INTERNATIONAL TELECOMMUNICATION UNION

ITU-T P.563

(05/2004)

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS

Objective measuring apparatus

Single-ended method for objective speech quality assessment in narrow-band telephony applications

ITU-T Recommendation P.563

ITU-T P-SERIES RECOMMENDATIONS

TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS

Vocabulary and effects of transmission parameters on customer opinion of transmission quality	Series	P.10
Subscribers' lines and sets	Series	P.30 P.300
Transmission standards	Series	P.40
Objective measuring apparatus	**Series**	**P.50** **P.500**
Objective electro-acoustical measurements	Series	P.60
Measurements related to speech loudness	Series	P.70
Methods for objective and subjective assessment of quality	Series	P.80 P.800
Audiovisual quality in multimedia services	Series	P.900

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation P.563

Single-ended method for objective speech quality assessment in narrow-band telephony applications

Summary

This Recommendation describes an objective single-ended method for predicting the subjective quality of 3.1 kHz (narrow-band) telephony applications. This Recommendation presents a high-level description of the method and advice on how to use it. An ANSI-C reference implementation, described in Annex A, is provided in separate files and forms an integral part of this Recommendation. A conformance testing procedure is also specified in Annex A to allow a user to validate that an alternative implementation of the model is correct. This ANSI-C reference implementation shall take precedence in case of conflicts between the high-level description as given in this Recommendation and the ANSI-C reference implementation.

This Recommendation includes an electronic attachment containing an ANSI-C reference implementation and conformance testing data.

Source

ITU-T Recommendation P.563 was approved on 14 May 2004 by ITU-T Study Group 12 (2001-2004) under the ITU-T Recommendation A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Electronic attachment: ANSI-C reference implementation and conformance testing data.

ITU-T Recommendation P.563

Single-ended method for objective speech quality assessment in narrow-band telephony applications[1]

1 Introduction

The P.563 algorithm is applicable for speech quality predictions without a separate reference signal. For this reason, this method is recommended for non-intrusive speech quality assessment, live network monitoring and assessment by using unknown speech sources at the far-end side of a telephone connection.

Real systems may include background noise, filtering and variable delay, as well as distortions due to channel errors and speech codecs. Up to now, methods for speech quality assessment of such systems, such as ITU-T Rec. P.862, require either a reference signal or they calculate only quality indexes based on a restricted set of parameters like level, noise in speech pauses and echoes.

The P.563 approach is the first recommended method for single-ended non-intrusive measurement applications that takes into account the full range of distortions occurring in public switched telephone networks and that is able to predict the speech quality on a perception-based scale MOS-LQO according to ITU-T Rec. P.800.1. This Recommendation is not restricted to end-to-end measurements; it can be used at any arbitrary location in the transmission chain. The calculated score is then comparable to the quality perceived by a human listener, who is listening with a conventional shaped handset at this point.

The validation of P.563 included all available experiments from the former P.862 validation process, as well as a number of experiments that specifically tested its performance by using an acoustical interface in a real terminal at the sending side. Furthermore, the P.563 algorithm was tested independently with unknown speech material by third party laboratories under strictly defined requirements.

It is recommended that P.563 be used for speech quality assessment in 3.1 kHz (narrow-band) telephony applications only.

2 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

– ITU-T Recommendation P.48 (1988), *Specification for an intermediate reference system*.

– ITU-T Recommendation P.800 (1996), *Methods for subjective determination of transmission quality*.

– ITU-T Recommendation P.810 (1996), *Modulated noise reference unit (MNRU)*.

---

[1] This Recommendation includes an electronic attachment containing an ANSI-C reference implementation and conformance testing data.

– ITU-T Recommendation P.830 (1996), *Subjective performance assessment of telephone-band and wideband digital codecs*.

– ITU-T Recommendation P.862 (2001), *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*.

– ITU-T P-series Recommendations – Supplement 23 (1998), *ITU-T coded-speech database*.

3 Abbreviations

This Recommendation uses the following abbreviations:

ACR	Absolute Category Rating
CELP	Code-Excited Linear Prediction
dBov	dB to overload point
DCME	Digital Circuit Multiplication Equipment
ERP	Ear Reference Point
HATS	Head and Torso Simulator
IRS	Intermediate Reference System
LPC	Linear Prediction Coefficient
MOS	Mean Opinion Score
MOS-LQO	Mean Opinion Score – Listening Quality Objective
MOS-LQS	Mean Opinion Score – Listening Quality Subjective
PCM	Pulse Code Modulation
SNR	Signal-to-Noise Ratio
SPL	Sound Pressure Level

4 Scope

Based on the benchmark results presented within Study Group 12 in 2003, an overview of the test factors, coding technologies and applications to which this Recommendation applies is given in Tables 1 to 3. Table 1 presents the relationships of test factors, coding technologies and applications for which this Recommendation has been found to show acceptable accuracy. Table 2 presents a list of conditions for which the Recommendation is known to provide inaccurate predictions or is otherwise not intended to be used. Finally, Table 3 lists factors, technologies and applications for which P.563 has not currently been validated. Although correlations between objective and subjective scores in the benchmark were around 0.89 for both known and unknown data, the P.563 algorithm cannot be used to replace subjective testing but it can be applied for measurements where auditory tests would be too expensive or not applicable at all.

It should also be noted that the P.563 algorithm does not provide a comprehensive evaluation of transmission quality. It only measures the effects of one-way speech distortion and noise on speech quality in the same way as it can be investigated by an auditory test assessing listening quality on an ACR scale. The P.563 algorithm scores the speech signal in that way, as it is presented to a human listener by using a conventional shaped handset and listening with a SPL of 79 dB at the ERP.

Because P.563 models the human quality perception in combination with a common receiving terminal, the degradation produced by a receiving terminal and other equipment in a real monitored connection, which are connected behind the measurement point, cannot be taken into account.

Because P.563 predicts listening quality scores, all effects degrading talking quality or conversational quality only cannot be taken into account. That means, the effects of loudness loss, delay, sidetone, talker-echo, and other impairments related to the talking quality or two-way interaction only are not reflected in the P.563 scores. Therefore, it is possible to have high P.563 scores, yet non-optimal quality of the connection overall.

It should be highlighted that P.563 is designed for the prediction of speech quality in public switched narrow-band telephone networks. The types and the amount of the distortions, technologies and applications in the validation procedure cover the range of common occurrences in such networks. Extreme situations, even if they fulfil the terms of Table 1, may be predicted inaccurately.

Table 1/P.563 – Factors for which P.563 had demonstrated acceptable accuracy and recommended application scenarios

Test factors
Characteristics of the acoustical environment (reflections, different reverberation times) as used in the validation phase. Mobile and conventional shaped handsets as well as handsfree terminals according to P.340 test-setup in office environments were used (See Note).
Environmental noise at the sending side
Characteristics of the acoustical interface of the sending terminal
Remaining electrical and encoding characteristics of the sending terminal
Speech input levels to a codec
Transmission channel errors
Packet loss and packet loss concealment with CELP codecs
Bit rates if a codec has more than one bit-rate mode
Transcodings
Effect of varying delay on listening quality in ACR tests
Short-term time warping of speech signal
Long-term time warping of speech signal
Transmission systems including echo cancellers and noise reduction systems under single talk conditions and as they will be scored on an ACR scale
Coding technologies
Waveform codecs, e.g., G.711; G.726; G.727
CELP and hybrid codecs ≥4 kbit/s, e.g., G.728, G.729, G.723.1
Other codecs: GSM-FR, GSM-HR, GSM-EFR, GSM-AMR, CDMA-EVRC, TDMA-ACELP, TDMA-VSELP, TETRA

Table 1/P.563 – Factors for which P.563 had demonstrated acceptable accuracy and recommended application scenarios

Recommended application scenarios for P.563
Live network monitoring using digital or analogue connection to the network
Live network end-to-end testing using digital or analogue connection to the network
Live network end-to-end testing with unknown speech sources at the far end side
NOTE – For more detailed information, please refer to the published testplans ("Joint Test Plan for Single-Ended Assessment Models", COM 12-D 121, January 2003).

Table 2/P.563 –- P.563 is known to provide inaccurate predictions when used in conjunction with these variables, or is otherwise not intended to be used with these variables

Test factors
Listening levels, Loudness loss (See Note.)
Sidetone
Effect of delay in conversational tests
Talker echo
Music or network tones as input signal
Coding technologies
LPC vocoder technologies at bit rates < 4.0 kbit/s, e.g., IMBE, AMBE, LPC10e
Applications
Predicting talking quality
Two-way communication performance
NOTE – P.563 assumes a standard listening level of 79 dB SPL and compensates for non-optimum signal levels in the input files. The subjective effect of deviation from optimum listening level is therefore not taken into account.

Table 3/P.563 – Factors, technologies and applications for which P.563 has not or not fully been validated at the time of the standardization

Test factors
Amplitude clipping of speech (was not included in the evaluation data)
Talker dependencies and multiple simultaneous talkers
Singing voice and child's voice as input to a codec
Bit-rate mismatching between an encoder and a decoder if a codec has more than one bit-rate mode
Artificial speech signals as input to a codec
Listener echo
Effects/artifacts from isolated echo cancellers
Effects/artifacts from isolated noise reduction algorithms
Evaluation of synthetic speech and/or using it as input to a speech codec

Table 3/P.563 – Factors, technologies and applications for which P.563 has not or not fully been validated at the time of the standardization

Coding technologies
CELP and hybrid codecs < 4 kbit/s
MPEG-4 HVXC
Applications
Measurements at the acoustic interface of the receiving terminal/handset, e.g., using HATS

5 Convention

Subjective evaluation of telephone networks and speech codecs may be conducted using listening-only or conversational methods of subjective testing. For practical reasons, listening-only tests are the only feasible method of subjective testing during the development of speech codecs, when a real-time implementation of the codec is not available. Also listening-only tests are often not practicable for live network monitoring. This Recommendation discusses an objective measurement technique for estimating subjective quality obtained in listening-only tests, using listening equipment conforming to the IRS or modified IRS receive characteristics.

The P.563 approach predicts the results of ACR listening quality (LQS) subjective experiments by calculating a listening quality value (LQO) using the common MOS scale from 1 to 5. This Recommendation should, therefore, be considered to relate primarily to the ACR LQ opinion scale.

6 Requirements on speech signals to be assessed

The described algorithm is designed for evaluating human speech only. It cannot be used for the evaluation of music, noise or other non-speech audio signals. The applicability if singing voice is transmitted over telephone connections has not yet been validated.

The speech signal to be assessed has to be recorded at an 'electrical' interface. That means, recordings made by an artificial ear in the acoustical domain cannot be used. Furthermore, outcomes simulation of speech transmissions or other speech processing can be used if they are covered by the scope given in Table 1 and do not include a terminal simulation.

The digitized speech signal has to fulfil the following requirements:

- Sampling frequency: 8000 Hz

  If higher frequencies are used for recording, a separate down-sampling by using a high quality flat low pass filter has to be applied. Lower sampling frequencies are not allowed.
- Amplitude resolution: 16 bit linear PCM
- Minimum active speech in file: 3.0 s
- Maximum signal length: 20.0 s
- Minimum speech activity ratio: 25%
- Maximum speech activity ratio: 75%
- Range of active speech level: –36.0 to –16.0 dBov

  A level adjustment to –26 dBov is part of P.563. The recommended level limitation should avoid additional artefacts by low SNR or amplitude clipping respectively.

7 Overview of P.563

In comparison to P.862 (a so-called 'double-ended' or 'intrusive' method) that compares a high quality reference signal to the degraded signal on a basis of a perceptual model, P.563 predicts the speech quality of a degraded signal without a given reference speech signal. Figure 1 illustrates the differences in these approaches.

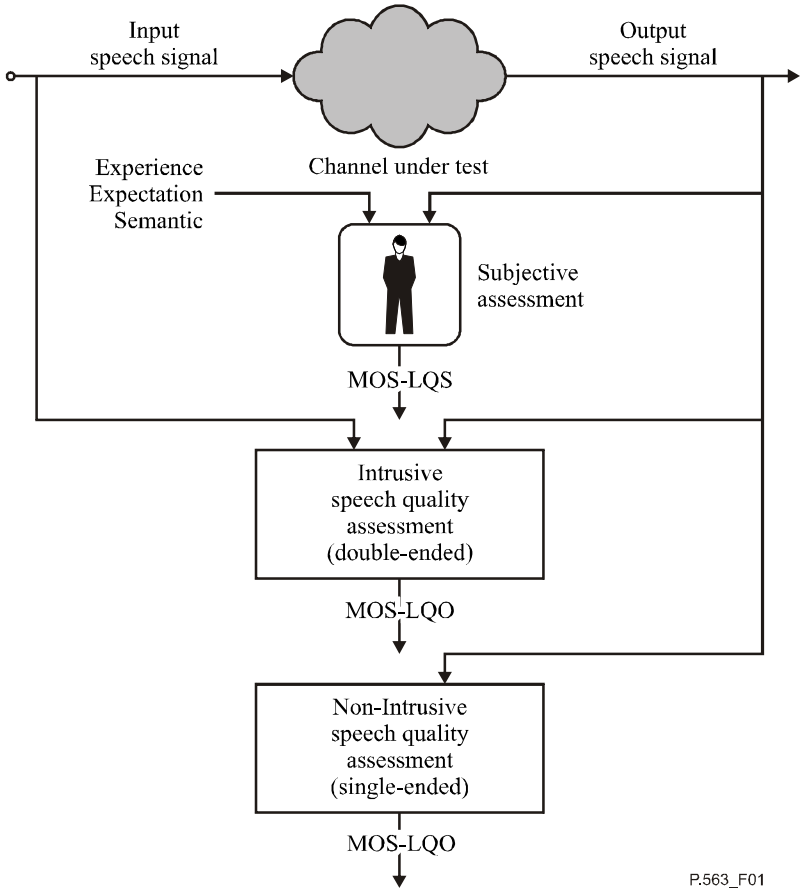


Figure 1/P.563 – Non-Intrusive versus Intrusive models

The P.563 approach could be visualized as an expert who is listening to a real call with a test device like a conventional handset into the line in parallel. This visualization explains also the main application and allows the user to rate the scores gained by P.563. The quality score predicted by P.563 is related to the perceived quality by linking a conventional handset at the measuring point.

Consequently, the listening device has to be part of the P.563 approach. Therefore, each signal will first be pre-processed. This pre-processing begins with the model of the receiving handset. Following this, a voice activity detector (VAD) is used to identify portions of the signal that contain speech and the speech level is calculated. Finally, a speech level adjustment to –26 dBov is applied. The pre-processed speech signal to be assessed will be investigated by several separate analyses, which detect, like a sensor layer, a set of characterizing **signal parameters**. This analysis will be applied at first to all signals. Based on a restricted set of **key parameters,** an assignment to a **main distortion** class will be made.

The key parameters and the assigned distortion class are used for the adjustment of the speech quality model. This provides a perceptual based weighting where several distortions are occurring

in one signal but one distortion class is more prominent than the others. The basic block-scheme of P.563 is shown in Figure 2.

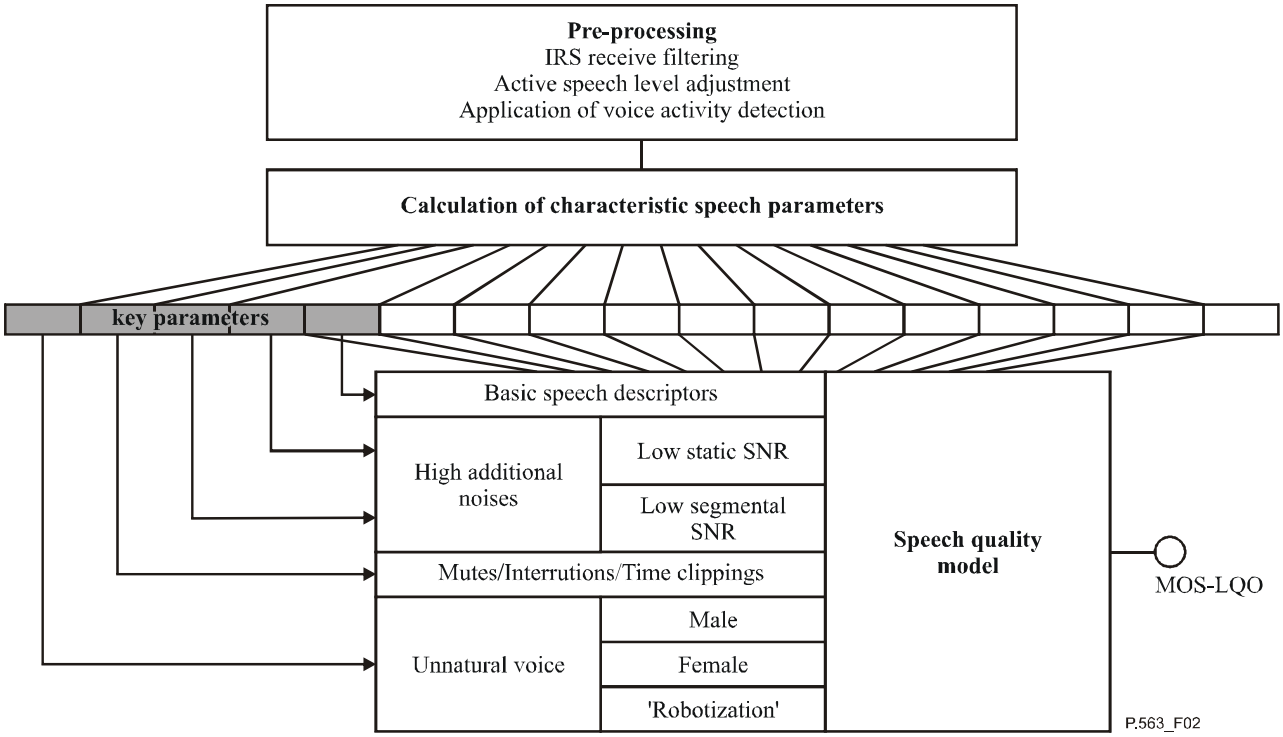


Figure 2/P.563 – Block scheme of P.563

Basically, the P.563 algorithm's signal parameterization can be divided into three independent functional blocks that correspond to the main classes of distortion:

– **Vocal tract analysis and unnaturalness of speech**
  1) Male voices;
  2) Female voices;
  3) Strong 'Robotization'.

– **Analysis of strong additional noise**
  1) Low static SNR (Background noise floor);
  2) Low segmental SNR (Noise that is related to the signal envelope).

– **Interruptions, mutes and time clipping**

In addition, a set of basic speech descriptors like active speech level, speech activity and level variations will be used, mainly for adjusting the pre-processing and the VAD. Some of the signal parameters calculated within the pre-processing stage will be used in these three functional blocks.

