1 for right>	Channel of input signal used to form the left channel of the refernce signal (0=left, 1=right)
	ChannelRefRight	<0 for left | 1 for right>	Channel of input signal used to form the right channel of the refernce signal (0=left, 1=right)
	ChannelTestLeft	<0 for left | 1 for right>	Channel of input signal used to form the left channel of the test signal (0=left, 1=right)
	ChannelTestRight	<0 for left | 1 for right>	Channel of input signal used to form the right channel of the test signal (0=left, 1=right)
-Delay			
	FixedDelay		Use a fixed delay for the measurement
	Delay	<delay>	Specify the fixed delay in samples
	TrackingOn		Switch delay tracking on
	Channel	<0 for left | 1 for right>	Channel used for the automatic delay compensation
	StaticDelay	<Delay in ms>	Additional static delay of the reference signal in ms

-Signal			
	StaticGainOn		Switch the static gain compensation on
	InvertTestSignal		Invert the test signal
	AutoInvertTestSig		Automatically invert the test signal
	DCFilterOn		Switch DC filtering on
-Trigger			
	StartTime	<Start time>	Specify start point of the measurement in ms
	EndTime	<End time>	Specify end point of the measurement in ms
	Channel	<0: Relate to reefernce | 1: realte to test>	Relate start and end point to the beginning of the refernce or the test signal
-Out		"<FileName>"	Name and path of result output file
-Append			Append results to existing result output file
–PassThrough		"<Additional Text>"	The additional text will be printed to the result file
-Cfg		"<File name>"	Name and path of a configuration file that contains more command line parameters

7.2.3 How to Use a Configuration File

A configuration file can be created containing e.g. frequently used settings of parameters. Inside the configuration file all parameters listed above are allowed. All parameters may be placed on separate lines. However between a command line option and the associated parameters no line break is allowed.Configuration files are best created using a text editor like notepad. To insert comment lines into the configuration file, a ';' must be put at the beginning of the line. When starting OPERA™ with a configuration file use the following syntax:

```
opera -Exec -Cfg <Name and path of the configuration file>
```

The configuration file must have the suffix ".cfg".

A combination of both versions may also be used:

```
opera -Exec -Cfg DefaultPara.cfg -Input Inp=0
                 File=InputFile1.wav
```

Here the corresponding setting made in DefaultPara.cfg is overwritten by the **-Input** command.

> **Note:**
>
> The command line parameters in the batch and configuration files are case sensitive.

7.2.4 Parameters Specific to the Measurement Algorithms

Parameters Specific to the PEAQ Algorithm

The current PEAQ implementation provides several algorithm specific command line parameters to:

- Set the version of the algorithm (basic, advanced)
- Set the listening level
- Set the logging of the results

These parameters essentially follow the settings of the algorithm parameters dialog and are listed with a short comment on their usage in the following:

Keyword	Add. Parameter	Comment
Version	0 = Basic; 1 = Advanced	Select the Version of the Algorithm
Level	Listening Level of a 1kHz 0dBfs sine tone	Set the listening level according to BS.1387
LogActive		Switch logging on
LogODG	float	Logging if ODG < = float
LogInterval	duration	Logging intervals in s
LogFileName	FileName	Name of the logfile

Parameters specific to the PSQM Algorithm

PSQM currently interprets the following algorithm specific command line switches. If no switches are specified, the default settings for correct measurements according to P.861 will be chosen (45dB Hoth noise, 4kHz upper limit, IRS filer, 101dBSPL listening level).

These commands are to be used together with the **-Algorithm Name=PSQM** switch:

Keyword	Add. Parameter	Comment
HothNoise		Use background masking noise
HothNoiseLevel	Level of Hoth noise in dBSPL	Level of background masking noise
ListeningLevel	Level of a 1kHz 0dBov sine tone in dBSPL	Listening level acc. to P.861
UpperFreq	Frequency in Hz	specify upper frequency limit for measuring
Flat		Listening condition: Flat frequency response
IRS		Listening condition: IRS (telephone) filtering

Headphones		Listening condition: Headphones
HeadphonesDiff		Listening condition: Headphones (diffuse field)
Speakers		Listening condition: Loudspeakers

Parameters Specific to the PESQ Algorithm

No Parameters are currently required for the PESQ algorithm.

Parameters Specific to the Echo Algorithm

The only parameter available for the echo algorithm defines the longest echo delay that can be measured.

MaxDelay <100...1000> ;maximum delay of echo measured in ms

7.2.5 Parameters Specific to OptiCall

To allow for automated execution from scripts, **OptiCall™** can also be started from a DOS window. It understands the following parameters:

/Exec		This must always be the first parameter!
-Loop		Perform a loop call
-Termination		Terminate the call
-Origin		Originate the call
-Cfg	<file name>	Read more parameters from configuration file
-Phonenumber	<phone number>	Phone number to dial
-RefFileOrigin	<file name>	Play file used on calling side
-RefFileTermination	<file name>	Play file used on terminating side
-DestinationPath	<drive:\\path | UNC path>	Destination directory for recorded files
-RootFilename	<root file name>	Root file name used for recorded files
-DoubleTalk		Let both sides of the call talk simultaneously. By default only the terminating side is talking.
-Mirror	<offset>	The call will be terminated on the originating interface plus *offset*.
-NumberOfCalls	<n>	Perform *n* calls
-Bulk	<k>	Perform the call on *k* consecutive interfaces simultaneously.
-NumRecordings	<j>	Perform the data acquisition *j* times during one call.

-DelayBetweenRecordings	<xxx>	Wait *xxx* seconds between two data acquisition phases during one call. Must be used together with *–NumRecordings*.
-RecordGainOrigin	<xxx>	Amplify the signal recorded at the originating side by *xxx* dB.
-RecordGainTermination	<xxx>	Amplify the signal recorded at the terminating side by *xxx* dB.
-Quiet		Suppress output to stdout
-OriginatingLine	<0|..N>	Index of the calling interface
-TerminatingLine	<0|..N>	Index of the terminating interface
-Player	<Bitmask>	The bitmask defines which interface is sending (playing) the file. Bit 0 is the originating interface and bit 1 is the terminating interface. Enter 3 for both interfaces.
-Recorder	<Bitmask>	The bitmask defines which interface is receiving (recording) the file. Bit 0 is the originating interface and bit 1 is the terminating interface. Enter 3 for both interfaces.
-Host	<hostname>	Name of the OPERA system on which the program should execute.This parameter is subject to a special network license!
-ListDevices		List all interfaces available for test calls. May be used together with –Host.

- Both play files should be of approximately the same duration (±0.25s).
- To find the proper index of an interface, open **OptiCall™** in the GUI mode, and click on the drop down list box as if to change the terminating or the originating interface. Find that all entries in the list box start with a number. This number is the index of the line. Alternatively OptiCall can be started with the options /Exec –ListDevices. It will then print a list of all available devices on the screen (this option can also be used together with –Host to list the interfaces of a remote machine.).
- If either the terminating or the originating side is an audio interface and not a telephony device, the timing printed to stdout as a result of the call is

meaningless. It is only provided in order to maintain compatibility between scripts.

Example (should be written on one line):

- Make a loop call from line 0 to Line 1, dial 01234, use the default reference file and store the results in C:\\temp, as Test-line0 (file with degraded signal) and Test-line1 (echo signal).

```
Opticall /Exec -Loop -OriginatingLine 0 -TerminatingLine 1 -
Phonenumber 01234 -RefFileOrigin
C:\programme\opera\WaveFiles\DefaultReffile.wav -
RefFileTermination
C:\programme\opera\WaveFiles\DefaultReffile.wav -
DestinationPath c:\temp -RootFilename Test
```

7.2.6 Example RunPsqm.bat

```
@echo off
rem batch file to compute PSQM from two stereo input files
rem
rem Parameters:
rem
rem   RunPsqm   <File1> <File2> <Outputfile>
rem
rem         File1:   File that contains the reference signal
rem
rem         File2:   File that contains the test signal.
rem
rem         Outputfile: Results are stored in this file. If
rem                     it exists already results are
rem                     appended to it, otherwise it will be
rem                     newly created.
rem
rem
echo **************************************************
echo ***** RunPSQM        V1.0  (c) OPTICOM, 1998   ***
echo **************************************************
echo.
echo ... Processing file %1
Opera -Exec -Algorithm Name=PSQM -Input Inp=0 File=%1 Inp=1
   File=%2 -Mux InpRefLeft=0 ChannelRefLeft=0 InpTestLeft=1
   ChannelTestLeft=0 -Signal StaticGainOn AutoInvertTestSig
   -Out %3 -Append
echo Done!
```

7.2.7 Example, Bulk Call Testing

In the directory c:\programme\Opera\Batch an advanced example for scripting can be located. The purpose is to automate bulk call generation. It starts with running OptiCall and subsequently after all calls are finished, a PERL script is used to form the proper file names and to call OPERA for the QoS calculation. The result is a tab separated text file, which can be directly import into e.g. Excel for further evaluation. The bulk call testing demo consists of the following files:

Note:

Although this example was mainly written for the purpose of bulk call testing, it can be easily adapted to simple calls or repeated calls. It may be the best starting point for own automated tests.

BulkDemo.bat

This is the "main" program. It sets up some parameters, runs OptiCall and calls the PERL script for the evaluation. In the first few lines of the batch file you find some parameters like e.g. phone numbers etc. which you should adjust according to your needs.

BulkDemo.pl

This is a PERL script which generates the file names required for the evaluation with OPERA and calls OPERA with these names. The result is a text file with the measurement results of all test calls.

Note:

On all OPERA systems the PERL scripting language is preinstalled. It is not the latest version, but it is lightweight and uncomplicated to install. It is in the directory "C:\Program Files\Perl".

RunPesq.bat

This is the final call of OPERA using the PESQ algorithm. Bulkdemo.pl calls this batch.