7.1 Vocal tract analysis and unnaturalness of speech

The main block looks for unnaturalness in the speech signal. This functional block contains a speech production model for extracting signal parts that could be interpreted as voice and separates them from the non-speech parts. Furthermore, high order statistical analysis gives additional information about how human-like the speech is.

The unnaturalness of speech will be rated separately for male and female voices. Furthermore, in the case of strong robotization[2], another separate rating is made, which is gender-independent.

In this clause the signal is investigated for the occurrence of tones like DTMF-tones or similar highly periodic signals that are not speech.

Other very annoying disturbances are repeated speech frames. In packet-based transmission systems, a typical error that can occur is the loss of packets. Some speech codecs employ error concealment methods in order to increase the received speech quality. In fact, some error concealment methods use packet (frame) repetitions that simply replace a lost packet by, for example, a previously successfully transmitted packet, and tend to decrease the quality of the signal rather than to increase it.

A more general description of the received speech quality is given by comparing the input signal with a pseudo reference signal generated by a speech enhancer.

7.2 Analysis of strong additional noise

The noise analysis calculates different characteristics of noise. Based on two key parameters, the decision will be made if additional noise is the main degradation. If additional noise is detected as the main degradation class, a decision is made for the type of noise. Either it is static and present over all the signal (at least during speech activity) such that the noise power is not correlated with the speech signal, or the noise power shows dependencies on the signal power envelope.

If there was noise found that is likely to be static, several detectors try to quantify the amount of noise 'locally' and 'globally'. The expression 'local' noise as it is used here, describes the signal parts found especially between phonemes, whereas 'global' noise was defined as the signal between utterances such as sentences. Distinguishing between those noise types is important as, for example, in mobile communications often different settings for speech active parts and non-active parts are applied, e.g., introduction of comfort noise.

7.3 Interruptions, mutes and time clipping

Mutes and interruptions also form a separate distortion class. Such distortions can only partly be described by outcomes of the vocal tract investigation. Hence, a separate analysis is made to detect and to rate time clippings and unnatural mutes in the signal.

Signal interruption can occur in two variants i.e., as temporal speech clipping or speech interruption. Both lead to a loss of signal information.

Temporal clipping may occur when voice activity detection is used, when DCME is used or the signal becomes interrupted. This clipping is an annoying phenomenon that cuts off a bit of speech in the instant it takes for the transmitter to detect presence of speech. It is possible to detect the interruptions of the speech signal, which occur during the active speech intervals. The algorithms used in P.563 are able to distinguish between normal word ends and abnormal signal interruptions as well as unnatural silence intervals in a speech utterance.

7.4 Distortion classification

Table 4 gives an overview for all calculated signal parameters. The key parameters that are used for classification of the main distortions are highlighted in grey.

The table also reflects the structure of the rest of this Recommendation. The main columns are in line with the sections or the corresponding main distortion classes respectively.

---

[2] Robotization is caused by a voice signal that contains too much periodicity.

All the names used for the parameters can be found and are explained in later sections of this Recommendation as well as in the example source code.

Table 4/P.563 – Overview for all used signal parameters in P.563

Basic speech descriptors	Unnatural speech		Noise analysis		Interruptions/ mutes
	Vocal tract analysis	Speech statistics	Static SNR	Segmental SNR	
PitchAverage	*Robotization*	*LPCcurt*	*SNR*	*EstSegSNR*	*SpeechInterrup-tions*
SpeechSection LevelVar	*ConsistentArt Tracker*	*LPCskew*	*EstBGNoise*	*SpecLevel Dev*	*SharpDeclines*
SpeechLevel	*VTPMaxTube Section*	*LPCskewAbs*	*NoiseLevel*	*SpecLevel Range*	*MuteLength*
LocalLevelVar	*FinalVtpAverage*	*CepCurt*	*HiFreqVar*	*RelNoise Floor*	*Unnatural Silence*
	VTPPeakTracker	*CepSkew*	*SpectralClarity*		*Unnatural SilenceMean*
	ArtAverage	*CepADev*	*GlobalBGNoise*		*Unnatural SilenceTotEnergy*
	VtpVadOverlap		*GlobalBGNoise TotEnergy*		
	PitchCrossCorrl Offset		*GlobalBGNoise RelEnergy*		
	PitchCrossPower		*GlobalBGNoise AffectedSamples*		
	BasicVoiceQuality		*LocalBGNoise Log*		
	BasicVoiceQuality Asym		*LocalBGNoise Mean*		
	BasicVoiceQuality Sym		*LocalBGNoise Stddev*		
	FrameRepeats		*LocalBGNoise*		
	FrameRepeats TotEnergy		*LocalBGNoise AffectedSamples*		
	UnnaturalBeeps				
	UnnaturalBeeps Mean				
	UnnaturalBeeps AffectedSamples				

8 Comparison between objective and subjective scores

Subjective votes are influenced by many factors such as the preferences of individual subjects and the context (the other conditions) of the experiment. Thus, a regression process is necessary before a direct comparison can be made. The regression must be monotonic so that information is preserved, and it is normally used to map the objective P.563 score onto the subjective score. A

good objective quality measure should have a high correlation with many different subjective experiments if this regression is performed separately for each one and, in practice, with P.563, the regression mapping is often almost linear, using a MOS-like scale.

A preferred regression method for calculating the correlation between the P.563 score and subjective MOS, which was used in the validation of P.563, uses a 3rd-order polynomial constrained to be monotonic. This calculation is performed on a per study basis. In most cases, condition MOS is the chosen performance metric, so the regression should be performed between condition MOS and condition-averaged P.563 scores. A condition should use at least four different speech samples. The result of the regression is a set of objective MOS scores for that test. In order to be able to compare objective and subjective scores the subjective MOS scores should be derived from a listening test that is carried out according to ITU-T Rec. P.830.

8.1 Correlation coefficient

The closeness of the fit between P.563 and the subjective scores may be measured by calculating the correlation coefficient. Normally, this is performed on condition-averaged scores, after mapping the objective to the subjective scores. The correlation coefficient is calculated with Pearson's formula:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

In this formula, x_i is the subjective condition MOS for condition i, and $\bar{x}$ is the average over the subjective condition MOS values, x_i. y_i is the mapped condition-averaged P.563 score for condition i, and $\bar{y}$ is the average over the condition-averaged P.563 MOS values y_i.

For 24 known ITU benchmark experiments, the average correlation was 0.88. For an agreed set of six experiments used in the independent validation, experiments that were unknown during the development of P.563, the average correlation was 0.90.

9 High level description of the functional blocks used in P.563

This clause explains the functional blocks used in P.563 and shown in Figure 2.

9.1 Description of basic speech descriptors and the signal pre-processing

9.1.1 Voice activity detection

The Voice Activity Detection (VAD) algorithm is based on an adaptive power threshold, using an iterative approach. Envelope frames above this threshold are classified as speech, and below as noise.

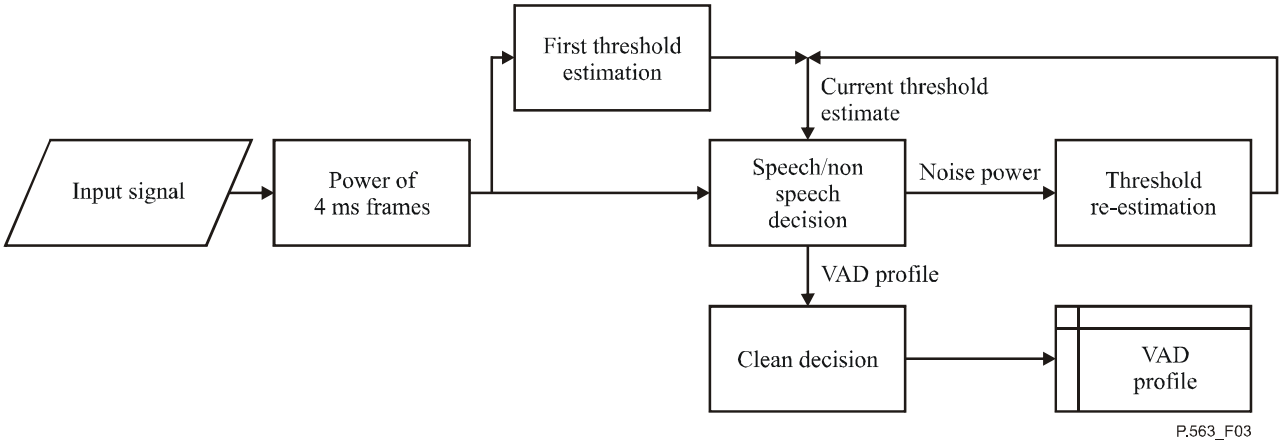


Figure 3/P.563 – Voice activity detection overview

As shown in Figure 3, the VAD threshold is initially set to the mean power of the signal. This initialization allows the first speech/noise decision to be made. The threshold is then iteratively re-estimated to be the mean plus two standard deviations of the power of the noise sections over the whole signal. Twelve iterations are completed, after which the final threshold is determined and used to make the final speech/noise decision.

$$\begin{cases} m_x = \sum_{i=0}^{n-1} \frac{x_{noise}(i)}{n} \\ th_{est} = m_x + 2 \times \sqrt{\sum_{i=0}^{n-1} \frac{(x_{noise}(i) - m_x)^2}{n}} \end{cases}$$

Additional processing is performed to label short events:

- Events that are above threshold for 12 ms or less are labelled as non-speech;
- Speech utterances that are separated by less than 200 ms are joined together.

9.1.2 IRS filtering

It is assumed that the listening tests were carried out using an IRS or a modified IRS receive characteristic in the handset. For some of the parameters, this has to be taken into account.

In this Recommendation, this is done in 2 steps:

Firstly, the level normalization to –26 dB is carried out by estimating the signal level in the range of 250 Hz to about 3000 Hz.

In the second step, a filter is applied in the frequency domain using an FFT filter over the length of the file. The filter characteristic is similar to the IRS receive characteristic given in ITU-T Rec. P.830.

The IRS filtered signal is used in the sections Sharp Declines, Robotization, Frame Repeats, Unnatural Beeps, Basic Voice Quality, Unnatural Silence, Local Background Noise and Global Background Noise.

9.1.3 Signal normalization and level calculation

Before being analysed, the signal is normalized to –26 dBov and filtered.

To normalize to –26 dBov, an estimate of the speech level is calculated based on the output from the VAD by calculating the RMS value of each active frame.

Once the signal level has been normalized, it is filtered using a 4th order Butterworth high-pass filter at 100 Hz cut-off frequency.

All the parameters that do not use the IRS filtered signal use this normalized signal, except mutes parameters which require the raw signal.

9.1.4 Variation of the speech section level

SpeechSectionsLevelVar is a descriptor of the level variation between sentences. The level of each speech section in the signal is computed. The difference between the maximum and the minimum level is calculated.

9.1.5 Local level variation

LocalLevelVar describes the energy variation of the signal. The RMS is calculated frame by frame. The first derivative of the RMS array is then calculated and power averages are extracted.

9.1.6 Pitch synchronous extraction

Pitch period length estimation and synchronous pitch mark labelling is required by multiple parts of the P.563 algorithm for extraction of parameters. The pitch synchronous extraction uses a hybrid temporal/spectral approach. Figure 4 shows an overall description of the algorithm.

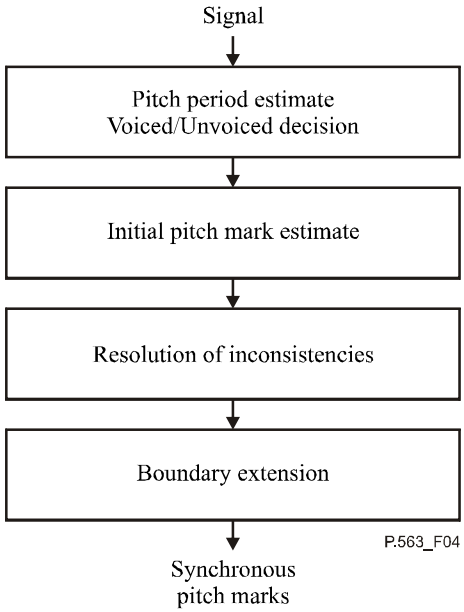


Figure 4/P.563 – Pitch synchronous extraction overview

9.1.6.1 Pitch period estimate and voice/unvoiced decision

Pitch period estimates are achieved through an autocorrelation method. The autocorrelation is calculated over 64 ms frames (512 elements), 50% overlapped.

The signal is first passed through a Hann window defined by:

$$y_i = 0.5x_i[1-\cos(w)] \text{ for } i = 0..(n-1)$$

where:

$$w = \frac{2\pi i}{n}$$

x is the input signal and

n is size of the frame

The autocorrelation is calculated using the inverse FFT of the power of the FFT:

$$R_{xx}(t) = y(t) \otimes y(t) = FFT^{-1}\left(FFT(y) \times FFT(y)^*\right)$$

R_{xx} is normalized by $R_{xx}(0)$, the maximum value of the calculated autocorrelation. The first 32 elements of the output array are filtered using half of a Hann window. The filtered autocorrelation is the n given by:

$$N_i = \frac{R_i}{R_0} \text{ for } i = 0,1,2,...31 \text{ and } N_i = \frac{R_i}{2R_0}[1-\cos(w)] \text{ for } i = 32,33,...,n-1$$

where:

$$w = \frac{2\pi i}{64}.$$

The maximum value is then searched between the 20th and the 100th elements, which covers a pitch range from 150 to 780 Hz. If this maximum is greater than 0.5, then the frame is classified as voiced and the candidate pitch length (T_{max}) is saved.

To avoid pitch doubling, a comparison is made with the pitch length of the last frame marked as voiced (T_{old}). T_{old} is firstly readjusted to make sure its position is a local peak, then if the following equation is validated,

$$N_{T_{old}} > \frac{1 + N_{T_{max}}}{3}$$

T_{max} is considered to be doubled, and T_{old} is set as the pitch period of the current frame.

9.1.6.2 Pitch mark placement

Once the pitch period is estimated, pitch mark labelling is performed. The method used is the cross-correlation of windowed speech segments with an impulse train. It is performed frame-by-frame to find the optimum pitch mark locations. The algorithm seeks the offset that gives the maximum absolute cross-correlation.

The cross-correlation is defined by: $Rxy_i = x_i \otimes yi$, with

$$x_i = 0.5 z_i [1 - \cos(w)], \text{ for } i = 0 \,..\, n$$

where:

$$w = \frac{2\pi i}{n}$$

z is the input signal

n is the frame size

$y_i = 1$ when i is a multiple of the pitch period (l)

$y_i = 0$ everywhere else

As a result, the equation can be simplified:

$$Rxy_i = \sum_{j=0}^{n/l} x_{i+j \cdot l}$$

where:

i is the offset

l the pitch length

Once the calculation is completed, i_{max} is set to be the index of the maximum of Rxy. Pitch marks are placed within the frame at regular intervals separated by the pitch length l, offset by i_{max}.

9.1.6.3 Resolution of inconsistencies

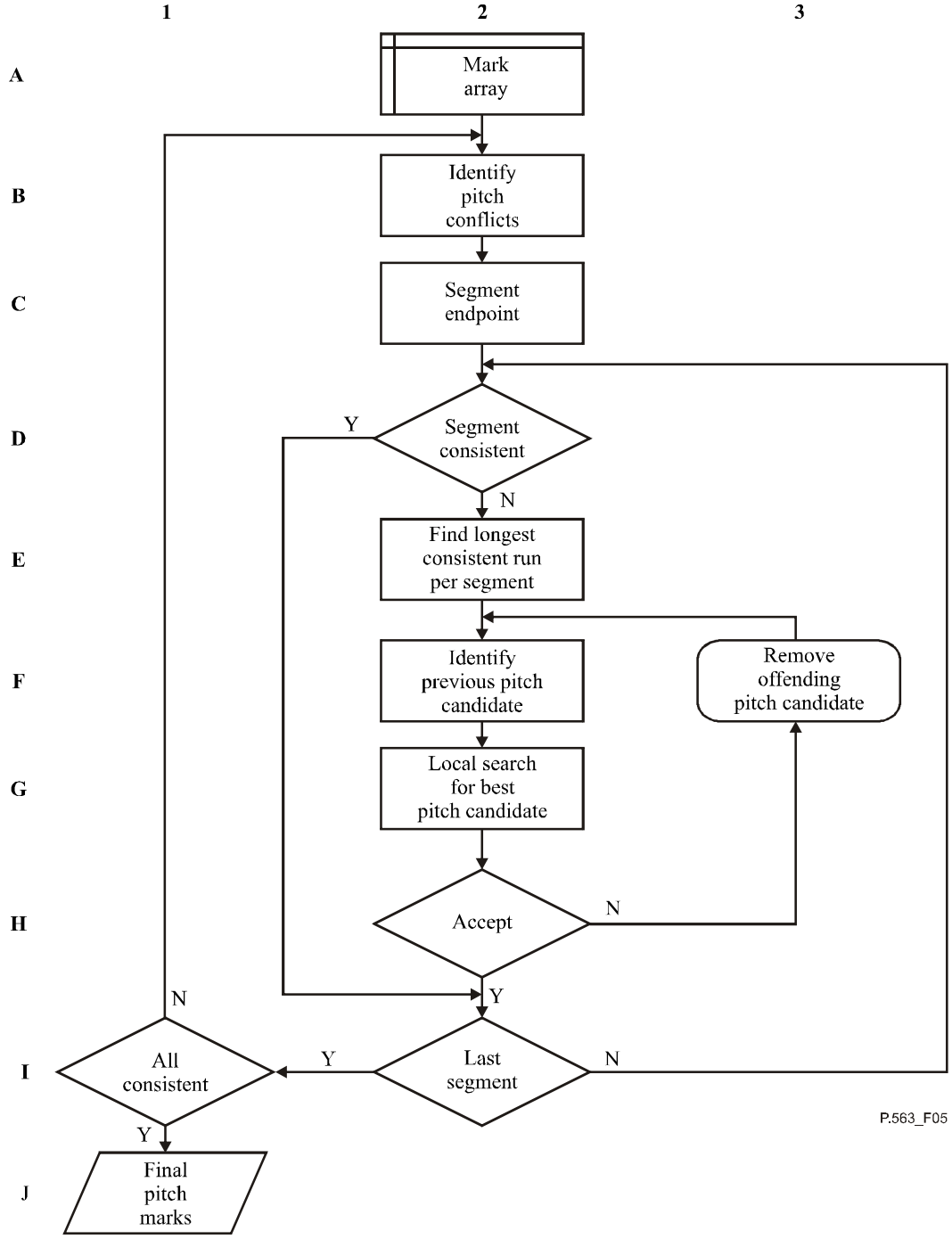


Figure 5/P.563 – Pitch-inconsistency handling

The combination of overlapping frames leads to multiple pitch marks within close proximity due to the inaccuracies of the cross-correlation time alignment. If two candidate pitch marks are separated by less than 10 samples, they are considered to be derived from the same pitch peak and are replaced by a single candidate at the most likely position of the two. A local search is conducted to find the best match with the waveform.