Chapter
8

8 TECHNICAL SPECIFICATIONS

8.1 Software

Framework

Sound File Formats:

WAVE-files containing:

- A-law
- μ-law
- linear PCM, 8 or 16 bit

Maximum Duration of Measurement Signals:

File based:	As limited by WAVE-format
Online (Limitation of current version):	$\frac{2^{32}}{fs * \frac{bitspersample}{8}}$ (e.g. ~12.4 h at 48 kHz, 16 bit resolution)

Maximum Delay Compensation:

Automatic Delay Compensation mode:	±1000 ms
Static Delay:	Additional ±10 s
Delay Tracking:	±512 samples

Gain Compensation:

Maximum gain difference:	±60 dB

PEAQ Algorithm

General:

Algorithm:	Based on standard ITU-R BS.1387, Basic model
Sample rates:	48 kHz (according to recommendation ITU-R BS.1387) In addition to BS.1387, the current OPERA™ implementation also supports 44.1 kHz

Maximum Delay Compensation File-Based Version:

Automatic Delay Compensation:	±1000 ms
Static Delay:	Additional ±10 s
Delay Tracking:	±512 samples

Maximum Delay Compensation Online Version:

Automatic Delay Compensation:	±500 ms
Static Delay:	Additional ±10 s
Delay Tracking:	±512 samples

Available Measurement Results:

- Timesignal
- Spectrum
- Excitation
- Noise-to-Mask Ratio (NMR), averaged
- Noise-to-Mask Ratio (NMR) vs. Time
- Masked Threshold
- Loudness
- Objective Difference Grade (ODG), averaged (Advanced and Basic Version)
- Objective Difference Grade (ODG) vs. Time
- Distortion Index (DI), for the Advanced and the Basic Version
- Delay between the Reference Signal and the output signal of the device under test
- Attenuation of the test signal compared to the reference signal
- Modulation of the reference and test signal

Available BS.1387 intermediate results (MOVs):

- Average Bandwidth of the Reference Signal (AvgBwRef)
- Average Bandwidth of the output signal of the device under test (AvgBwTst)
- Total Noise-to-Mask Ratio (NMRtotB)

- Relative fraction of frames for which at last one frequency band contains a significant noise component (RDF)
- Average Distorted Block (=Frame), taken as the logarithm of the ratio of the total distortion to the total number of severely distorted frames (ADB)
- Maximum of the Probability of Detection after low pass filtering (MFPD)
- Harmonic structure of the error over time (EHS)
- Windowed averaged difference in modulation (envelopes) between Reference Signal and Signal Under Test (WinModDif1B)
- Averaged modulation difference (AModDif1B)
- Averaged modulation difference with emphasis on introduced modulations and modulation changes where the reference contains little or no modulations (AmodDif2B, RModDifA)
- RMS value of the averaged noise loudness with emphasis on introduced components (NloudB, NLA)
- Averaged Linear Distortions (ALD).

PSQM Algorithm

General:

Algorithm:	Based on ITU-T P.861 standard , including PSQM+ (improved for GSM)
Sample rates:	8 kHz and 16 kHz

Delay Compensation:

Automatic Delay Compensation mode:	±1000 ms
Static Delay:	Additional ±10 s
Delay Tracking:	±512 samples

Available Measurement Results:

- Timesignal
- Spectrum
- Excitation
- Percentage of silent intervals during a measurement (Silence)
- Percentage of time clipped frames during measurement (Time Clipped)
- Percentage of severely distorted frames during measurement (Sev. Distorted)
- PSQM according to P.861, silence weight = 0.0 (PSQM-W0)
- PSQM according to P.861, silence weight = 0.2 (PSQM-W2)
- PSQM according to P.861, silence weight = 0.4 (PSQM-W4)

- PSQM value of the silent intervals (PSQM-Silence)
- Mean Opinion Score (MOS) according to P.861, silence weight = 0.0 (OMOS-W0)
- Mean Opinion Score (MOS) according to P.861, silence weight = 0.2 (OMOS-W2)
- Mean Opinion Score (MOS) according to P.861, silence weight = 0.4 (OMOS-W4)
- Mean Opinion Score (MOS) according to PSQM+ (OMOS+)
- Mean Opinion Score (MOS) vs. Time
- Delay between the Reference Signal and the output signal of the device under test
- Attenuation of the test signal compared to the reference signal
- Modulation of the reference and test signal

PESQ Algorithm

General:

Algorithm:	Based on standard ITU-T P.862
Sample rates:	8 kHz and 16 kHz

Available Measurement Results:

- Timesignal
- Min., max. and average delay
- Delay Jitter, Delay vs. Time, Delay Probability Distribution Function
- MOS (total, speech, background noise)
- R-Factor
- PESQ-LQ
- Front-End-Clipping (FEC)
- Hold-Over-Time (HOT)
- Dropouts
- Loudness
- Signal Levels
- Attenuation
- Gain Variation

Echo Algorithm

General:

Maximum echo delay:	1000 ms
Frame size:	16 ms at 8 kHz
Averaging window size:	800 ms at 8 kHz

Available Measurement Results:

- Echo Return Loss (ERL)
- Momentary attenuation of the highest echo peak (ERLmom)
- Momentary delay of the highest echo peak (ERLmom Delay)
- Attenuation of the highest echo peak during the whole measurement period (ERLpeak)
- Delay of the highest echo peak during the whole measurement period (ERLpeak Delay)

8.2 Hardware

POTS Telephony Board

General:

Board Capacity:	4 analog 2 wire loop start interfaces.