At this point, the voicing decision becomes important. The speech file is divided into distinct voiced segments for further processing. These may not be the final boundaries but are a necessary division for the next phase of processing. Adjacent pitch intervals are compared and those whose lengths differ by more than a given threshold are labelled as inconsistent.

Within each of the identified segments, the longest run of consistent candidate pitch marks is identified. This run of consistent pitching is then extended through the rest of the section. This will reinforce some candidate positions, remove some altogether and correct others as shown in Figure 5.

At the ends of the consistent pitch-run, the next pitch-mark is identified (Figure 5, F2). A local search is conducted to discover if this pitch mark would conform to the consistency threshold if placed more sensibly within the local waveform (Figure 5, G2). If no conformance is achieved, the pitch mark is removed, (Figure 5, F3), and the algorithm moves outwards to try to achieve consistency with the next pitch mark. In this way a consistent pitch run is achieved throughout all voiced segments.

9.1.6.4 Boundary review

At this point, the voiced sections have plausible candidate pitch marks, but may not be correct at the boundaries due to the frame-based nature of the initial voiced estimate. Figure 6 shows the algorithm used to assess and correct those marks at the boundaries of voiced sections.

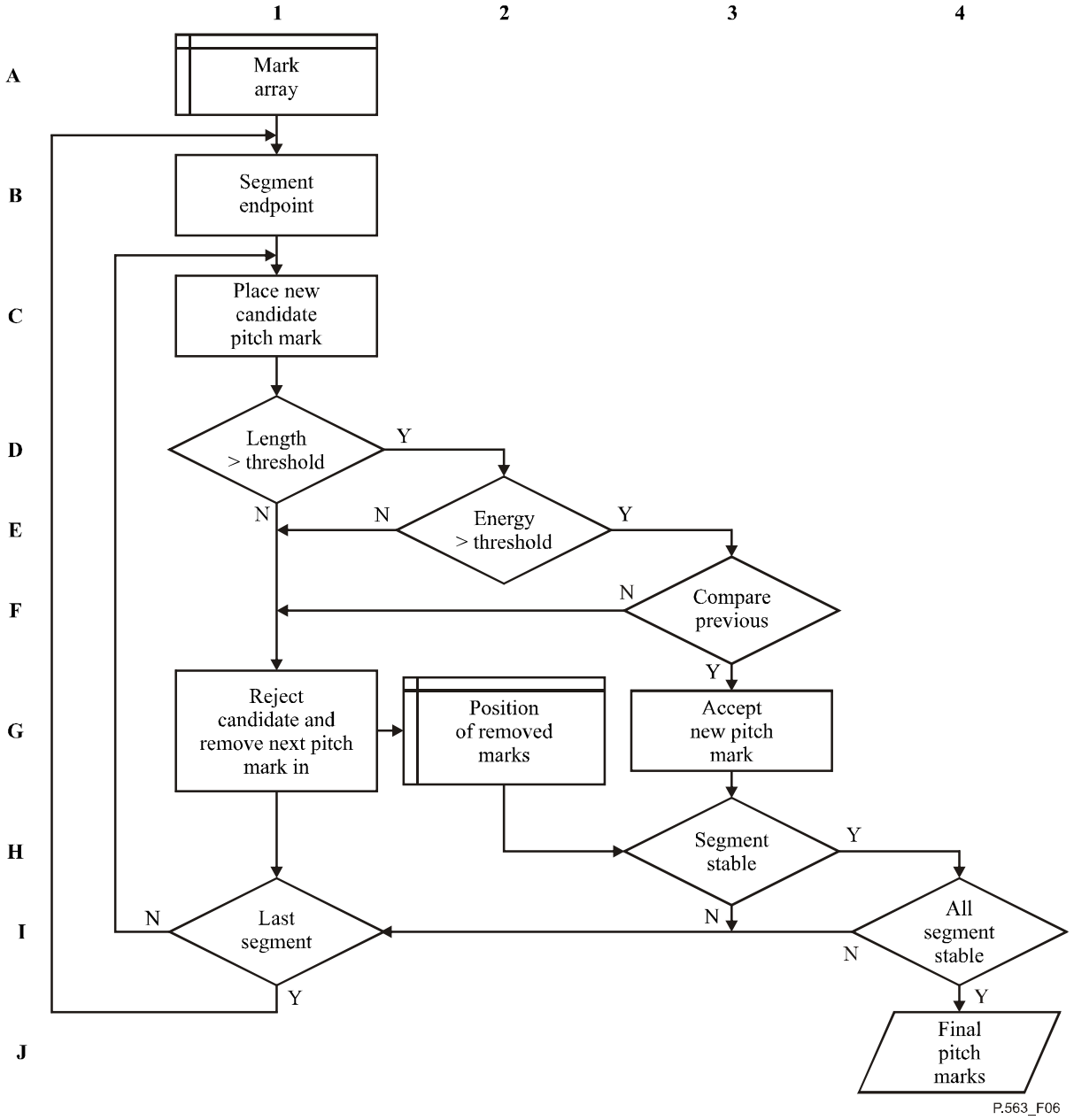


Figure 6/P.563 – Voiced segment boundary pitch review

At the segment boundaries a further candidate is placed outside the segment for assessment (Figure 6, C1). The properties of this newly defined pitch-cycle are compared with thresholds (Figure 6, D1, E2, F3). If it fails any of the tests it is rejected and the last known pitch mark is removed (Figure 6, G1). The position that this pitch mark is removed from is stored for subsequent stability assessment (Figure 6, G2). On the next iteration the pitch mark inside the old segment boundary comes under scrutiny. If the newly defined pitch cycle is within all defined thresholds, it is accepted as valid and the pitch mark is added (Figure 6, G3). In this way, the segments can be reduced and extended, sometimes joining segments together, according to the rules defined below. When a new pitch mark is accepted, a comparison is made with those previously removed marks (Figure 6, H3). When a match is found, the segment is stable and no further extension or reduction is required.

The rules used to define acceptability are:

- Length of cycle > Pre-determined threshold (Figure 6, D1);
- RMS energy of cycle > Pre-determined threshold (Figure 6, E2);
- Frequency content of new and current frame comparable (Figure 6, F3).

9.2 Description of the functional block 'Vocal tract analysis and Unnatural Voice'

The assignment of the degradations into the class unnatural voice and the quality scoring itself is based on the largest set of signal parameters in P.563. Basically, these signal parameters can be subdivided into two groups:

- Speech statistics;
- Vocal tract analysis.

9.2.1 Calculation of speech statistics for unnatural voice detection

The parameters used for speech statistics are mainly based on high order statistical evaluation of cepstral and LPC analyses, which are standard signal processing techniques. The two higher-order moments, kurtosis and skewness, are especially suitable for further analysis of the signal properties. These are classical measures of how un-gaussian the statistical signal is. Kurtosis measures the degree of fat tails of a distribution, whereas skewness measures the coefficient of asymmetry of a distribution. See Figure 7.

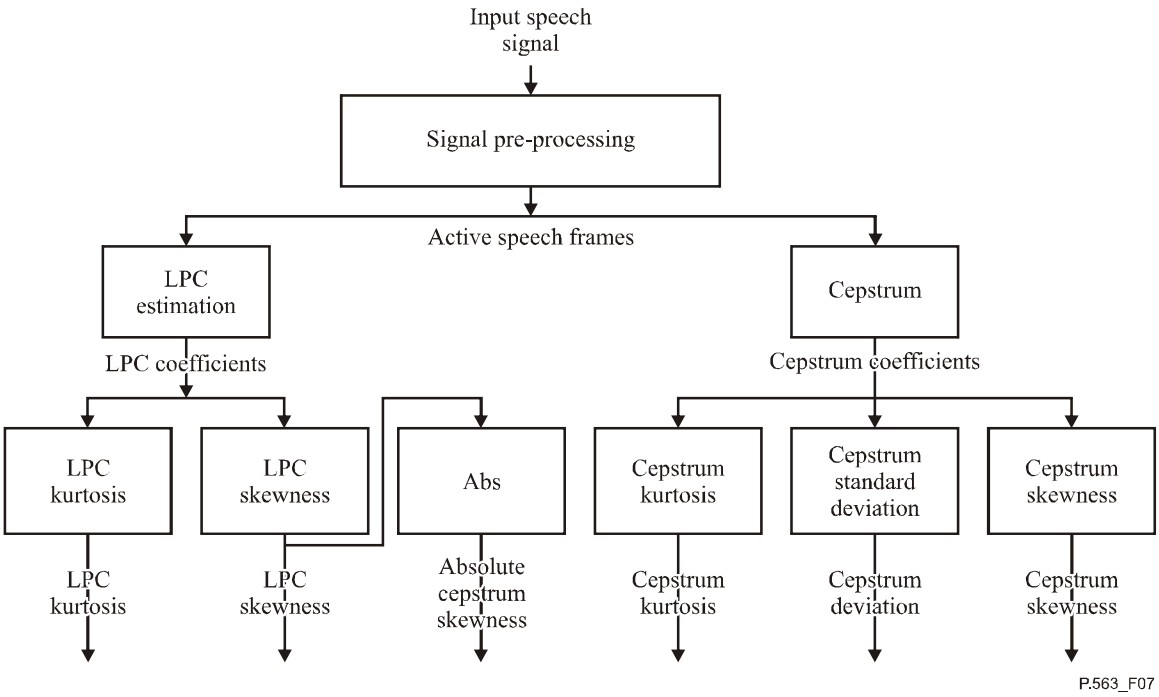


Figure 7/P.563 – Calculation of speech statistics

Skewness, the third central moment of a distribution of a set of data, is a measure of the asymmetry of the distribution of a real-valued random variable. Roughly speaking, a distribution has positive skew if the positive tail is longer, and negative skew if the negative tail is longer.

For a set of N values of a random variable *x*, Skewness 'ζ' is defined as:

$$\zeta = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{x_n - \bar{x}_N}{\sigma_N}\right)^3 \quad \left| \begin{array}{l} n = 1...N, \text{where } N \text{ is the length of the vector} \\ \sigma_N : \text{standard deviation of } x_1...x_N \end{array} \right.$$

Here are the skewness values for some typical distributions:

- Laplace, normal, uniform: 0;
- Exponential: 2.

Kurtosis, the fourth central moment of a distribution of a set of data, is a measure of how peaky the distribution of a real-valued random variable is. A normal Gaussian distribution has a kurtosis of zero. A distribution with positive kurtosis is called leptokurtic, and one with negative kurtosis platykurtic. For a vector of N values the kurtosis 'κ' is defined as:

$$\kappa = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{x_n - \bar{x}_N}{\sigma_N}\right)^4 - 3 \quad \left| \begin{array}{l} n = 1...N, \text{where } N \text{ is the length of the vector} \\ \sigma_N : \text{standard deviation of } x_1...x_N \end{array} \right.$$

where σ_N is the standard deviation and $\bar{x}_N$ is the arithmetic mean over *x*. The '–3' term at the end of this formula is defined as a correction to make the kurtosis of the normal distribution equal to zero.

Here are the kurtosis values for some typical distributions:

- Laplace: 3;
- Exponential: 6;
- Normal (Gaussian): 0.

9.2.1.1 Calculation of *LPCcurt* and *LPCskew*

The LPC kurtosis and the LPC skewness, as well as the absolute value of LPC skewness, are especially suitable for the estimation of the speech signal properties. The number of LP coefficients α_m used is 21, which is also the order of LPC evaluation.

The 'per frame' LPC standard deviation will be calculated as usual:

$$\sigma_n^{LPC} = \frac{1}{P}\left[\sum_{p=1}^{P}\alpha_p{}^2 - \frac{1}{P}\left(\sum_{p=1}^{P}\alpha_p\right)^2\right] \quad \left| \; p = 1...P, \text{ where } P \text{ is the number of coefficients} \right.$$

The LPC kurtosis and the LPC skewness are derived by evaluating the LPC vector of each active frame:

$$\kappa_n^{LPC} = \frac{1}{P}\sum_{p=1}^{P}\left(\frac{\alpha_p - \frac{1}{P}\sum_{p=1}^{P}\alpha_p}{\sigma_n^{LPC}}\right)^4 - 3 \quad \left| \begin{array}{l} p = 1...P, \text{ where } P \text{ is the number of coefficients} \\ n : \text{Number of current frame} \end{array} \right.$$

and

$$\zeta_n^{LPC} = \frac{1}{P}\sum_{p=1}^{P}\left(\frac{\alpha_p - \frac{1}{P}\sum_{p=1}^{P}\alpha_p}{\sigma_n^{LPC}}\right)^3 \quad \left| \begin{array}{l} p = 1...P, \text{ where } P \text{ is the number of coefficients} \\ n : \text{Number of current frame} \end{array} \right.$$

The 'per frame' results will be averaged over all active frames.

$$\overline{\kappa}^{LPC} = \frac{1}{N}\sum_{n=1}^{N}\kappa_n^{LPC} \quad \left| \begin{array}{l} n = 1...N, \text{where } N \text{ is the number of active frames} \\ \text{if } E_n \geq E_{minSpeech} \end{array} \right.$$

and

$$\overline{\zeta}^{LPC} = \frac{1}{N}\sum_{n=1}^{N}\zeta_n^{LPC} \quad \left| \begin{array}{l} n = 1...N, \text{where } N \text{ is the number of active frames} \\ \text{if } E_n \geq E_{minSpeech} \end{array} \right.$$

9.2.1.2 Calculation of *CepADev*, *CepCurt* and *CepSkew*

The 'per frame' cepstral standard deviation used, the second moment of the vector, will be calculated in a common way:

$$\sigma_n^{cep} = \frac{1}{M}\left[\sum_{m=1}^{M}c(m)^2 - \frac{1}{M}\left(\sum_{m=1}^{M}c(m)\right)^2\right] \quad \left| \begin{array}{l} \text{where } M \text{ is the length of the vector} \\ \text{and } c \text{ is the cepstrum coefficient} \end{array} \right.$$

This 'per frame' standard deviation will be averaged over all active frames:

$$\overline{\sigma}^{cep} = \frac{1}{N}\sum_{n=1}^{N}\sigma_n^{cep} \quad \left| \begin{array}{l} n = 1...N, \text{where } N \text{ is the number of active frames} \\ \text{if } E_n \geq E_{minSpeech} \end{array} \right.$$

The absolute value of that standard deviation forms the parameter *CepADev*.

The cepstrum kurtosis will be calculated for the cepstral coefficients c_m of each frame containing active speech. For each signal frame, where signal power E_i exceeds the minimum threshold $E_{minThres}$ is defined as an active speech frame.

$$\kappa_n^{cep} = \frac{1}{M}\sum_{m=1}^{M}\left(\frac{c_m - \frac{1}{M}\sum_{m=1}^{M}c_m}{\sigma_n^{cep}}\right)^4 - 3 \quad \left| \begin{array}{l} n = 1...M, \text{ where } M \text{ is the number of ceptstral coefficients} \\ n : \text{Number of current frame} \end{array} \right.$$

The 'per-frame' kurtosis κ_n is averaged over all active frames:

$$\overline{\kappa}^{cep} = \frac{1}{N}\sum_{n=1}^{N}\kappa_n^{cep} \quad \left| \begin{array}{l} n = 1...N, \text{where } N \text{ is the number of active frames} \\ \text{if } E_n \geq E_{minSpeech} \end{array} \right.$$

The 'per frame' cepstral skewness will be calculated in the same way as the kurtosis:

$$\zeta_n^{cep} = \frac{1}{M}\sum_{m=1}^{M}\left(\frac{c_m - \frac{1}{M}\sum_{m=1}^{M} c_m}{\sigma_n^{cep}}\right)^3 \quad \left| \begin{array}{l} n = 1...M, \text{ where } M \text{ is the number of ceptstral coefficients} \\ n : \text{Number of current frame} \end{array} \right.$$

These 'per frame' results will be also averaged over all active frames:

$$\overline{\zeta}^{cep} = \frac{1}{N}\sum_{n=1}^{N}\zeta_n^{cep} \quad \left| \begin{array}{l} n = 1...N, \text{where } N \text{ is the number of active frames} \\ \text{if } E_n \geq E_{minSpeech} \end{array} \right.$$

The resulting cepstral skewness describes the grade of distortion of the speech signal in a similar way as the cepstral kurtosis does. A lower value range 0 to 1 is responsible for highly degraded speech signals. Typical values for undistorted signals are in the range 2 to 4.

9.2.2 Calculation of vocal tract parameters for unnatural voice detection

In this module, the human vocal tract is modelled as a set of tubes of different lengths and time varying cross-sectional areas. From the speech signal, these cross-sectional areas are determined. These areas are then analysed for unnatural variations.

9.2.2.1 Vocal tract model extraction

The pitch-labelled speech stream enables the extraction of parameters from the voiced sections of speech on a variable time base. This allows more faithful representation of the speech than with a fixed-frame analysis that contains varying amounts of averaging depending on the speech event. This also means that the analysis is synchronized with the speech waveform production system, i.e., the human speech production organs.

The vocal tract parameters extraction requires the steps shown in Figure 8.

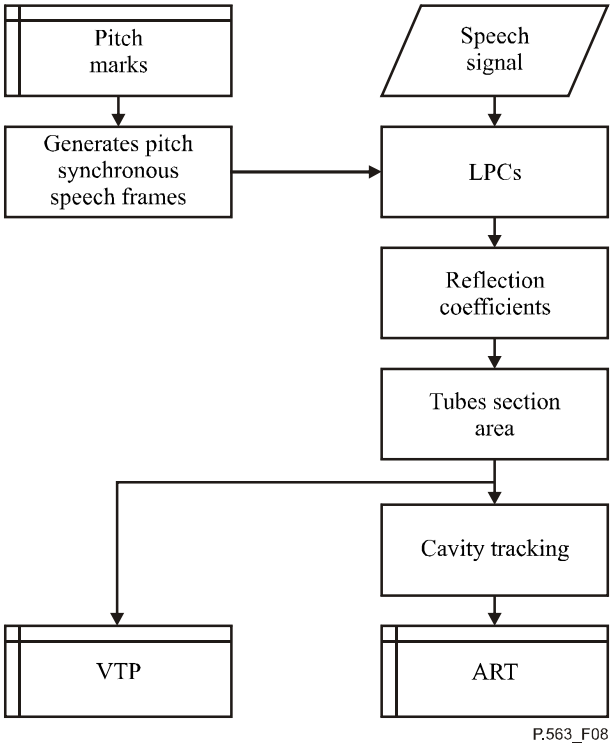


Figure 8/P.563 – Vocal Tract Parameters extraction overview

9.2.2.1.1 Speech frames

Instead of calculating LPCs frame by frame over a fixed length and at regular intervals, LPCs are performed on pitch synchronous frames, 50% overlapped. A speech frame always includes two consecutive pitch marks. The length of the frame is equal to twice the distance separating two consecutive pitch marks. The frame starts half of the pitch cycle length before the first mark, and finishes half of the pitch cycle length after the second pitch mark.