Power / Environment:

Power requirements +5V/+12V/-12V:	0.8A (typ. 0.3A)/<0.1A/<0.1A
Operating: temp / humidity (noncondensing):	0°C/+50°C 5%/80%, non-condensing
Storage : temp / humidity (noncondensing):	-20°C/+70°C 5%/80%, non-condensing

Loop Parameters:

Impedance:	Country specific, linear (600Ω or 900Ω) or complex (Europe: according to ETSI TBR 21)

Audio Signal Processing:

Receive Range:	-50 to +0 dBm (optional configurable AGC above nominal -44 dBm).
Transmit:	Programmable (nominal –12 dBm).
Silence Detection:	Programmable (nominal –44 dBm).
Sampling Rates:	8 ksamples/sec (telephone industry standard).
Speech:	• 64 kbps μ-law or A-law per ITU-T G.711 • 16, 24, 32, 40 kbps ADPCM using ITU-T G.726 algorithm • 16, 24, 32 kbps NMS compatible ADPCM • 32 kbps VOX compatible ADPCM • 8, 16 bit PCM 11, 22, 44 kHz • 16 bit mono PCM 8 kHz

Audio Output:

Frequency:	300-3400 Hz
Impedance:	100 Ω
Level:	3 dB
Output Connector:	3.5 mm stereo jack

Audio Input:

Frequency:	300-3400 Hz
Impedance:	47 kΩ
Level:	2 Vcc peak-to-peak
Output Connector:	3.5 mm stereo jack

Tone Dialling:

DTMF Digits:	0-9,*,#, and ABCD per ITU-T Q.23 and Q.24
Rate:	Programmable (10 digits/sec nominal)
Dialling Parameters:	Software controllable
Dialling Amplitude:	Network compatible programmable range -33 dBm to 1dBm
Dial Tone Wait:	Wait-for-dial-tone capability

Audio Interface Option (LynxONE)

Analog:

Interface Type:	Two inputs and two outputs, cross-coupled electronically balanced, XLR connectors on audio cables
Level:	+4dBu nominal / +20dBu max or –10dBV nominal / +6dBV max, 600Ω load on outputs
Input impedance:	Balanced: 24kΩ, unbalanced 12kΩ
Output impedance:	Balanced: 100Ω, unbalanced 50Ω
Output drive capability:	600Ω impedance, 0.16µF capacitance
A/D and D/A converters:	24bit, 128x oversampling, sigma delta
Bit depth:	8, 16, 24 or 32 bit file types
Frequenzy response:	20Hz..20kHz, +0/-0.35dB
Dynamic range:	>103dB, A wtd., analog in to analog out
Signal to Noise Ratio	>99dB, A wtd., analog in to analog out
Channel crosstalk:	<-103dB, analog in to analog out, 1kHz signal @ -1dBFS
Input THD+N	0.0022%typ., 1kHz signal @ -1dBFS, 22Hz..22kHz BW, analog in to digital out
Output THD+N	0.0015%typ., 1kHz signal @ -1dBFS, 22Hz..22kHz BW, digital in to analog out

Digital:

Interface Type:	One input and one output / AES/EBU or S/P DIF format, transformer coupled, XLR connectors on audio cables
Sample rates:	32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz
Bit depth:	8, 16, 24 or 32 bit file types

Clock / Synchronisation:

Type:	External BNC input and output, internal clock or board to board synchronisation through internal header cable
External: Level / impedance:	TTl / 75Ω
External: Input frequency range	25kHz to 27MHz

Audio Interface Option (Digigram)

Power / Environment:

Power requirements +5V/+12V/-12V:	0.8A/0.3A/0.2A
Operating: temp / humidity (noncondensing):	0°C/+50°C 5%/90%
Storage : temp / humidity (noncondensing):	-5°C/+70°C 0%/95%

Inputs/Outputs:

Analog inputs (stereo):	2 balanced
Maximum input level/impedance:	+26 dBu/600 Ω or >15 kΩ (switchable)
Programmable input gain:	digital and analog
Digital inputs (stereo):	2 AES/EBU or S/PDIF
Other inputs:	Wordclock, LTC
Analog outputs (stereo):	2 balanced
Maximum output level/impedance:	+22 dBu/low impedance
Programmable output level:	digital and analog
Digital outputs (stereo):	2 AES/EBU or S/PDIF
AES11 synchronization:	yes
Connector:	62-pin SUB-D

Audio Specifications:

Sampling frequencies available:	Programmable from 6 kHz to 50 kHz in steps of 0.02 Hz
A/D and D/A converter resolutions:	20 bits
PCM recording (encoding):	8, 16 and 24 bits
Frequency response at 48 kHz (record + play):	20 Hz -20 kHz: ±0.2 dB
Signal to noise ratio (unweighted):	>90 dB
Distortion + noise at 1 kHz (record + play):	<–87 dB
Channel phase difference: 20 Hz/20 kHz:	<0.5°/1°
Analog channel crosstalk at 1 kHz:	<-95 dB

OPERA Workstation

The workstation type is available as a custom specific system on request only.

OPERA Portable

DC Power Supply:

Wattage:	400 W
Voltage:	90 to 135 V at 60 Hz; 180 to 265 V at 50 Hz Autoranging 90 to 265 V

Temperature:

Operating:	10° to 35°C°* (50° to 95°F)
Storage:	-40° to 65°C (-40° to 149°F)
Relative humidity:	20% to 80% (noncondensing)

Regulatory:

This device complies with part 15 of the FCC Rules.

REFERENCES

Literature

[BEER92] BEERENDS J. G., STEMERDINK J. A., *A perceptual audio quality measure based on a psychoacoustic sound representation*, J. Audio Eng. Soc., Vol. 40, No. 12, pp. 963-987, 1992

[BEER94] BEERENDS J. G., STEMERDINK J. A., *A perceptual speech quality measure based on a psychoacoustic sound representation*, J. Audio Eng. Soc., Vol. 42, No. 3, pp. 115-123, 1994

[BEER95] BEERENDS J. G., *Measuring the Quality of Speech and Music Codecs, an Integrated Psychoacoustic Approach*, 98th AES Convention, Paris 1995, Preprint #3945

[BEER02a] BEERENDS J. G., RIX A. W., HOLLIER M. P., HEKSTRA A. P., Perceptual Evaluation of Speech Quality (PESQ) The New ITU Standard for End-to-End Speech Quality Assessment, Part I – Time-Delay Compensation, J. Audio Eng. Soc., Vol. 50, No. 10, 2002

[BEER02b] BEERENDS J. G., RIX A. W., HOLLIER M. P., HEKSTRA A. P., Perceptual Evaluation of Speech Quality (PESQ) The New ITU Standard for End-to-End Speech Quality Assessment, Part II – Psychoacoustic Model, J. Audio Eng. Soc., Vol. 50, No. 10, 2002

[BENJ02] BENJAMIN E., *Evaluating Digital Audio Artifacts with PEAQ*, 113th AES Convention, Los Angeles 2002, Preprint #5711

[BRAN87] BRANDENBURG K., *Evaluation of Quality for Audio Encoding at low Bit Rates*, 82nd AES Convention, London 1987, Preprint #2433

[BRAN89] BRANDENBURG K., *Ein Beitrag zu den Verfahren und der Qualitätsbeurteilung für hochwertige Musikcodierung*, Ph.D. Thesis, Erlangen 1989

[BRAN97] BRANDENBURG K., BOSI M., *Overview of MPEG Audio: Current and Future Standards for Low-Bit-Rate Audio Coding*, J. Audio Eng. Soc., Vol. 45, No. 1/2, pp. 4-21, 1997

[BRAN92] BRANDENBURG K., SPORER Th.: *'NMR' and 'masking flag': Evaluation of Quality using Perceptual Criteria*, Proc. of the 11th International AES Conference on Audio Test and Measurement, Portland 1992, pp. 169-179

[COLO95] COLOMES C., LEVER M., RAULT J.B., DEHERY Y.F., *A perceptual model applied to audio bit-rate reduction*, J. Audio Eng. Soc., Vol. 43, pp. 233-240, 1995

[COLO99] COLOMES C., Schmidmer C., Thiede T., Treuniet W., *Perceptual Quality Assessment for Digital Audio: PEAQ – the new ITU Standard for Objective Measurement of Perceived Audio Quality*, Proc. Of the AES 17th International Conference, pp. 337-351, 1999, Florence Italy

[FLAN72] FLANAGEN J. L., *Speech Analysis, Synthesis and Perception,* Springer-Verlag, Berlin - Heidelberg - New York, 1972

[GILC96] GILCHRIST N., GREWIN Ch. (Editors), *Collected Papers on Digital Audio Bitrate Reduction,* AES Special Publication, AES 1996

[HERR92a] HERRE J., EBERLEIN E., SCHOTT H., BRANDENBURG K., *Advanced Audio Measurement System using Psychoacoustic Properties*, 92th AES Convention, Vienna 1992, Preprint #3332

[HERR92b] HERRE J., EBERLEIN E., SCHOTT H., SCHMIDMER Ch., *Analysis Tool for Realtime Measurements using Perceptual Criteria*", Proc. of the 11th International AES Conference on Audio Test and Measurement, Portland 1992, pp.180-190

[KARJ85] KARJALEINEN M., *A New Auditory Model for the Evaluation of Sound Quality of Audio Systems*, Proc. of the ICASSP 1985, pp. 608-611

[KEYH93] KEYHL M., HERRE J., SCHMIDMER Ch., *NMR Measurements of Consumer Recording Devices Which Use Low Bit-Rate Audio Coding*, 94th AES Convention, Berlin 1993, Preprint #3616

[KEYH96] KEYHL M., HERRE J., SCHMIDMER Ch., *NMR Measurements on Multiple Generations Audio Coding*, 96th AES Convention, Amsterdam, 1994, Preprint #3803