The following is an example, which is illustrated in Figure 9.

Consider 3 consecutive pitch marks, respectively placed at A, B and C.

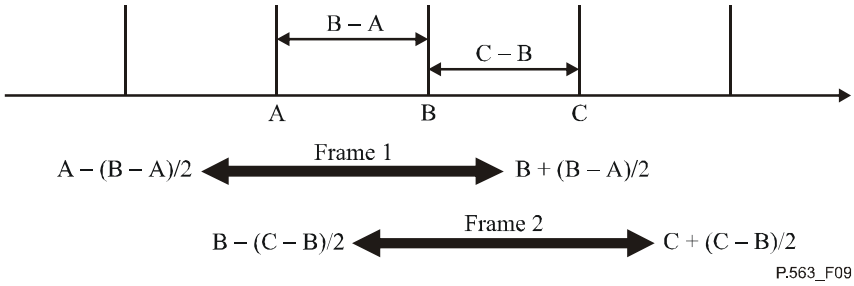


Figure 9/P.563 – Pitch synchronous LPC frames

Frame 1 includes pitch marks A and B.

The pitch cycle is equal to $(B - A)$, so the width of the frame is $2 \times (B - A)$, and it starts at $A - \frac{B - A}{2}$.

9.2.2.1.2 Linear Prediction Coefficients (LPC) calculation

The 8th order Linear Prediction Coefficients are extracted for each frame previously defined. A Hamming window is applied before calculating the autocorrelation function. Schur's recursion is used to obtain the predictor coefficients.

9.2.2.1.3 Tube section area

The section area, S_m, of the eight tubes is calculated using the reflection coefficients, μ, using the following equation:

$$S_m = \frac{1+\mu_m}{1-\mu_m} S_{m+1}, \qquad \mathrm{m} = \mathrm{M}, \mathrm{M}-1, ..., 1$$

Sections are calculated for each pitch cycle and the resulting array is called VTP, representing the tube sections of the vocal tract, from the glottis to the lips.

9.2.2.1.4 Cavity tracking

The VTP information is then reduced down to 3 parameters representing the front, middle and rear cavities. To generate these parameters, VTP tubes are grouped in the following way:

Cavity	Tubes
Rear	1,2,3
Middle	4,5,6
Front	7,8

Results are stored for each pitch mark in an array called ART (articulators).

9.2.2.2 Basic VTP descriptors

9.2.2.2.1 *VTPMaxTubeSection*

This parameter is the maximum section size of the first VTP tube over the whole input signal.

9.2.2.2.2 *FinalVtpAverage*

It represents the averaged section of the last VTP tube.

9.2.2.2.3 *ARTAverage*

It is the averaged section of the back cavity.

9.2.2.2.4 *ConsistentArtTracker*

This parameter describes how well the back and middle cavity correlate. For each pitch frame, the difference in section areas between the back and middle cavity is calculated. The length of smooth sections in the generated array is then calculated. Sections are considered to be consistent when consecutive elements do not vary by more than ±0.25. The length of each section is then averaged over the number of sections found and over the number of pitch cycles.

9.2.2.2.5 *VTPPeakTracker*

This parameter tracks the amplitude variations within the vocal tract. For each frame, the largest of the eight tubes is found. The derivative of the position array is calculated to determine the variations. The variation array is then averaged.

9.2.2.2.6 *VtpVadOverlap*

This parameter calculates the ratio of the total length of voiced sections within speech sections over the total length of speech sections.

9.2.3 Unnatural periodicity parameters

9.2.3.1 Pitch frames correlation

These two parameters are based on consecutive frames of voiced sections. Frames are determined similarly to the pitch synchronous LPC frames (see Figure 9).

- *PitchCrossPower* is determined by calculating the cross power between 2 frames for each consecutive frame. The peak to mean ratio is calculated for each cross power computation. The peak to mean array is then averaged.
- *PitchCorrelationOffset* is calculated as the cross-correlation of consecutive frames. For each pair of frames, the distance between the position of the maximum value and the middle of the frame is calculated. The averaged distance is then calculated.

9.2.3.2 Robotization

A voice signal that contains too much periodicity is declared to be robotic. Such signals are mostly the result of band-limitations such as those used in GSM networks.

To classify the signal as robotized, only frequencies in the range between 2200 Hz and 3300 Hz where found to be useful. Using the components of the signal in this range, the periodicity is calculated by a cross-correlation of adjacent time signal frames of 32 ms length. In this process, the next step classifies the frames as non-silent and periodic: a frame is non-silent, if its power exceeds 10^6. It is periodic if its respective cross-correlation exceeds 0.84 and its L_1 norm is at least 2.5^2.

If the percentage of periodic frames among the non-silent frames is greater than 3.4%, the module output value for robotization is computed by declaring at least 3.4% of all periodic frames as robotic frames.

9.2.3.3 Frame repeats

This module is trained to detect repeated frames in the signal. The algorithm makes use of the usual high cross-correlation of repeated frames. The detailed description of the algorithm is given as shown in Figure 10.

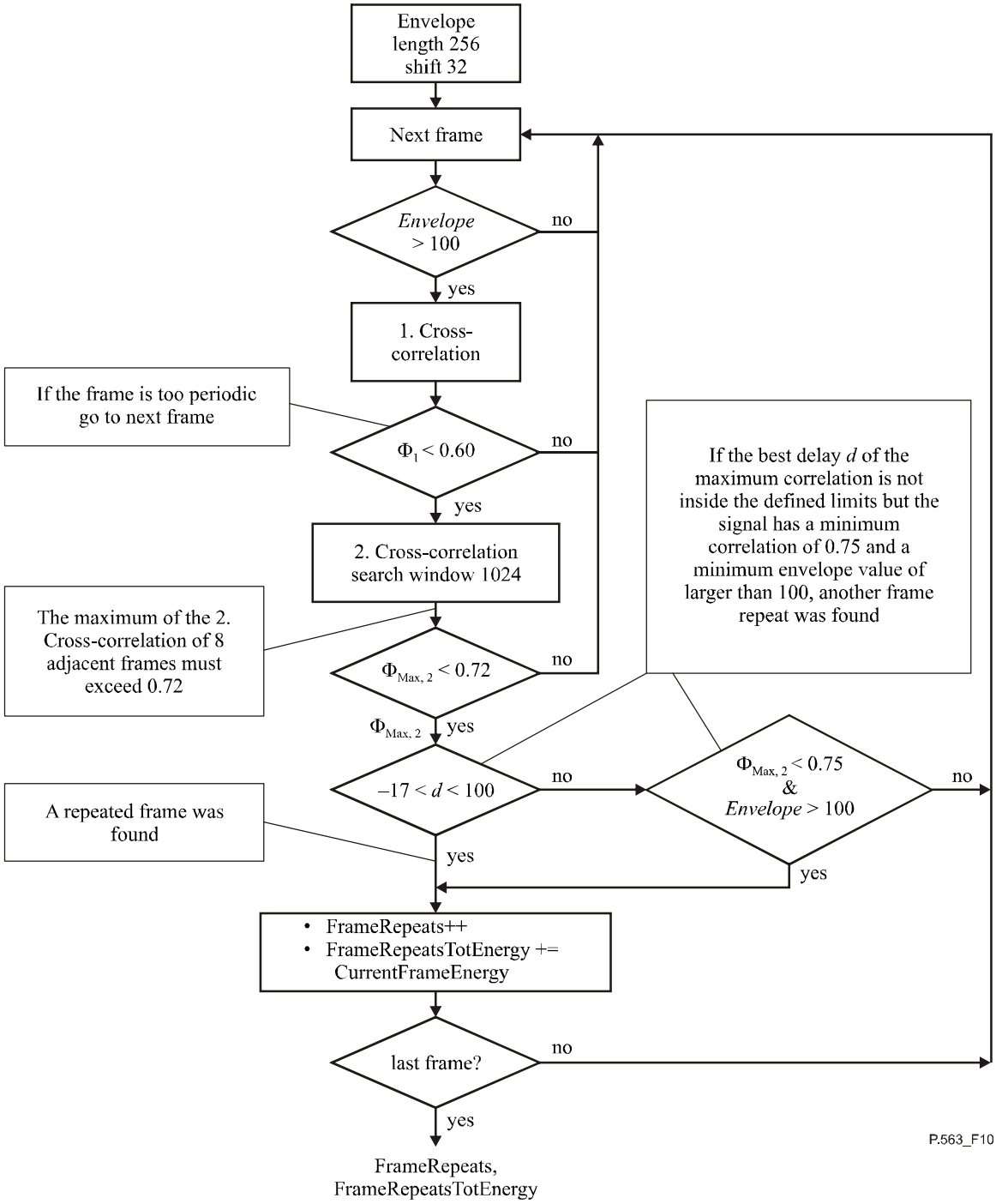


Figure 10/P.563 – Evaluation of frame repeats

The module's outputs are two values:

- *FrameRepeats*

  This value represents the number of detected frame repetitions in the current signal.

- *FrameRepeatsTotEnergy*

  This is the sum of energies of all detected repeated frames.

9.2.3.4 Unnatural beeps

When a signal is voiced it can be represented by a complex harmonic tone. However, a natural voice always includes voiced parts over a minimum length of time. In this signal descriptor, complex tones that have a short duration are declared as unnatural beeps.

The detector for unnatural beeps considers the periodicities in time signal frames of 32 ms duration over a total length of 160 ms with shifts of 16 ms. The periodicity itself is derived from the spectral unflatness in a frequency range between 250 and 1200 Hz. If the periodicities rise and fall within the duration of 160 ms, an unnatural beep has been detected and the module output values are calculated as shown below:

- *UnnaturalBeeps*

  This value represents the number of detected unnatural beeps multiplied by 1000 and divided by the number of processed samples.

- *UnnaturalBeepsMean*

  Mean sum of energies of the frames that have been found to contain beeps.

- *UnnaturalBeepsAffectedSamples*

  Sum of samples of the frames that contain beeps.

9.2.4 Basic voice quality

The basic quality module evaluates the influence of distortions with the help of a psychoacoustic model. Figure 11 shows the principle interworking of the two function blocks which are namely the speech enhancer and a perceptually based intrusive speech quality measurement.

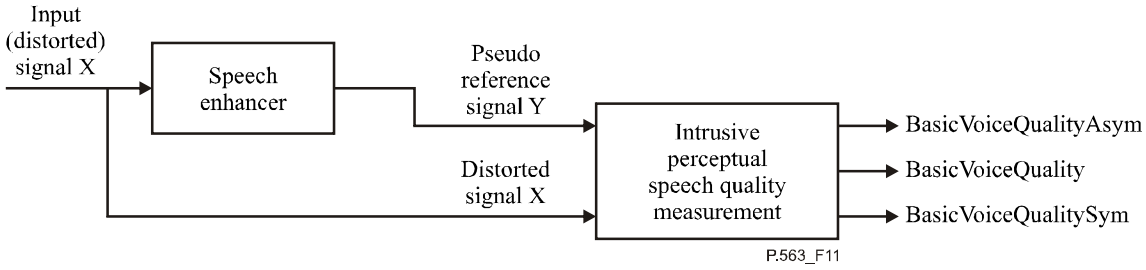


Figure 11/P.563 – Scheme for calculation of basic voice quality

9.2.4.1 The speech enhancer

The speech enhancer can be divided into three logical modules, as shown in Figure 12.

1) LPC-Analysis: Using the Levinson-Durbin-algorithm the time signal is analysed yielding the signal's residue and 10 LPC-coefficients.

2) Vocal-tract model: The LPC-coefficients are modified to fit into a vocal-tract model of a human talker.

3) LPC-Synthesis: The residue plus the modified LPC-coefficients are brought together again to rebuild a voice-enhanced time-signal.

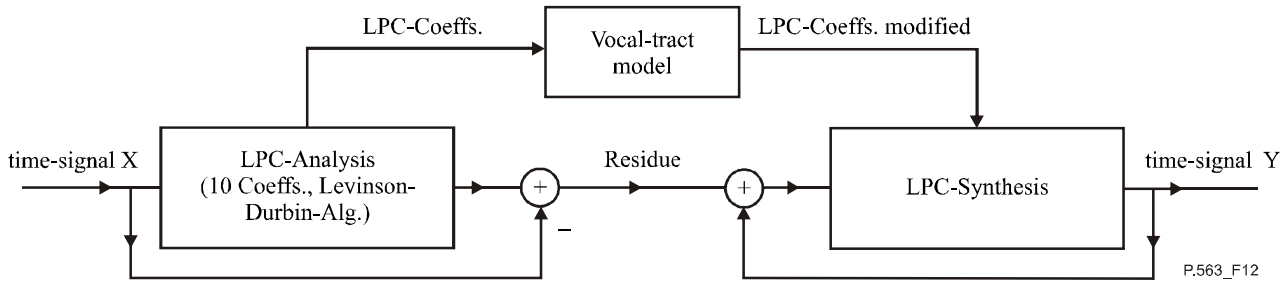


Figure 12/P.563 – Scheme of the Speech Enhancer

9.2.4.2 Intrusive perceptual speech quality measurement – Basic voice quality

The intrusive perceptual speech quality measurement is complex. Therefore, a full description is not provided here and the reader is referred to the C source code for a detailed description. Figures 13 and 14 give an overview of the algorithm in the form of a block diagram. The core of the perceptual model and the final determination of the Basic Voice Quality are presented. For each of the blocks a high-level description is given.

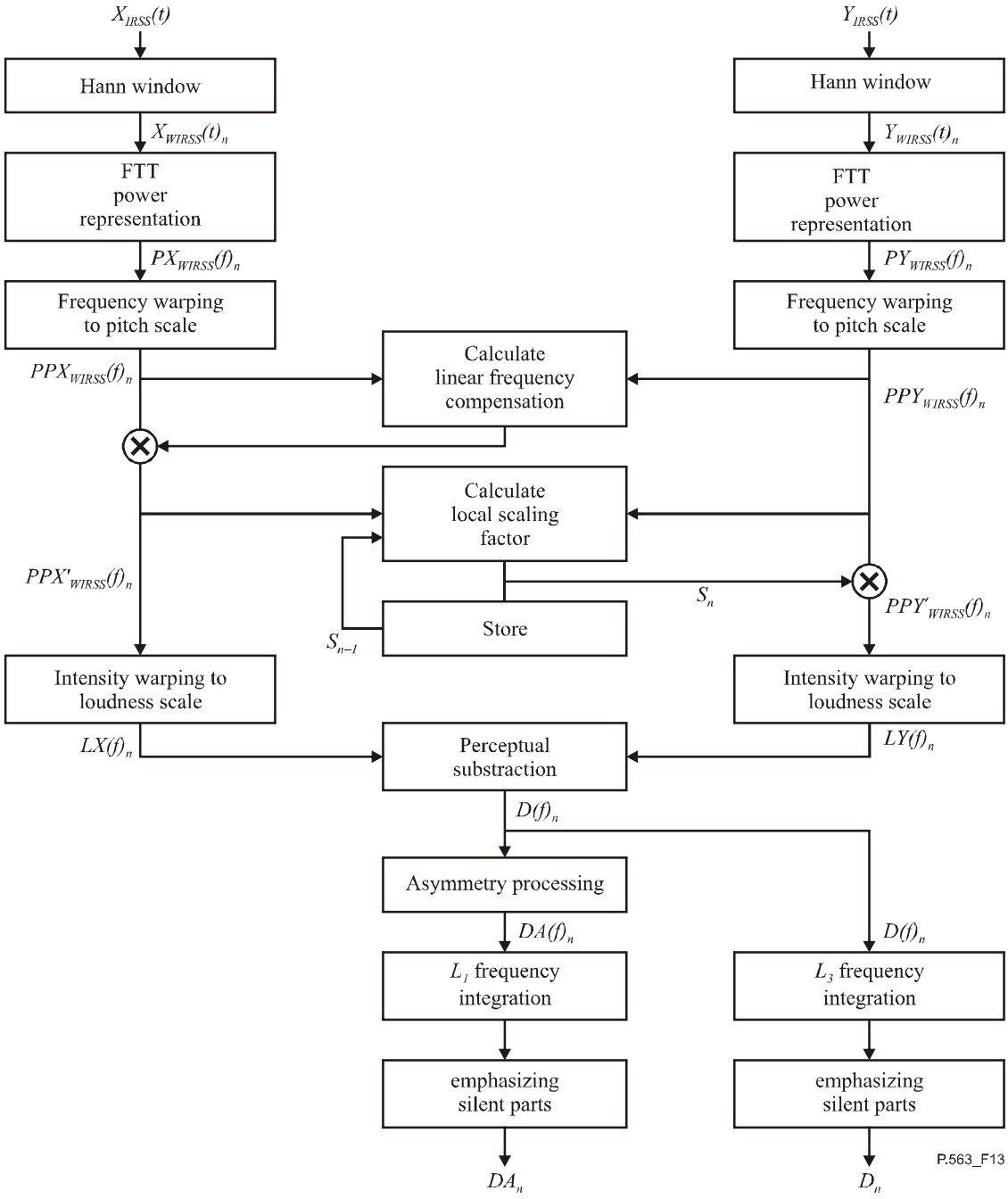


Figure 13/P.563 – Overview of the perceptual model

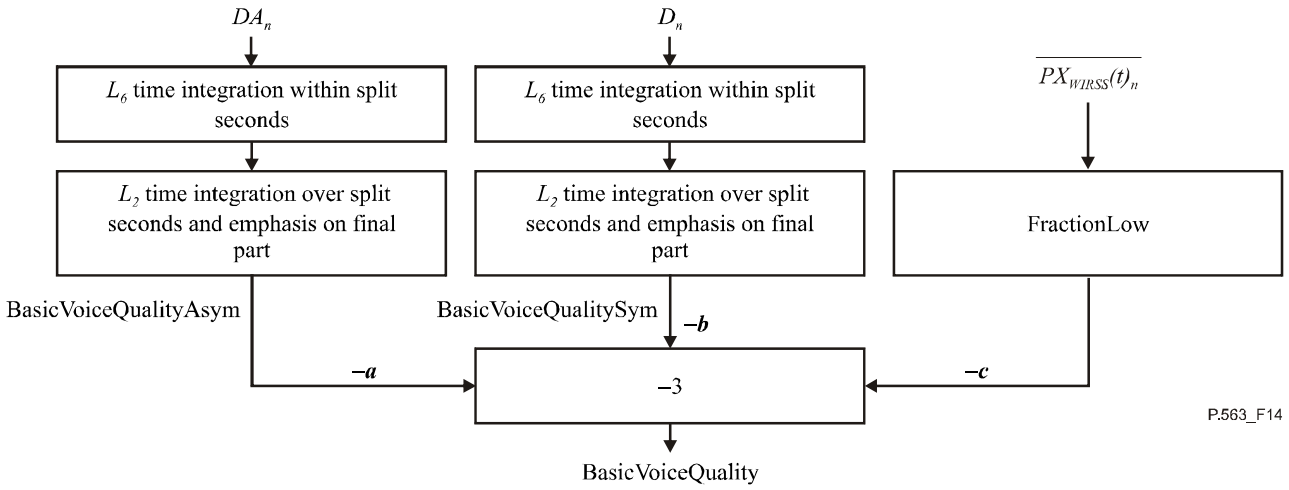


Figure 14/P.563 – Overview of the perceptual model – Integration part

9.2.4.2.1 FFT window size

The time signals are mapped to the time-frequency domain using a short-term FFT with a Hann window of size 32 ms (8 kHz sampling rate) and an overlap of 50%.