[KEYH98a] KEYHL M., SCHMIDMER Ch., HERRE J., HILPERT J., *Maintaining Sound Quality - Experiences and Constraints of Perceptual Measurements in Today´s and Future Networks*", 98th AES Convention, Paris, 1995, Preprint #3946

[KEYH98b] KEYHL M., SCHMIDMER Ch., SPORER Th., PETERSON R., *Quality Assurance Tests of MPEG Encoders for a Digital Broadcasting System (Part 2) - Minimizing Subjective Test Efforts by Perceptual Measurements*, 104th AES Convention, Amsterdam, 1998, Preprint #4753 (P16-7)

[KEYH00] KEYHL M., SCHMIDMER Ch., Wachter H., Rath S., Stoll G., Colomes C., Sporer T., *Evaluating the Perceived Audio Quality (PEAQ) of Internet Audio Codecs*, 109th AES Convention, Los Angeles, 2000

[LYNX00] LynxONE, Installation and Users Guide, Copyright (C) 2000, Lynx Studio Technology, Inc.

[MÖLL02] MÖLLER S., BERGER J., *Describing Telepone Speech Codec Quality Degradations by Means of Impairment Factors,* J. Audio Eng. Soc., Vol. 50, No. 9 , 2002

[NMS99] NATURAL MICROSYSTEMS, *Natural Access*, Documentation

[PAIL92] PAILLARD B., MABILLEAU P., MORISETTE S., SOUMAGNE J., *PERCEVAL: Perceptual evaluation of the quality of audio signals*, J. Audio Eng. Soc., Vol. 40, 21-31, 1992

[PRAC98] PRACHT St., *Voice Quality*, COMMUNICATE, November 1998, p. 43-46

[SEIT89] SEITZER D., BRANDENBURG K., KAPUST R., EBERLEIN E., GERHÄUSER H., KRÄGELOH S., SCHOTT H.: *DSP based real time implementation of an advanced analysis tool for audio channels*, Proc. ICASSP'89, pages 2057-2060, 1989

[SPOR96] SPORER Th., *Evaluating Small Impairments with the Mean Opinion Scale - Reliable or Just a Guess?*, 101st AES Convention 1996, Preprint #4396 (E-1)

[SPOR95a] SPORER Th., BRANDENBURG K., *Constraints of Filter Banks Used for Perceptual Measurement*, J. Audio Eng. Soc., Vol. 43, No. 3, 1995 March, pp. 107-116

[SPOR95b] SPORER Th., GBUR U., HERRE J., KAPUST R., *Evaluating a Measurement System*, J. Audio Eng. Soc., Vol. 43, No. 5, 1995 May, pp. 353-363

[SPOR97] SPORER Th., *Objective Audio Signal Evaluation - Applied Psychoacoustics for Modeling the Perceived Quality of Digital Audio*, 103rd AES Convention, New York, 1997 Preprint #4512

[SPOR98] SPORER Th., KEYHL M., SCHMIDMER Ch., PETERSON R., *Quality assurance tests of MPEG encoders for a digital broadcasting system (Part 1) - A method for subjective assessments of very-low bit-rate audio*, 104th AES Convention, Amsterdam, 1998

[STOL99] STOLL G., Beerends J., Bitto R., Brandenburg K., Colomes C., Feiten B., Keyhl M., Schmidmer C., Sporer T., Thiede T.,Treurniet W., *PEAQ - der neue ITU-Standard zur objektiven Messung der wahrgenommenen Audioqualität* , RTM - Rundfunktechnische Mitteilungen, die Fachzeitschrift für Hörfunk und Fernsehtechnik, September 1999, 43. Jahrgang, Seiten 81-120, ISSN 0035-9890

[TERH79] TERHARDT E., *Calculating Virtual Pitch*, Hearing Research, Vol. 1, 1979, p. 155-182

[THIE96] THIEDE Th., KABOT E., *A New Perceptual Quality Measure for Bit Rate Reduced Audio*, 100th AES Convention, Copenhagen, 1996, Preprint #4280

[THIE00] THIEDE Th., Treurniet W., Bitto R., Schmidmer C., Sporer T., Beerends J., Colomes C., Keyhl M., Stoll G., Brandenburg K., Feiten B., *PEAQ – The ITU Standard for Objective Measurement of Perceived Audio Quality*, J. Audio Eng. Soc., Vol. 48, 2000

[TREU00] TREURNIET W. C. , SOULODRE G. A. , *Evaluation of the ITU-R Objective Audio Quality Measurement Method*, J. Audio Eng. Soc., Vol. 48, Number 3, March 2000

[ZWIC67] ZWICKER E., FELDTKELLER R., *Das Ohr als Nachrichtenempfänger*, Hirzel-Verlag, Stuttgart, 1967

[ZWIC82] ZWICKER E., *Psychoakustik*, Springer-Verlag, Berlin - Heidelberg - New York, 1982

Standards

[ETSI96] ETSI Technical Report ETR 250, *Transmission and Multiplexing (TM); Speech communication quality from mouth to ear for 3,1 kHz handset telephony across networks*, ETSI 1996

[ISO97] ISO/IEC/JTC1/SC29/WG11 Draft Document N1557, *Evaluation Methods and procedures for MPEG-4 tests*, 1997

[ITUR562] ITU-R Recommendation BS.562-3, *Subjective assessment of sound quality*

[ITUR1116] ITU-R Recommendation BS.1116-1, *Methods for the Subjective Assessment of small Impairments in Audio Systems including Multichannel Sound Systems*, 1997

[ITUR1387] ITU-R Recommendation BS.1387-1, *Method for Objective Measurements of Perceived Audio Quality (PEAQ)*, Revised 11/01

[ITUR1534] ITU-R Recommendation BS.1534, *Method for the subjective assessment of intermediate quality level of coding systems)*, June 2001

[ITUT96a] ITU-T Contribution COM12-74-E, *Review of Validation Tests for Objective Speech Quality Measures*, March 1996

[ITUT107] ITU-T Recommendation G.107, *The E-model, a computational model for use in transmission planning,*May 2000

[ITUT420] ITU-T Recommendation E.420, *Checking the Quality of the International Telephone Service – General Considerations,* 1988, (Extract from the *Blue Book)*

[ITUT562] ITU-T Recommendation P.562, *Analysis and interpretation of INMD voice-service measurements,* May 2000

[ITUT800] ITU-T Recommendation P.800, *Methods for subjective determination of transmission quality*, 1996

[ITUT810] ITU-T Recommendation P.810, *Modulated Noise Reference Unit (MNRU),* 1996

[ITUT830] ITU-T Recommendation P.830, *Subjective Performance Assessment of Telephone-Band and Wideband Digital Codecs*, 1996

[ITUT833] ITU-T Recommendation P.833, *Methodology for Derivation of Equipment Impairment Factors from Subjective Listening-Only Tests,* 2001

[ITUT834] ITU-T Recommendation P.834, *Methodology for the Derivation of Equipment Impairment Factors from Instrumental Models,* 2002

[ITUT861] ITU-T Recommendation P.861, *Objective Quality measurement of telephone-band (300 - 3400 Hz) speech codecs*, 1996

[ITUT862] ITU-T Recommendation P.862, *PESQ an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs*, February 2001

Glossary of Terms

ACR

Absolute category rating test method according to the ITU-T recommendation P.800 used for the assessment of speech codecs. Within the ACR test method, a five grade impairment scale is applied. Because of the telecomunication environment the testing is done without a comparison to an undistorted reference.

ADB

Average Distorted Block

ADPCM

According to standard ITU-T G.726. Bit rate of 32 kbit/s (also possible with 16, 24 and 40 kbit/s).

AES

The Absolute Error Score (AES) is derived from a formula developed especially for evaluating the quality of the results obtained from an objective perceptual measurement method. It takes the confidence intervals of the average values of subjective listening tests into account

Artefact

Spurious effects or imperfections introduced into a signal as a result of signal processing.

ATM

Asynchronous Transfer Mode

ASD

Auditory Spectral Difference, a psychoacoustically motivated measure for the audible difference between two sounds.

ASR

Answer Seizure Ratio. Defines the ratio between sucessful call attempts and the total number of calls.

BAQ

The Basic Audio Quality (BAQ) is defined as a global subjective attribute which includes any and all detected differences between the Reference Signal and a processed version of it.

BER

Bit Error Rate

CCI

Call Clarity Index

CELP

Code Excited Linear Prediction

CI

Confidence Interval

CM

Coding Margin

The Coding Margin is a quality parameter which measures the headroom of inaudible coding artefacts to the threshold when these artefacts become audible.

CTI

Computer Telephony Integration

Short for computer-telephony-integration, which refers to systems that enable a computer to act as a call center, accepting incoming calls and routing them to the appropriate device or person. Today's CTI systems are quite sophisticated and can handle all sorts of incoming and outgoing communications, including phone calls, faxes, and Internet messages.

ETSI

European Telecommunications Standardization Institute

IETF

Internet Engineering Task Force

ISDN

Integrated Services Digital Network

ITU-R

The radio communication sector of the International Telecommunication Union, Geneva, (former CCIR), see also http://www.itu.int.

ITU-T

The telecommunication sector of the International Telecommunication Union, Geneva, (former CCIR), see also http://www.itu.int.

LD-CELP

Low-Delay CELP (Code Excited Linear Prediction) Speech Coder. According to standard ITU-T G.726. Bit rate of 16 kbit/s.

MNRU

Modulated noise reference units

MOS

Mean listening-quality Opinion Score, or simply Mean Opinion Score. The MOS is the mean of the given scores for a device under test of all test subjects in a subjective listening test.

MOV

The Model Output Variables are intermediate output values of the perceptual measurement method. These variables are based on basic psycho-acoustical findings and may therefore be used to characterize the coding artefacts further.

MPLS

Multi-Protocol Label Switching

MUSHRA

Stands for "Multiple Stimulus With Hidden Reference Anchors". A new draft ITU recommendation on testing very low bit rate audio.