9.2.4.2.2 Absolute hearing threshold

The absolute hearing threshold $P_0(f)$ is interpolated to get the values at the centre of the Bark bands that are used. These values are stored in an array and are used in Zwicker's loudness formula.

9.2.4.2.3 The power scaling factor

There is an arbitrary gain constant following the FFT for time-frequency analysis. This constant is computed from a sine wave of a frequency of 1000 Hz with an amplitude at 29.54 (40 dB SPL) transformed to the frequency domain using the windowed FFT over 32 ms. The (discrete) frequency axis is then converted to a modified Bark scale by binning of FFT bands. The peak amplitude of the spectrum binned to the Bark frequency scale (called the "pitch power density") must then be 10 000 (40 dB SPL). The latter is enforced by a postmultiplication with a constant, the power scaling factor S_p.

9.2.4.2.4 The loudness scaling factor

The same 40 dB SPL reference tone is used to calibrate the psychoacoustic (Sone) loudness scale. After binning to the modified Bark scale, the intensity axis is warped to a loudness scale using Zwicker's law, based on the absolute hearing threshold. The integral of the loudness density over the *Bark* frequency scale, using a calibration tone at 1000 Hz and 40 dB SPL, must then yield a *value of* 1 Sone. The latter is enforced by a postmultiplication with a constant, the loudness scaling factor S_l.

9.2.4.2.5 Computation of the active speech time interval

If the pseudo-enhanced and degraded speech file start or end with large silent intervals, this could influence the computation of certain average distortion values over the files. Therefore, an estimate is made of the silent parts at the beginning and end of these files. The sum of five successive absolute sample values must exceed 500 from the beginning and end of the pseudo-enhanced speech file in order for that position to be considered as the start or end of the active interval. The interval between this start and end is defined as the active speech time interval. In order to save computation cycles and/or storage size, some computations can be restricted to the active interval.

9.2.4.2.6 Short-term Fast Fourier Transform

The time-frequency transformation is implemented by a short-term FFT with a window size of 32 ms. The overlap between successive time windows (frames) is 50 per cent. The power spectra, the sum of the squared real and squared imaginary parts of the complex FFT components, are stored in separate real valued arrays for the pseudo-enhanced and degraded signals. Phase information within a single Hann window is discarded and all calculations are based on only the power representations $PX_{WIRSS}(f)_n$ and $PY_{WIRSS}(f)_n$.

9.2.4.2.7 Calculation of the pitch power densities

The Bark scale reflects that at low frequencies, the human hearing system has a finer frequency resolution than at high frequencies. This is implemented by binning FFT bands and summing the corresponding powers of the FFT bands with a normalization of the summed parts. The resulting signals are known as the pitch power densities $PPX_{WIRSS}(f)_n$ and $PPY_{WIRSS}(f)_n$.

9.2.4.2.8 Partial compensation of the pseudo-enhanced pitch power density for transfer function equalization

To deal with filtering in the system under test, the power spectrum of the pseudo-enhanced and degraded pitch power densities are averaged over time. This average is calculated over speech active frames using time-frequency cells whose power is more than 1000 times the absolute hearing threshold. For each modified Bark bin, a partial compensation factor is calculated from the ratio of the degraded spectrum to the pseudo-enhanced spectrum. The maximum compensation is never more than 20 dB. The pseudo-enhanced pitch power density $PPX_{WIRSS}(f)_n$ of each frame n is then multiplied with this partial compensation factor to equalize the pseudo-enhanced to the degraded signal. This results in an inversely filtered pseudo-enhanced pitch power density $PPX'_{WIRSS}(f)_n$.

This partial compensation is used because severe filtering can be disturbing to the listener.

9.2.4.2.9 Partial compensation of the distorted pitch power density for time-varying gain variations between distorted and pseudo-enhanced signal

Short-term gain variations are partially compensated by processing the pitch power densities frame-by-frame. For the pseudo-enhanced and the degraded pitch power densities, the sum in each frame n of all values that exceed the absolute hearing threshold is computed. The ratio of the power in the pseudo-enhanced and the degraded files is calculated and bounded to the range $[3 \cdot 10^{-4}, 5]$. A first-order low-pass filter (along the time axis) is applied to this ratio. The distorted pitch power density in each frame, n, is then multiplied by this ratio, resulting in the partially gain compensated distorted pitch power density $PPY'_{WIRSS}(f)_n$.

9.2.4.2.10 Calculation of the loudness densities

After partial compensation for filtering and short-term gain variations, the pseudo-enhanced and degraded pitch power densities are transformed to a Sone loudness scale using Zwicker's law.

$$LX(f)_n = S_l \cdot \left(\frac{P_0(f)}{0.5} \right)^{\gamma} \cdot \left[\left(0.5 + 0.5 \cdot \frac{PPX'_{WIRSS}(f)_n}{P_0(f)} \right)^{\gamma} - 1 \right]$$

with $P_0(f)$ the absolute threshold and S_l the loudness scaling factor from 9.2.4.2.4.

Above 4 Bark, the Zwicker power, γ, is 0.23. Below 4 Bark, the Zwicker power is increased slightly to account for the so-called recruitment effect. The resulting two-dimensional arrays $LX(f)_n$ and $LY(f)_n$ are called loudness densities.

9.2.4.2.11 Calculation of the disturbance density

The signed difference between the distorted and pseudo-enhanced loudness density is computed. When this difference is positive, components such as noise have been added. When this difference is negative, components have been omitted from the pseudo-enhanced signal. This difference array is called the raw disturbance density.

The minimum of the pseudo-enhanced and degraded loudness density is computed for each time-frequency cell. These minima are multiplied by 0.25. The corresponding two-dimensional array is called the mask array. The following rules are applied in each time-frequency cell:

- If the raw disturbance density is positive and larger than the mask value, the mask value is subtracted from the raw disturbance.
- If the raw disturbance density lies in between plus and minus the magnitude of the mask value, the disturbance density is set to zero.
- If the raw disturbance density is more negative than the product of the mask value and –1, the mask value is added to the raw disturbance density.

The net effect is that the raw disturbance densities are pulled towards zero. This represents a dead zone before an actual time frequency cell is perceived as distorted. This models the process of small differences being inaudible in the presence of loud signals (masking) in each time-frequency cell. The result is a disturbance density as a function of time (window number n) and frequency, $D(f)_n$.

9.2.4.2.12 Cell-wise multiplication with an asymmetry factor

The asymmetry effect is caused by the fact that, when a codec distorts the input signal, it will, in general, be very difficult to introduce a new time-frequency component that integrates with the input signal, and the resulting output signal will thus be decomposed into two different components, the input signal and the distortion, leading to clearly audible distortion. When the codec leaves out a time-frequency component, the resulting output signal cannot be decomposed in the same way and the distortion is less objectionable. This effect is modelled by calculating an asymmetrical disturbance density $DA(f)_n$ per frame by multiplication of the disturbance density $D(f)_n$ with an asymmetry factor. This asymmetry factor equals the ratio of the distorted and pseudo-enhanced pitch power densities raised to the power of 1.2. If the asymmetry factor is less than 3, it is set to zero. If it exceeds 12, it is clipped at that value. Thus, only those time-frequency cells remain, as non-zero values, for which the degraded pitch power density exceeded the pseudo-enhanced signal pitch power density.

9.2.4.2.13 Aggregation of the disturbance densities over frequency and emphasis on soft parts of the pseudo-enhanced signal

The disturbance density $D(f)_n$ and asymmetrical disturbance density $DA(f)_n$ are integrated (summed) along the frequency axis using two different *Lp* norms and a weighting on soft frames (having low loudness):

$$D_n = M_n \sqrt[3]{\sum_{f=1,\ldots Number\,of\,Bark\,bands} \left(\left|D(f)_n\right| W_f\right)^3}$$

$$DA_n = M_n \sum_{f=1,\ldots Number\,of\,Bark\,bands} \left(\left|DA(f)_n\right| W_f\right)$$

with M_n a multiplication factor, *1/(power of pseudo-enhanced frame plus a constant)*$^{0.04}$, resulting in an emphasis of the disturbances that occur during silences in the pseudo-enhanced speech fragment, and W_f a series of constants proportional to the width of the modified Bark bins. After this

multiplication, the frame disturbance values are limited to a maximum of 45. These aggregated values, D_n and DA_n, are called frame disturbances. They are used to calculate the perceived quality.

9.2.4.2.14 Aggregation of the disturbance within split second interval

Next, the frame disturbance values and the asymmetrical frame disturbance values are aggregated over split second intervals of 20 frames (accounting for the overlap of frames: approx. 320 ms) using L_6 norms. These intervals also overlap 50 per cent and no window function is used.

9.2.4.2.15 Aggregation of the disturbance over the duration of the speech signal, including a recency factor

The split second disturbance values and the asymmetrical split second disturbance values are aggregated over the active interval of the speech files (the corresponding frames) now using L_2 norms. The higher norm value for the aggregation within split-second intervals as compared to the lower norm value of the aggregation over the speech file is due to the fact that when parts of the split seconds are distorted, that split second loses meaning whereas, if a first sentence in a speech file is distorted, the quality of other sentences remains intact.

9.2.4.2.16 Computation of the Basic Voice Quality

The final output value (*BasicVoiceQuality*) is a linear combination of the average disturbance value (*BasicVoiceQualitySym*), the average asymmetrical disturbance value (*BasicVoiceQualityAsym*) and the fraction (*FractionLow*) of the averaged power spectrum in the frequency range between 20 and 170 Hz over the total averaged power spectrum in dB. The range of the Basic Voice Quality is 1 to 11.

The output values of this model are:

- *BasicVoiceQuality*

  This value represents an estimate of the audible disturbance.

- *BasicVoiceQualityAsym*

  This value is equivalent to the value after the integration of the asymmetric frame disturbance.

9.3 Description of the functional block 'Additive Noise'

The assignment of the degradations into the class additive noise and the quality scoring itself is based on two subsets of parameters to be calculated:

- Static Noise Level and SNR; and
- Multiplicative Noises and Segmental SNR.

9.3.1 Calculation of parameters describing the static noise background

9.3.1.1 SNR Calculation

The signal-to-noise level ratio (SNR) algorithm uses the VAD output. Because the VAD is calculated using level thresholds, the VAD will be less accurate with high speech activity signals. As a result, the estimation of the SNR is different depending on the speech activity. This is shown in Figure 15.

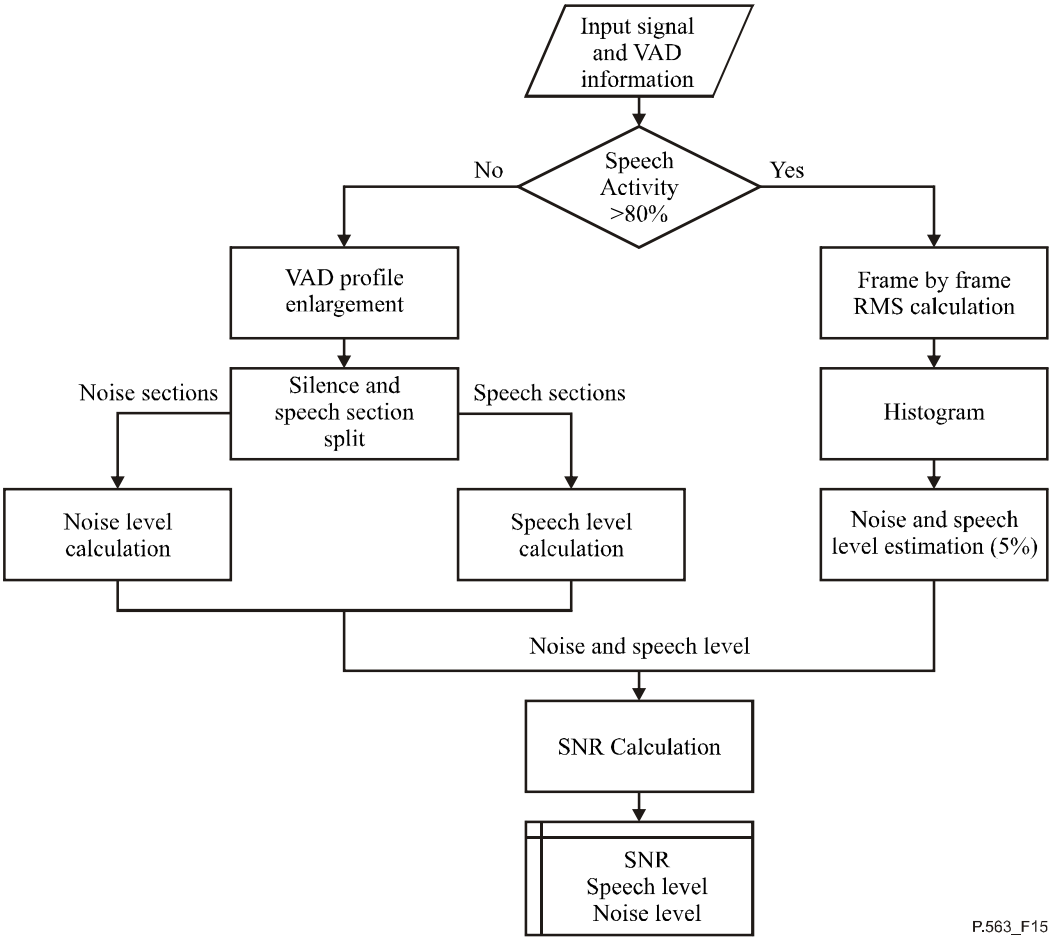


Figure 15/P.563 – SNR Calculation overview

Speech activity less than or equal to 80%

The noise and speech levels are estimated using the VAD profile. Speech and noise frames are dissociated and the RMS values of both groups are calculated. To ensure that none of the speech is included in the noise level calculation, both edges of the speech sections are extended using the following rule:

Consider two consecutive speech sections A and B.

- If they are separated by more that 2 seconds, the right edge of A and the left edge of B are extended by 0.5 seconds.
- If they are separated by less than 2 seconds, 1/8 of the gap length is added on the right side of A and left side of B.
- The beginning of the signal array is considered as the end of a speech section.
- The end of the signal array is considered as the beginning of a speech section.

Speech activity greater than 80%

As the speech activity becomes too high to be able to calculate the noise level correctly, a statistical method is used. The RMS of the input signal is processed frame by frame (4 ms), and the maximum RMS value is calculated (RMS_{max}). A discreet histogram of the RMS signal is built using 5000 bins and intervals equal to $\Delta = \frac{RMS_{max}}{5000}$.

The noise level is estimated by calculating the average of the 5% lowest RMS values.

The speech level is estimated by calculating the average of the remaining 95% of the RMS values.

SNR, noise and speech levels calculation

The previous calculation estimated the RMS values of the speech and noise sections. The SNR (in dB) is deduced as followed:

$$SNR_{dB} = 20\log\left(\frac{RMS_{Speech}}{RMS_{Noise}}\right) = 20\log\left(RMS_{Speech}\right) - 20\log\left(RMS_{Noise}\right)$$

Noise and speech levels are returned in dBov relative to the maximum level a signed 16-bit signal could reach, $dBov_{ref} = 90.3$.

As a result,

$$NoiseLevel_{dBov} = 20\log\left(RMS_{Noise}\right) - dBov_{ref}$$

$$SpeechLevel_{dBov} = 20\log\left(RMS_{Speech}\right) - dBov_{ref}$$

9.3.1.2 Local background noise

The term local background noise refers to the noise between phonemes. A phoneme is defined as an interval for which an envelope's value doubles within 100 ms and decreases to its near original value within 400 ms. The envelopes are RMS-values calculated on non-overlapping frames of 20 ms length.

The basic assumption of this module is that any interval of normal speech of 1 second contains at least 4 start or stop events, which are marked as utterances. If the number of start or stop events in 1 second is less than 4, it is likely that the investigated interval does not only contain phonemes but also background noise.

The model's output values are derived from the 200 ms subinterval for which the signal's energy is minimal in the specified one second window, as it supposedly represents the samples where local background noise is located.

- *LocalBGNoise*

  Percentage of samples that were classified as local background noise compared with the total number of samples in the signal.
- *LocalBGNoiseMean*

  Mean of the energies of the frames that have been found to contain local background noise.
- *LocalBGNoiseLog*

  LocalBGNoiseMean in dB: *LocalBGNoiseLog* = 10log(|*LocalBGNoiseMean*| +1)
- *LocalBGNoiseAffectedSamples*

  Number of samples of the frames that contain local background noise.

9.3.1.3 Global background noise

It is assumed that for each speech file, a minimum fraction of time does not contain speech but background noise. In order to detect this background noise, the signal is subdivided into 32 ms frames, each shifted by 16 ms. For each of these frames an RMS-value (envelope) of the signal is calculated. All envelopes in the current envelope search window (14 frames to the left of the current frame and 14 frames to the right) are set to the RMS value of the current envelope if they have smaller values than that and if they are larger than zero. This results in a smoothened sequence of envelopes.

The frames containing global background noise are now defined as the frames with the smallest envelopes but only up to a certain fraction of the total time signal. The fraction is given by 7% of the signal time minus 1 second[3] and minus at most 50% of the fraction that has already been classified to be local background noise. The proportion that is subtracted is dependent on the amount of local background noise.

- *GlobalBGNoise*

  Percentage of samples that were classified as global background noise compared with the total number of samples in the signal.

- *GlobalBGNoiseAffectedSamples*

  Number of samples of the frames that contain global background noise.

9.3.1.4 High frequencies flatness analysis

HiFreqVar describes the presence of high frequencies in the input signal introduced by the noise. The signal is analysed in speech sections only, extracting a descriptor of the spectral flatness in the 2500-3500 Hz band. The method used is shown in Figure 16.

[3] It is assumed that one can talk for one second without breathing.

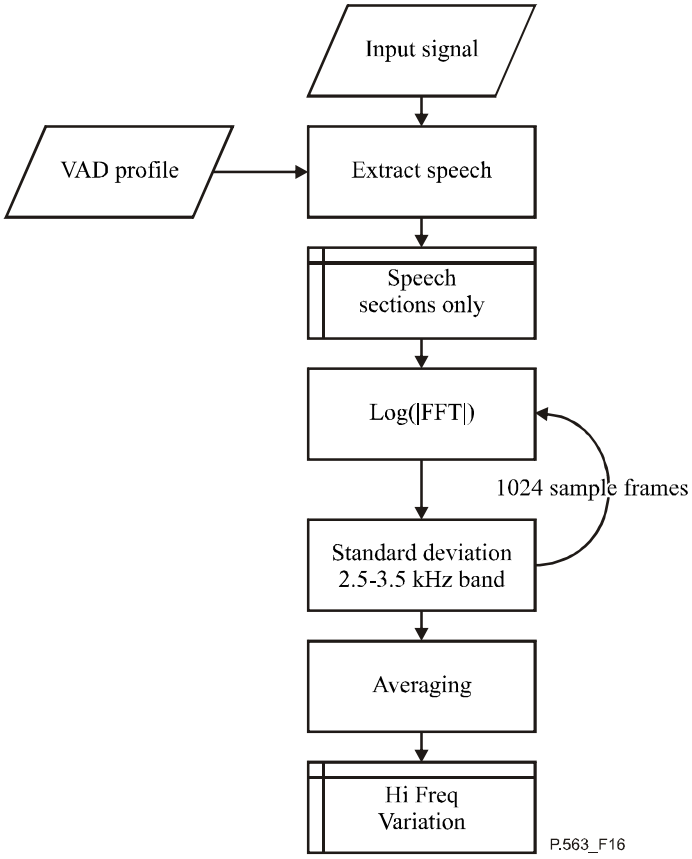


Figure 16/P.563 – High frequency analysis

The first step in the process is the extraction of speech sections from the input signal using the VAD profile. The logarithm of the modulus of FFT from the speech sections is then calculated using 1024 sample frames. For each frame, the standard deviation of the 2.5-3.5 kHz band is calculated. *HiFreqVar* is generated by averaging the standard deviation array.

9.3.1.5 Spectral clarity analysis

This parameter is generated from the portions of signal which have been marked as voiced and for which the synchronous pitch marks have been extracted. The main steps taken in generating this parameter are shown in Figure 17.

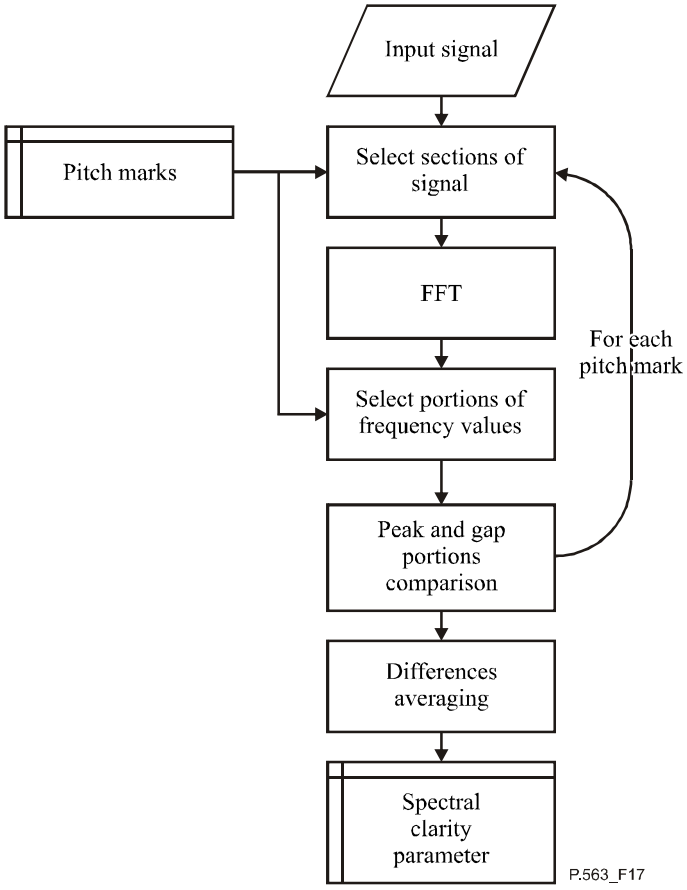


Figure 17/P.563 – Spectral clarity analysis

A section comprising 512 samples is selected such that a pitch mark, P, is central in the selected section. A Blackman Harris window is then applied to the portion and a log of the modulus of the FFT is calculated.

For each pitch mark, an instant pitch is calculated by taking the minimum of the distance between P and P – 1, and the distance between P and P + 1. The length is used to estimate the location of the fundamental H0 in the corresponding FFT frame. Harmonics (H1-H5) are estimated to occur at multiples of the pitch frequency estimate H0.

Portions comprising a frequency range of half the pitch frequency estimate are selected. The centre frequency of the portions selected are equal either to a frequency value of a harmonic (peak portions), or to a frequency value half way between two harmonics (gap portions).

The energy of peak and gap portions is averaged, and the difference is calculated.

The operation is executed on each pitch mark, and the spectral clarity parameter is calculated by averaging the differences computed for each frame.

9.3.1.6 Estimated background noise (*EstBGnoise*)

The estimated background noise floor is expressed in dBov. It is calculated based on a histogram of the RMS amplitude values of the signal. The histogram shows which amplitudes are most common; the more signal there is at a particular amplitude, the higher the histogram value will be for that amplitude. The result is a number of occurrences of a value in a data set. A histogram table presents

the energy-grade boundaries and the number of scores between the lowest bound and the current bound.

At first, the levels in all frames of the signal will be evaluated. The frame power and the corresponding levels are calculated on the frame length, 16 samples or 2 ms.

A histogram evaluates the individual frequencies for the calculated noise levels and is the basis for the derivation of the noise floor.

Position of the mean in a histogram is performed by the following formula:

$$HistogramMean = MAX[0, MIN(Hist\_size - 1, ((Hist\_size - 1) \cdot (hist\_mean - least)/(high - least)))]$$

where *least* is the lowest energy value and *high* the highest value of the histogram,

where *Hist*_size is set to 50, and $hist\_mean = \frac{sum\_of\_energy}{NrFrames}$.

The Noise floor, also known as the "bound position", is calculated as a minimum value between a noise peak (left from mean of histogram) and a mean itself. A search for "bound position" is started from noise peak to the direction of the mean value of the histogram. A search is stopped (noise floor found) when the histogram value at the bound position is lower than 20% of the value at the peak position.

If the search for "bound position" has failed, it will be stopped at least 10 dB from "max position" (noise position).

An example of the histogram is shown in Figure 18 for a noisy speech signal of 10 seconds duration. The histogram is presented with 50 bins between minimum and maximum level values. There are two maximum values, one for noise and one for speech active intervals. In our example, the first maximum is found at –45.1 dB, which is noise level. Second peak is at about –26 dBov which is speech active level. A noise floor (bound position) is detected at –37.2 dBov.

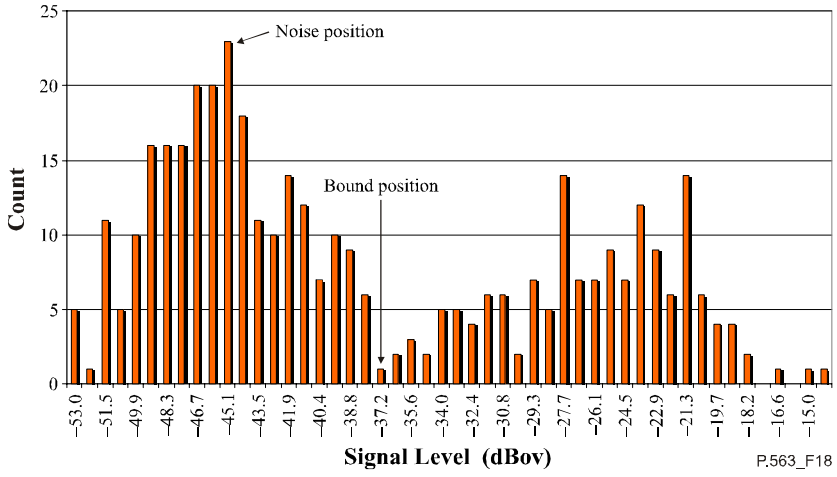


Figure 18/P.563 – Level histogram used for noise threshold measurement

9.3.2 Calculation of parameters describing multiplicative noises

This functional block detects multiplicative noise and produces main indicating values for describing the distortions in the speech signal. Multiplicative noise could be introduced by (mainly cascaded) logarithmic PCM and ADPCM systems as well as by signal-form speech-codecs. In addition such distortions will be produced by MNRU simulations according to ITU-T Rec. P.48, which are commonly used in auditory tests.

The analysis of the introduced noise that follows the signal envelope works mainly on evaluation of spectral statistics. It is assumed that the inserted noise shows a flat spectral characteristic and forms a 'noise-floor' in the spectral domain. Because of its multiplicative conjunction with the speech, it is only present during speech activity and so the evaluation has to be concentrated on active speech parts.

The scheme in Figure 19 shows the basic components of this approach.

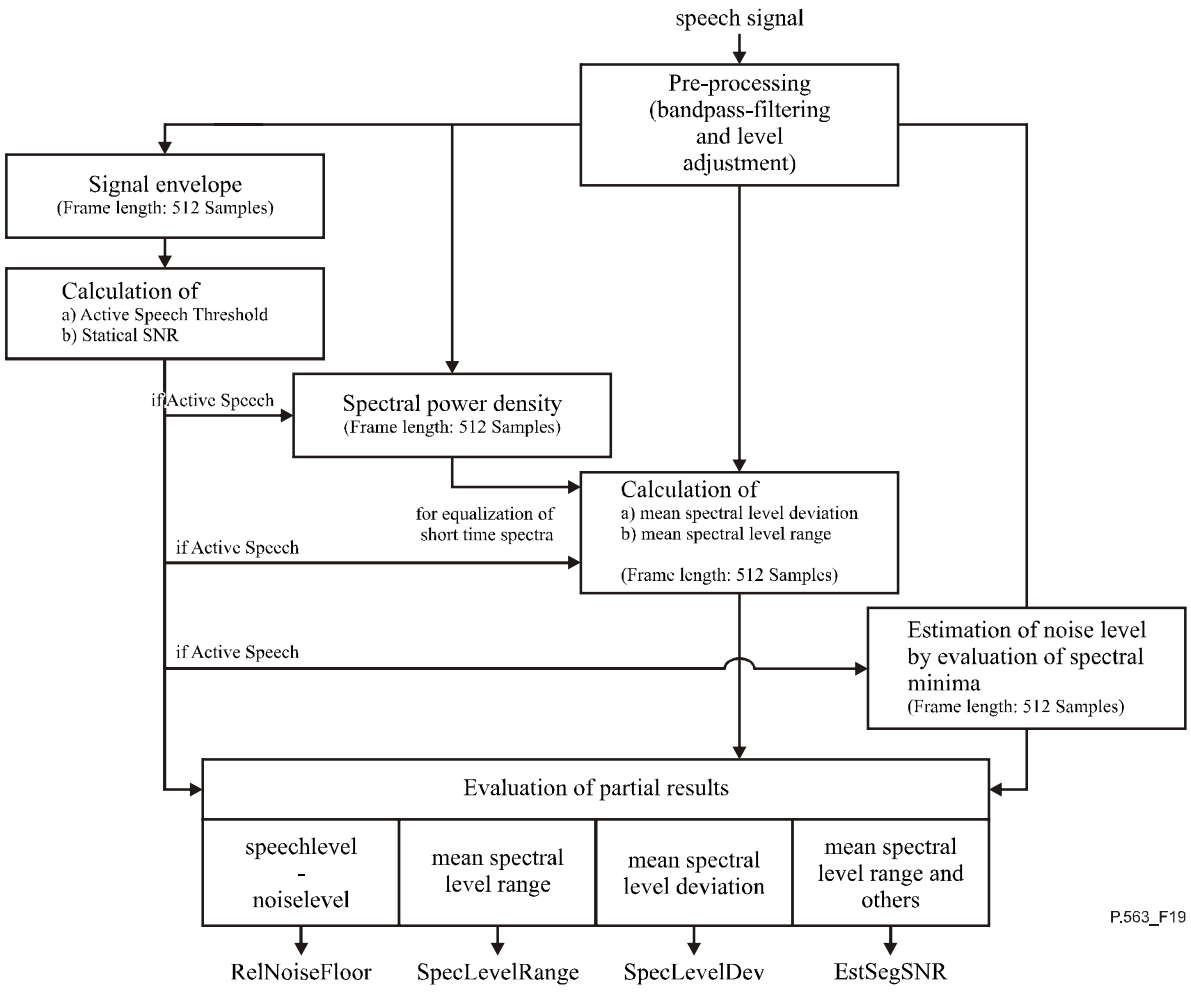


Figure 19/P.563 – Calculation of non-static noise parameters

Before the signal is analysed, a weak 3.1 kHz telephony bandpass is applied to the complete signal.

9.3.2.1 Calculation of the signal envelope and a modified speech activity threshold

Based on the filtered signal, a signal envelope is calculated. Therefore, the signal is divided into frames of length M = 512 samples (64 ms at 8 kHz) with an overlap of 75%. For each frame n the signal power E_n is calculated after removing of the DC-offset for this frame.

$$E_n = \frac{1}{M}\left[\sum_{m=nK}^{(n+1)M} x(m)^2 - \frac{1}{M}\left(\sum_{m=nK}^{(n+1)M} x(m)\right)^2\right] \quad | \; n = 1...N, \text{where } N \text{ is the number of frames}$$

The envelope will be transformed in the level domain by normalizing with the squared maximal value of the amplitude range *(MaxVal)*.

$$L_n = 10\log(E_n / MaxVal^2) \text{ dB} \quad | \; n = 1 \ldots N, \text{ where } N \text{ is the number of frames}$$

The vector L_n contains the normalized level profile of the signal. By calculating the distribution of this envelope vector the upper and lower bounds can be derived. As the upper bound the 20% percentile value $L_{20\%}$ is defined; for the lower bound the 80% percentile value $L_{80\%}$ is used. It is estimated that the $L_{80\%}$ value mainly describes the noise floor in speech pauses; the $L_{20\%}$ value is a threshold where speech is definitely active. The difference between both bounds describes a static ratio between speech and noise floor in speech pauses:

$$SNR_{Stat} = (L_{20\%} - L_{80\%}) \text{ [dB]}$$

The 20% percentile value $L_{20\%}$ is also used for defining a threshold to declare the corresponding frame as containing active speech:

$$L_{SpeechThresh} = (L_{20\%} - 4.0) \text{ [dB]}$$

Using this threshold the speech level L_{Speech} is calculated easily by using all frames which have been defined as active (above $L_{SpeechThresh}$).

$$L_{Speech} = 10\log\left[\frac{1}{N_{Speech}}\sum_{n-1}^{N} E_n\right] \;\Bigg|\; \begin{array}{l} n = 1...N (\text{number of frames}) \\ \text{if } L_n \geq L_{SpeechThresh} \end{array}$$

9.3.2.2 Calculation of the overall spectral density

The overall spectral density of the signal is calculated by averaging the estimated short-term power spectra (periodograms). For the calculation only frames with active speech according to the threshold $L_{SpeechThresh}$ will be considered.

The frame-length is 512 Samples (64 ms at 8 kHz sampling frequency) with an overlap of 75%.

The overall spectral density $\Phi_{XX}(m)$ for all spectral tabs m is calculated by averaging the estimated spectral short time density $\Phi_{XX}(n, m)$ over all frames n. The spectral short time density itself is derived from result F_X of the Fourier-transformation of the active signal-frame after weighting by using a Hann-window:

$$\Phi_{XX}(n,m) = \frac{C_{Hann}}{M}\left|F_X\left(l, e^{j\Omega_m}\right)\right|^2 \;\Bigg|\; \begin{array}{l} \text{with } M = \text{length of frame} \\ C_{Hann} = \dfrac{1}{\sum_{m=1}^{M} w_{Hann}(m)^2} \end{array}$$

and

$$\Phi_{XX}(m) = \frac{1}{N}\sum_{n=1}^{N}\Phi_{XX}(n,m) \;\Bigg|\; \begin{array}{l} n = 1...N(\text{number of frames}) \\ \text{if } L_n \geq L_{SpeechThresh} \end{array}$$

9.3.2.3 Calculation of the spectral level range and deviation (*SpcLevelRange*, *SpcLevelDev*)

The mean spectral level deviation σ_{Spec} and the mean level range δ_{Spec} are partial results and are used for description of the amount of the multiplicative noise to be detected.

Both partial results will be directly calculated from short time spectral densities. Each short time spectral density ('per-frame') is weighted by the overall spectral density to equalize the spectra. The level in dB is the basis for further calculation:

$$X(n,m) = 10\log\left(\frac{\Phi_{XX}(n,m)}{\Phi_{XX}(m)}\right)dB$$

The mean spectral level deviation is the average of all 'per-frame' standard deviations of all spectral levels in the tabs in each frame above $L_{SpeechThresh}$. Note that only the spectral tabs from M/8 to M/4, which cover the range from f = 1000 to 2000 Hz are used here:

$$\sigma_{spec}(n) = \frac{1}{M}\left[\sum_{m=M/8}^{M/4} X(n,m)^2 - \frac{1}{M}\left[\sum_{m=M/8}^{M/4} X(n,m)\right]^2\right] \;\Bigg|\; m = 1...M\,(\text{number of tabs})$$

and

$$\overline{\sigma}_{spec} = \frac{1}{N_{Speech}}\sum_{n<1}^{N}\sigma_{spec}(n) \;\Bigg|\; \begin{array}{l} n = 1...N(\text{number of frames}) \\ \text{if } L_n \geq L_{SpeechThresh} \end{array}$$

The mean level range is calculated also as an average of the individual 'per-frame' level spread. For calculation of the level spread $\delta_{Spec}(n)$ in the frame n the level-distribution for all frequency bins will be calculated. The level spread $\delta_{Spec}(n)$ is the difference between a level close to the minimum level (the 80% percentile level $P_{80\%}$) and a level close to the maximum level (the 15% percentile level $P_{15\%}$ is used). Here no restriction on the frequency range is made.

$$\overline{\delta}_{spec} = \frac{1}{N_{Speech}}\sum_{n=1}^{N}\delta_{spec}(n) = \frac{1}{N_{Speech}}\sum_{n=1}^{N}\left[P_{15\%}(n) - P_{80\%}(n)\right] \;\Bigg|\; \begin{array}{l} n = 1...N(\text{number of frames}) \\ \text{if } L_n \geq L_{SpeechThresh} \end{array}$$

The resulting mean level range δ_{Spec} is the main value for estimating the segmental SNR, which describes the amount of the multiplicative noise related to the corresponding speech level.