NMR

The measurement scheme NMR (Noise-to-Masked-Ratio) [BRAN87] evaluates the level-difference between the masked threshold and the noise signal. A DFT with a Hann window of about 20 ms is used to analyse the frequency content of the signal. The transform coefficients are combined to bands according to the Bark scale. The masked threshold is estimated for each band. The slope of the masked threshold is derived using a worst case approach taking into account the fact that the slopes are steeper for weak signals but run into the absolute threshold at higher levels. The absolute threshold is adapted to the resolution of the input signal (usually 16 bits), but not to psycho-acoustic demands. Due to these facts NMR is robust to changes of the reproduction level. The pitch scale resolution is about 1 Bark. Since the required computational power is low it was possible to implement NMR as a real time system at an early stage of its development.

The model has been in use since 1987 and has proven its basic reliability.

The most important output values of NMR are the masking flag rate, giving the percentage of frames with audible distortions, as well as the total and mean NMR which are different ways of averaging the distance between the error energy and the masked threshold.

ODG

According to the ITU-R recommendation BS.1387, the ODG (Objective Difference Grade) is the output variable from the objective measurement method and corresponds to the SDG (Subjective Difference Grade) in the subjective domain. The resolution of the ODG is limited to one decimal. One should however be cautious and not generally expect that a difference between any pair of ODGs of a tenth of a grade is significant. The same remark is valid when looking at results from a subjective listening test.

Offline measurements

Measurement procedure which does not interact with the ongoing programme transmission.

Online measurements

Measurement procedure which relies on the ongoing programme transmission, or parts thereof.

Origin

The near end of a telephone call. This is usually the party which initiates a call, i.e. the party that dials. In older versions of OPERA this was the "Caller".

PBX

Private Branch Exchange

PCM

Pulse Code Modulation

Pulse Code Modulation. According to standard ITU-T G.711. Bit rate of 64 kbit/s

PDD

Post Dial Delay

Refers to the time elapsed between the last dial tone and the first response of the network.

PDF

the Probability Density Function of a vector, shows the probability for the occurrence of each individual number in the vector.

PEAQ

Perceptual Evaluation of Audio Quality

Stands for \Perceptual Evaluation of Audio Quality\", the perceptual measurement technique recommended for wide band (music) audio signals as ITU-R BS.1387 in 1999. See also www.peaq.org."

PESQ

PESQ stands for “Perceptual Evaluation of Speech Quality” the new ITU standard P.862. At the time PSQM was standardized as P.861, the scope of the standard was to assess speech codecs, used primarily for mobile transmission, like GSM. VoIP was not yet a topic. The requirements for measurement equipment have changed dramatically since then. As a consequence, the ITU set up a working group to revise the P.861 standard and to cope with the new demands arising from next generation networks like VoIP. Within these networks, the measurement algorithm has to deal with much higher distortions than with GSM codecs, but perhaps the most eminent factor is that the delay between the reference and the test signal is no longer constant. PESQ combines the excellent psycho- acoustic and cognitive model of PSQM+ with a time alignment algorithm adopted from PAMS, that handles varying delays perfectly. PESQ is not designed for streaming applications, which is itís only drawback. This is why it cannot fully replace PSQM+. With PSQM and PESQ there are now two standards that cover the entire problem of measuring speech quality. See also www.pesq.org.

PGAD

Post Gateway Answer Delay

POTS

Plain Old Telephony Service (often used to characterize the traditional analog telephone service).

PSQM

Perceptual Speech Quality Measure

Stands for \Perceptual Speech Quality Measure\", the perceptual measurement technique recommended in ITU-T P.861. See also www.psqm.org."

Reference

Test excerpt, reproduced without the processing by a test object, used as a comparison basis for an impairment test.

RSVP

Resource Reservation Protocol

SDG

According to ITU-R BS.1387, the analysis of the results from a subjective listening test is in general based on the SDG (Subjective Difference Grade) defined as:

SDG = GradeSignal Under Test - GradeReference Signal

The SDG values should ideally range from 0 to -4, where 0 corresponds to an imperceptible impairment and -4 to an impairment judged as very annoying.

Side Tone

Short echoes with a delay shorter than approximately 5ms and an attenuation > 15dB.

Subject

A test person evaluating the stimuli in a listening test.

TAPI

Telephony application protocol interface

An application protocol interface defined by Microsoft.

ToS

Type of Service

Termination

This is the far end of a telephone conversation. In general this is the party which receives a phone call. In older OPERA versions this was the "Called" telephone line.

VAD

Voice Activity Detection. VAD is part of most VoIP systems and Echo cancelers. It is used to distinguish active speech from silence.

VoIP

Voice over Internet Protocol

Voice over Internet Protocol, a series of techniques permitting transmission of telephony over the Internet. Often makes use of ITU-T G.7xx audio compression recommendations.

WFQ

Weighted Fair Queuing

Index

S

T

U

V

W

APPENDIX

White Paper: State of the Art Voice Quality Testing

White Paper: OPERA Application Notes (1)

ITU-T Recommendation P.861

ITU-T Recommendation P.862

White Paper: PEAQ – The ITU Standard for Objective Measurement of Perceived Audio Quality

ITU-R Recommendation BS.1387-1