9.3.2.4 Calculation of the relative noise floor (*RelNoiseFloor*)

The noise level calculation estimates the varying noise floor during speech activity. It is used for confirmation of the estimated segmental SNR, which is calculated mainly by the mean spectral level distance δ_{Spec}.

This calculation step is also based on an analysis of the short time spectra. At first, for each frame, the spectral power density, Φ, will be calculated. In the next stage for blocks of 10 frames, one after the other, the spectral minima are detected and stored. Because of the 0.75 overlap used, this block is effectively 1664 samples (= 208 ms at 8 kHz sampling frequency) long:

$$\Phi_{X\min}(r,m) = \min(\Phi_{XX}(n=10r-9,m),...,\Phi_{XX}(n=10r,m)) \;\Bigg|\; \begin{array}{l} r = 1...N/10 \\ m = 1...M \end{array}$$

For practical reasons, the accepted minima are limited at 10^{-8}, which corresponds to the noise threshold of –80 dBov.

$$\Phi_{X\min}(r,m) = \max(10^{-8}, \Phi_{X\min}(r,m)) \left| \begin{array}{l} r = 1...N/10 \\ m = 1...M \end{array} \right.$$

The resulting vector contains the minimal values for each spectral tab in the current block of 10 frames and describes an assumed noise floor in this time range. The noise level will be calculated in a restricted frequency range (f = 1000 to 2000 Hz) and is set valid for the 10 corresponding frames.

$$E_{Noise}(n) = \sum_{m=M/8}^{M/4} \Phi_{X\min}(r = \text{int}(n/10), m) \left| \begin{array}{l} n = 1...N(\text{number of frames}) \\ m = 1...M(\text{number of tabs}) \end{array} \right.$$

At the last step, the power of the estimated noise floor will be cumulated for all active speech frames:

$$L_{Noise} = 10\ \log\left[\frac{1}{\mathrm{N}_{Speech}} \sum_{n=1}^{N} E_{Noise}(n) \right] \left| \begin{array}{l} n = 1...N(\text{number of frames}) \\ \text{if } L_n \geq L_{SpeechThresh} \end{array} \right.$$

The difference between the speech level L_{Speech} and the noise level in active frames is used as *RelNoiseFloor* as a parameter for the speech quality model as well as an intermediate result for confirmation of the estimated SNR to avoid incorrect predictions.

9.3.2.5 Weighting of intermediate results (*SegSNR*)

In a post-processing step all partial results will be used for the detection of multiplicative noise as well as inputs for the speech quality estimation. The main parameter is the estimated segmental SNR, which is used as the detection flag of quality degrading multiplicative noise.

$$SNR_{Seg} = 2.7\left(\overline{\delta}_{spec} - 10.5\right)$$

The segmental SNR will be only accepted if it is below 25 dB and if several partial conditions are fulfilled. Otherwise, it is set to 40 dB, which means there is no multiplicative noise detected.

$$SNR_{Seg} = \begin{cases} 40, & \text{if } SNR_{Seg} > 25.0 \\ 40, & \text{if } \overline{\sigma}_{spec} > 8.1 \\ 40, & \text{if } \left| 3\left(L_{Speech} - L_{Noise}\right) - SNR_{Seg} \right| > 86.0 \\ 40, & \text{if } SNR_{Stat} > 38.0 \end{cases}$$

A quality degrading multiplicative noise is accepted as detected if the resulting SNR_{Seg} is below 25 dB and the LPC kurtosis from the speech statistics module is between 2.0 and 9.0.

9.4 Description of the 'Mutes/Interruptions' functional block components

9.4.1 Mute length estimation

The duration of muting is calculated as shown in Figure 20.

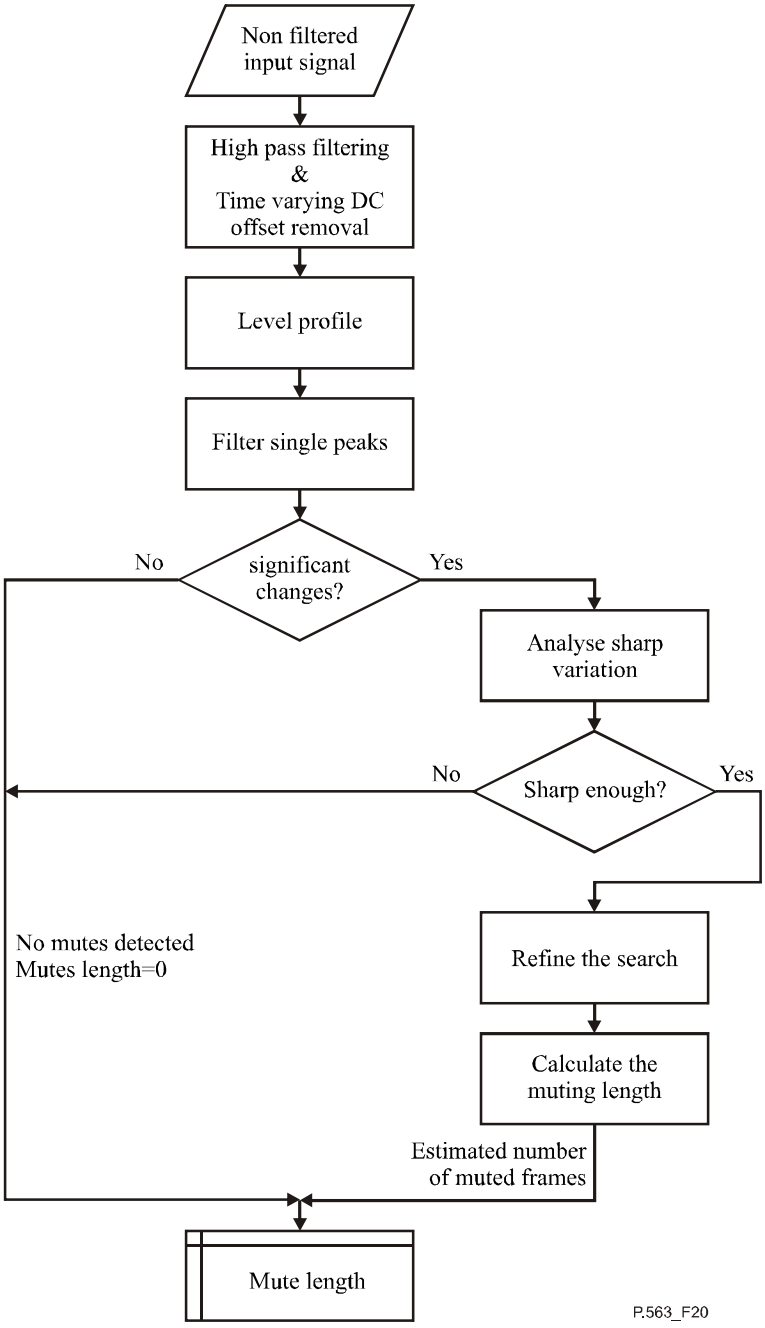


Figure 20/P.563 – Mute length calculation

A part of the signal is considered as muted if a drop in the signal can be located. A drop in the signal is characterized by a deep gap with sharp edges, defined by significant level changes. The unfiltered input signal is used as a different filter and is applied within this algorithm.

9.4.1.1 Filtering

The first step is to remove secondary distortions introduced by mutes. These resulting effects appear in the low frequency band.

The signal is passed through a 500 Hz 4th order high pass Butterworth filter, and a temporal DC offset removal filter using a Notch design.

9.4.1.2 Level profile generation

The level profile (Lvl) is generated by calculating the level in dB of the filtered signal using 4 ms frames. The level array is cleaned up by removing single peaks, characterized by three consecutive frames where the 2nd frame level is at least 30 dB above the 1st and the 3rd frame. A single peak is defined when the following two conditions are met:

$$\begin{cases} Lvl[i] - Lvl[i-1] > 30 \text{ dB} \\ Lvl[i] - Lvl[i+1] > 30 \text{ dB} \end{cases}$$

9.4.1.3 Level profile analysis

The level profile is parsed, searching for sharp drops and rises of the signal. The decision is made as follows:

A drop is marked as a potential mute beginning if $Lvl[i-2] - Lvl[i] > LvlThres$. A rise is marked as a potential mute end if $Lvl[i+2] - Lvl[i-2] > LvlThres$.

Initially, the analysis is performed with *LvlThres* set to 30 dB. If potential mute starts and stops are found, the analysis is repeated with a reduced threshold, 20 dB.

If both potential mute starts and stops have not been found, the algorithm returns with no mutes.

9.4.1.4 Mute length calculation

The previous step located all gaps at least 20 dB deep in the signal. If two gaps are separated by less that 360 ms (90 frames), they are joined to form one large gap. The mute length is represented by the sum of the width of the gaps. The calculated mute amount is capped at 250 frames (1 sec) over the whole file.

9.4.2 Sharp declines

The main assumption for this signal descriptor is that a natural voice signal can never decline abruptly. Voice signals always have certain fadings at the beginning and the end of every natural utterance.

The detector works in several steps:

The signal power of two frames each with a duration of 10 ms are compared to each other. If the power ratio of a frame at time t and a frame at time t + 50 ms becomes larger than 10.5, two further indicators are calculated to avoid having detected the ending of a legal vowel instead of a 'sharp decline'. The two indicators are derived from a periodicity measure based on two cross-correlations with the following features:

	Frequency range	**Frame size**	**Location of calculation (t = current time)**	**Correlation threshold**
1. Periodicity	250-1000 Hz	16 ms	t-30 ms into the past	0.70
2. Periodicity	250-1500 Hz	16 ms	t-60 ms into the past	0.65

If both periodicities in the past blocks are smaller than a certain correlation threshold (see table), or if the power ratio (as mentioned above) is larger than 24 and if the signal contains enough energy, most likely a 'sharp decline' has been detected.

The *SharpDeclines* parameter is given by the number of sharp declines multiplied by 1000 and divided by the length of the total number of processed samples.

9.4.3 Unnatural silence

This module detects unnaturally silenced speech signals.

Initially, the signal's envelope (RMS) is determined using 320 ms frames with a shift of 40 ms. The algorithm iterates through the envelopes and sets a flag for each envelope to either low or high to indicate that the envelope is low (≤ 100) or high (≥ 300) using a hysteresis approach. The number of transitions from low to high and vice versa within an interval of 2.6 seconds is counted. If there are more than 4 transitions in the current interval, which is more than it is assumed to occur in natural speech, unnatural silence is detected.

The following parameter is generated:

- *UnnaturalSilenceTotEnergy*

  Total energy of 320 ms frames classified as unnatural silence.

9.4.4 Signal interruption detection

Signal interruption can occur in two variants i.e., as temporal speech clipping or speech interruption. Both lead to a loss of signal information. Furthermore, the algorithm must be able to distinguish between the normal word ends and the abnormal signal interruptions.

The algorithm implemented in this Recommendation works on two speech frames, each 32 ms long. It can detect the interruptions from a couple of samples up to about 80 ms. The result of the detection algorithm is the interruption position, length, estimated power and estimated pitch frequency at place of the interruption.

For this module, it is also assumed that the input signal is normalized for level and any DC offset has been removed. The following tasks are performed during the interruption detection procedure and the features mentioned are shown in Figure 21:

- Evaluate pitch frequency within a single frame of 32 ms duration.
- Based on pitch value a processing unit of 32 ms is divided into fundamental frames.
- A maximum signal value and its position is evaluated in each fundamental window (MAX(i)).
- Based on maximum values of each window a preselection of subframes is performed.
- The mean level of potential interruption frames as an additional threshold for decision is calculated.
- Interruption duration measurement.

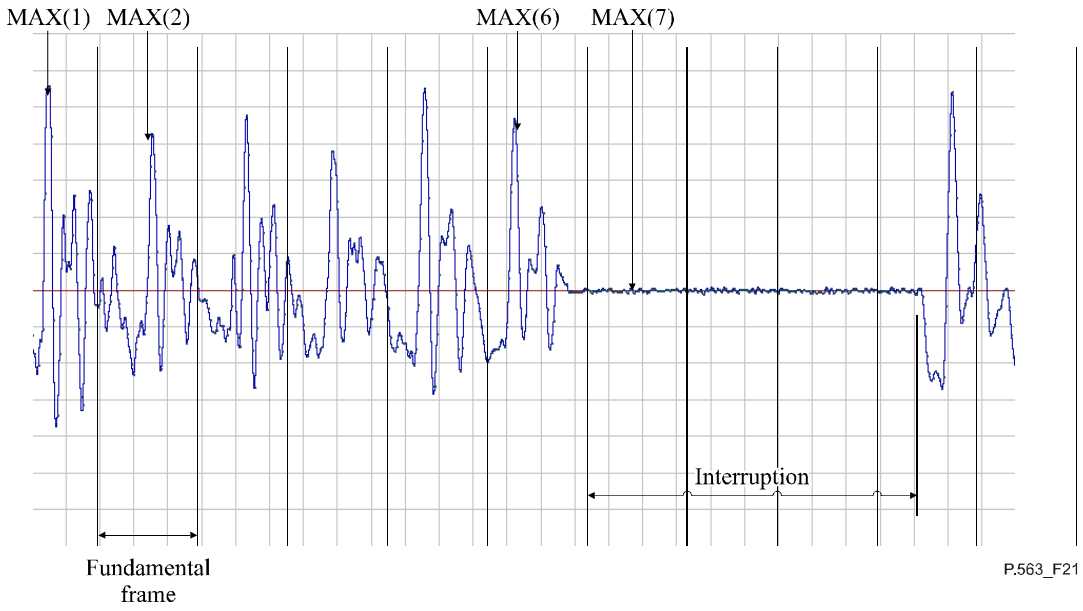


Figure 21/P.563 – Fundamental frames when detecting interruptions

In order to be able to detect signal interruptions of more than 32 ms (one processing buffer), an access to more buffers has to be provided. For this reason there are three processing buffers allocated as working space for interruption detection.

9.5 Description of the Speech Quality Model

The basic block-scheme of P.563 in Figure 22 shows the speech quality model is composed of three main blocks:

– Decision on a distortion class.

– Speech quality evaluation for the corresponding distortion class.

– Overall calculation of speech quality.

9.5.1 Distortion class decision

The assigned distortion class calculated in the key parameters block is used for the adjustment of the speech quality model. In the case of several distortions occurring in the signal, a prioritization is applied on the distortion classes as shown in Figure 22. As shown in this figure, if the signal is undistorted, it will be assessed using the same rule as unnatural voice signals.

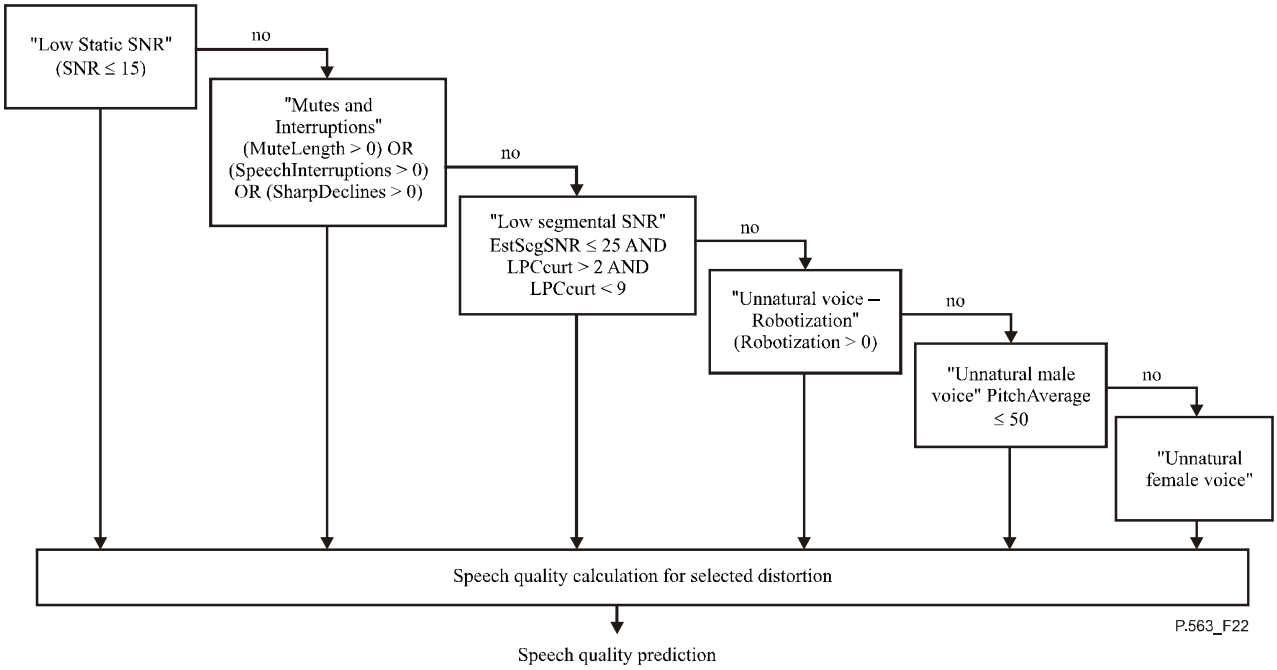


Figure 22/P.563 – Distortion class decision

9.5.2 Speech quality related to the class distortion

Each class distortion uses a linear combination of parameters to generate the intermediate speech quality. Table 5 details the necessary parameters.

Table 5/P.563 – Parameters used for the distortion speech

Unnatural female voice	Unnatural male voice	Unnatural voice/Robotization
SNR (9.3.1.1)	*SpectralClarity* (9.3.1.5)	*CepCurt* (9.2.1.2)
SpectralClarity (9.3.1.5)	*SNR* (9.3.1.1)	*UnnaturalBeeps* (9.2.3.4)
SpeechSectionsLevelVar (9.1.4)	*FinalVtpAverage* (9.2.2.2.2)	*PitchCrossPower* (9.2.3.1)
VTPMaxTubeSection (9.2.2.2.1)	*BasicVoiceQuality* (9.2.4.2.16)	*VTPPeakTracker* (9.2.2.2.5)
LPCskewAbs (9.2.1.1)	*SegSNR* (9.3.2.5)	*LPCcurt* (9.2.1.1)
HiFreqVar (9.3.1.4)	*PitchAverage* (9.1.6)	*ARTAverage* (9.2.2.2.3)
BasicVoiceQuality (9.2.4.2.16)	*EstBGNoise* (9.3.1.6)	*SpeechSectionsLevelVar* (9.1.4)
PitchAverage (9.1.6)	*UnnaturalSilenceTotEnergy* (9.4.3)	*VTPMaxTubeSection* (9.2.2.2.1)
ARTAverage (9.2.2.2.3)	*LocalBGNoiseMean* (9.3.1.2)	*LPCskew* (9.2.1.1)
CepADev (9.2.1.2)	*LPCskewAbs* (9.2.1.1)	*SpectralClarity* (9.3.1.5)
EstBGNoise (9.3.1.6)	*VtpVadOverlap* (9.2.2.2.6)	*HiFreqVar* (9.3.1.4)
UnnaturalSilenceTotEnergy (9.4.3)	*SpeechSectionsLevelVar* (9.1.4)	*UnnaturalSilenceTotEnergy* (9.4.3)
Low segmental SNR	**Mutes and Interruptions**	**Low static SNR**

9.5.3 Overall speech quality

The final speech quality is calculated by combining the intermediate quality result with some additional signal features, as shown in Figure 23.

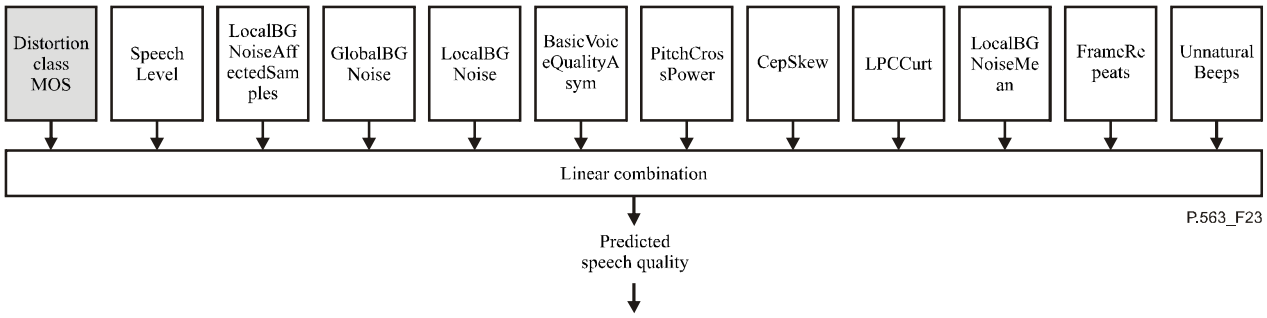


Figure 23/P.563 – Overall speech quality prediction

Annex A

Source code for reference implementation and conformance tests

A.1 List of files provided for the ANSI-C reference implementation

The ANSI-C reference implementation of P.563 is contained in the following text files which are provided in the `source` subdirectory of the CD-ROM distribution:

C files

- back_noise.c *Local and Global Background Noise*
- beeprob.c *Evaluation of UnnaturalBeeps Robotization UnnaturalSilences SharpDeclines and FrameRepeats*
- dsp.c *Signal processing functions*
- Enhance.c *Speech enhancement modules*
- EvalQual.c *Basic Quality evaluation*
- hosm.c *Speech statistics functions*
- inter_detect.c *Signal interruption functions*
- lpc.c *LPC calculation functions*
- LpcAnalysis.c *Linear Prediction modules for Speech enhancement*
- mapping.c *Perceptual mapping*
- module1.c *Pitch, vocal tract, noise and mutes analysis functions*
- module2.c *Unnatural speech, noise and mutes analysis functions*
- module3.c *Speech statistics interruptions and segmental SNR functions*
- mytools.c *Basic Arithmetic functions*
- p563.c *Main program*
- pitch.c *Pitch extraction functions*
- Quant.c *Speech enhancement modules*
- SignalsPercept.c *Basic Voice quality functions*
- SpeechLib.c *Basic filter functions*
- Statistics.c *Basic Statistic functions*
- tools.c *Arithmetic functions*
- vector_lib.c *Basic vector arithmetic functions*

Header files

- back_noise.h *Header file for back_noise.c*
- beeprob.h *Header file for beeprob.c*
- defines.h *P.563 model definitions*
- dsp.h *Header file for dsp.c*
- Enhance.h *Header file for Enhance.c*
- EvalQual.h *Header file for EvalQual.c*
- generic_typedefs.h
- hosm.h *Header file for hosm.c*

- interr_detect.h *Header file for inter_detect.c*
- lpc.h *Header file for lpc.c*
- LpcAnalysis.h *Header file for LpcAnalysis.c*
- mapping.h *Header file for mapping.c*
- module1.h *Header file for module1.c*
- module2.h *Header file for module2.c*
- module3.h *Header file for module3.c*
- mytools.h *Header file for mytools.c*
- pitch.h *Header file for pitch.c*
- Quant.h *Header file for Quant.c*
- QuantTab.h *Constants for Quant.c*
- resource.h
- SignalsPercept.h *Header file for SignalsPercept.c*
- SpeechLib.h *Header file for SpeechLib.c*
- Statistics.h *Header file for Statistics.c*
- tools.h *Header file for tools.c*
- vector_lib.h *Header file for vector_lib.c*

The ANSI-C reference implementation is provided in separate files and forms an integral part of this Recommendation. The ANSI-C reference implementation shall take precedence in case of conflicts between the high level description as given in this Recommendation and the ANSI-C reference implementation.

A.2 List of files provided for conformance validation

The conformance validation process described below makes reference to the following files, which are provided in the `conform` subdirectory of the CD-ROM distribution:

- supp23.txt *File pairs and P.563 scores for Supplement 23 exp1 and exp3*.
- process.bat *Sample batch script to assist with preparing material*.

EXP1-A	EXP1-D	EXP1-O	EXP3-A	EXP3-C	EXP3-D	EXP3-O
S23e1a\ae1f5901.out	S23e1d\de1f5901.out	S23e1o\oe1f5901.out	S23e3a\ae3f6501.out	S23e3c\ce3f6501.out	S23e3d\de3f6501.out	S23e3o\oe3f6501.out
S23e1a\ae1f8501.out	S23e1d\de1f8501.out	S23e1o\oe1f8501.out	S23e3a\ae3f9701.out	S23e3c\ce3f9701.out	S23e3d\de3f9701.out	S23e3o\oe3f9701.out
S23e1a\ae1m0101.out	S23e1d\de1m0101.out	S23e1o\oe1m0101.out	S23e3a\ae3m0101.out	S23e3c\ce3m0101.out	S23e3d\de3m0101.out	S23e3o\oe3m0101.out
S23e1a\ae1m2d01.out	S23e1d\de1m2d01.out	S23e1o\oe1m2d01.out	S23e3a\ae3m3301.out	S23e3c\ce3m3301.out	S23e3d\de3m3301.out	S23e3o\oe3m3301.out
S23e1a\ae1f5a02.out	S23e1d\de1f5a02.out	S23e1o\oe1f5a02.out	S23e3a\ae3f6602.out	S23e3c\ce3f6602.out	S23e3d\de3f6602.out	S23e3o\oe3f6602.out
S23e1a\ae1f8602.out	S23e1d\de1f8602.out	S23e1o\oe1f8602.out	S23e3a\ae3f9802.out	S23e3c\ce3f9802.out	S23e3d\de3f9802.out	S23e3o\oe3f9802.out
S23e1a\ae1m0202.out	S23e1d\de1m0202.out	S23e1o\oe1m0202.out	S23e3a\ae3m0202.out	S23e3c\ce3m0202.out	S23e3d\de3m0202.out	S23e3o\oe3m0202.out
S23e1a\ae1m2e02.out	S23e1d\de1m2e02.out	S23e1o\oe1m2e02.out	S23e3a\ae3m3402.out	S23e3c\ce3m3402.out	S23e3d\de3m3402.out	S23e3o\oe3m3402.out
S23e1a\ae1f5b03.out	S23e1d\de1f5b03.out	S23e1o\oe1f5b03.out	S23e3a\ae3f6703.out	S23e3c\ce3f6703.out	S23e3d\de3f6703.out	S23e3o\oe3f6703.out
S23e1a\ae1f8703.out	S23e1d\de1f8703.out	S23e1o\oe1f8703.out	S23e3a\ae3f9903.out	S23e3c\ce3f9903.out	S23e3d\de3f9903.out	S23e3o\oe3f9903.out
S23e1a\ae1m0303.out	S23e1d\de1m0303.out	S23e1o\oe1m0303.out	S23e3a\ae3m0303.out	S23e3c\ce3m0303.out	S23e3d\de3m0303.out	S23e3o\oe3m0303.out
S23e1a\ae1m2f03.out	S23e1d\de1m2f03.out	S23e1o\oe1m2f03.out	S23e3a\ae3m3503.out	S23e3c\ce3m3503.out	S23e3d\de3m3503.out	S23e3o\oe3m3503.out
S23e1a\ae1f5c04.out	S23e1d\de1f5c04.out	S23e1o\oe1f5c04.out	S23e3a\ae3f6804.out	S23e3c\ce3f6804.out	S23e3d\de3f6804.out	S23e3o\oe3f6804.out
S23e1a\ae1f8804.out	S23e1d\de1f8804.out	S23e1o\oe1f8804.out	S23e3a\ae3f9a04.out	S23e3c\ce3f9a04.out	S23e3d\de3f9a04.out	S23e3o\oe3f9a04.out
S23e1a\ae1m0404.out	S23e1d\de1m0404.out	S23e1o\oe1m0404.out	S23e3a\ae3m0404.out	S23e3c\ce3m0404.out	S23e3d\de3m0404.out	S23e3o\oe3m0404.out
S23e1a\ae1m3004.out	S23e1d\de1m3004.out	S23e1o\oe1m3004.out	S23e3a\ae3m3604.out	S23e3c\ce3m3604.out	S23e3d\de3m3604.out	S23e3o\oe3m3604.out
S23e1a\ae1f5d05.out	S23e1d\de1f5d05.out	S23e1o\oe1f5d05.out	S23e3a\ae3f6905.out	S23e3c\ce3f6905.out	S23e3d\de3f6905.out	S23e3o\oe3f6905.out
S23e1a\ae1f8905.out	S23e1d\de1f8905.out	S23e1o\oe1f8905.out	S23e3a\ae3f9b05.out	S23e3c\ce3f9b05.out	S23e3d\de3f9b05.out	S23e3o\oe3f9b05.out
S23e1a\ae1m0505.out	S23e1d\de1m0505.out	S23e1o\oe1m0505.out	S23e3a\ae3m0505.out	S23e3c\ce3m0505.out	S23e3d\de3m0505.out	S23e3o\oe3m0505.out
S23e1a\ae1m3105.out	S23e1d\de1m3105.out	S23e1o\oe1m3105.out	S23e3a\ae3m3705.out	S23e3c\ce3m3705.out	S23e3d\de3m3705.out	S23e3o\oe3m3705.out
S23e1a\ae1f5e06.out	S23e1d\de1f5e06.out	S23e1o\oe1f5e06.out	S23e3a\ae3f6a06.out	S23e3c\ce3f6a06.out	S23e3d\de3f6a06.out	S23e3o\oe3f6a06.out
S23e1a\ae1f8a06.out	S23e1d\de1f8a06.out	S23e1o\oe1f8a06.out	S23e3a\ae3f9c06.out	S23e3c\ce3f9c06.out	S23e3d\de3f9c06.out	S23e3o\oe3f9c06.out
S23e1a\ae1m0606.out	S23e1d\de1m0606.out	S23e1o\oe1m0606.out	S23e3a\ae3m0606.out	S23e3c\ce3m0606.out	S23e3d\de3m0606.out	S23e3o\oe3m0606.out
S23e1a\ae1m3206.out	S23e1d\de1m3206.out	S23e1o\oe1m3206.out	S23e3a\ae3m3806.out	S23e3c\ce3m3806.out	S23e3d\de3m3806.out	S23e3o\oe3m3806.out
S23e1a\ae1f5f07.out	S23e1d\de1f5f07.out	S23e1o\oe1f5f07.out	S23e3a\ae3f6b07.out	S23e3c\ce3f6b07.out	S23e3d\de3f6b07.out	S23e3o\oe3f6b07.out
S23e1a\ae1f8b07.out	S23e1d\de1f8b07.out	S23e1o\oe1f8b07.out	S23e3a\ae3f9d07.out	S23e3c\ce3f9d07.out	S23e3d\de3f9d07.out	S23e3o\oe3f9d07.out
S23e1a\ae1m0707.out	S23e1d\de1m0707.out	S23e1o\oe1m0707.out	S23e3a\ae3m0707.out	S23e3c\ce3m0707.out	S23e3d\de3m0707.out	S23e3o\oe3m0707.out
S23e1a\ae1m3307.out	S23e1d\de1m3307.out	S23e1o\oe1m3307.out	S23e3a\ae3m3907.out	S23e3c\ce3m3907.out	S23e3d\de3m3907.out	S23e3o\oe3m3907.out
S23e1a\ae1f6008.out	S23e1d\de1f6008.out	S23e1o\oe1f6008.out	S23e3a\ae3f6c08.out	S23e3c\ce3f6c08.out	S23e3d\de3f6c08.out	S23e3o\oe3f6c08.out
S23e1a\ae1f8c08.out	S23e1d\de1f8c08.out	S23e1o\oe1f8c08.out	S23e3a\ae3f9e08.out	S23e3c\ce3f9e08.out	S23e3d\de3f9e08.out	S23e3o\oe3f9e08.out
S23e1a\ae1m0808.out	S23e1d\de1m0808.out	S23e1o\oe1m0808.out	S23e3a\ae3m0808.out	S23e3c\ce3m0808.out	S23e3d\de3m0808.out	S23e3o\oe3m0808.out
S23e1a\ae1m3408.out	S23e1d\de1m3408.out	S23e1o\oe1m3408.out	S23e3a\ae3m3a08.out	S23e3c\ce3m3a08.out	S23e3d\de3m3a08.out	S23e3o\oe3m3a08.out
S23e1a\ae1f6109.out	S23e1d\de1f6109.out	S23e1o\oe1f6109.out	S23e3a\ae3f6d09.out	S23e3c\ce3f6d09.out	S23e3d\de3f6d09.out	S23e3o\oe3f6d09.out
S23e1a\ae1f8d09.out	S23e1d\de1f8d09.out	S23e1o\oe1f8d09.out	S23e3a\ae3f9f09.out	S23e3c\ce3f9f09.out	S23e3d\de3f9f09.out	S23e3o\oe3f9f09.out
S23e1a\ae1m0909.out	S23e1d\de1m0909.out	S23e1o\oe1m0909.out	S23e3a\ae3m0909.out	S23e3c\ce3m0909.out	S23e3d\de3m0909.out	S23e3o\oe3m0909.out
S23e1a\ae1m3509.out	S23e1d\de1m3509.out	S23e1o\oe1m3509.out	S23e3a\ae3m3b09.out	S23e3c\ce3m3b09.out	S23e3d\de3m3b09.out	S23e3o\oe3m3b09.out

EXP1-A	EXP1-D	EXP1-O	EXP3-A	EXP3-C	EXP3-D	EXP3-O
S23e1a\ae1f6210.out	S23e1d\de1f6210.out	S23e1o\oe1f6210.out	S23e3a\ae3f6e10.out	S23e3c\ce3f6e10.out	S23e3d\de3f6e10.out	S23e3o\oe3f6e10.out
S23e1a\ae1f8e10.out	S23e1d\de1f8e10.out	S23e1o\oe1f8e10.out	S23e3a\ae3fa010.out	S23e3c\ce3fa010.out	S23e3d\de3fa010.out	S23e3o\oe3fa010.out
S23e1a\ae1m0a10.out	S23e1d\de1m0a10.out	S23e1o\oe1m0a10.out	S23e3a\ae3m0a10.out	S23e3c\ce3m0a10.out	S23e3d\de3m0a10.out	S23e3o\oe3m0a10.out
S23e1a\ae1m3610.out	S23e1d\de1m3610.out	S23e1o\oe1m3610.out	S23e3a\ae3m3c10.out	S23e3c\ce3m3c10.out	S23e3d\de3m3c10.out	S23e3o\oe3m3c10.out
S23e1a\ae1f6311.out	S23e1d\de1f6311.out	S23e1o\oe1f6311.out	S23e3a\ae3f6f11.out	S23e3c\ce3f6f11.out	S23e3d\de3f6f11.out	S23e3o\oe3f6f11.out
S23e1a\ae1f8f11.out	S23e1d\de1f8f11.out	S23e1o\oe1f8f11.out	S23e3a\ae3fa111.out	S23e3c\ce3fa111.out	S23e3d\de3fa111.out	S23e3o\oe3fa111.out
S23e1a\ae1m0b11.out	S23e1d\de1m0b11.out	S23e1o\oe1m0b11.out	S23e3a\ae3m0b11.out	S23e3c\ce3m0b11.out	S23e3d\de3m0b11.out	S23e3o\oe3m0b11.out
S23e1a\ae1m3711.out	S23e1d\de1m3711.out	S23e1o\oe1m3711.out	S23e3a\ae3m3d11.out	S23e3c\ce3m3d11.out	S23e3d\de3m3d11.out	S23e3o\oe3m3d11.out
S23e1a\ae1f6412.out	S23e1d\de1f6412.out	S23e1o\oe1f6412.out	S23e3a\ae3f7012.out	S23e3c\ce3f7012.out	S23e3d\de3f7012.out	S23e3o\oe3f7012.out
S23e1a\ae1f9012.out	S23e1d\de1f9012.out	S23e1o\oe1f9012.out	S23e3a\ae3fa212.out	S23e3c\ce3fa212.out	S23e3d\de3fa212.out	S23e3o\oe3fa212.out
S23e1a\ae1m0c12.out	S23e1d\de1m0c12.out	S23e1o\oe1m0c12.out	S23e3a\ae3m0c12.out	S23e3c\ce3m0c12.out	S23e3d\de3m0c12.out	S23e3o\oe3m0c12.out
S23e1a\ae1m3812.out	S23e1d\de1m3812.out	S23e1o\oe1m3812.out	S23e3a\ae3m3e12.out	S23e3c\ce3m3e12.out	S23e3d\de3m3e12.out	S23e3o\oe3m3e12.out
S23e1a\ae1f6513.out	S23e1d\de1f6513.out	S23e1o\oe1f6513.out	S23e3a\ae3f7113.out	S23e3c\ce3f7113.out	S23e3d\de3f7113.out	S23e3o\oe3f7113.out
S23e1a\ae1f9113.out	S23e1d\de1f9113.out	S23e1o\oe1f9113.out	S23e3a\ae3fa313.out	S23e3c\ce3fa313.out	S23e3d\de3fa313.out	S23e3o\oe3fa313.out
S23e1a\ae1m0d13.out	S23e1d\de1m0d13.out	S23e1o\oe1m0d13.out	S23e3a\ae3m0d13.out	S23e3c\ce3m0d13.out	S23e3d\de3m0d13.out	S23e3o\oe3m0d13.out
S23e1a\ae1m3913.out	S23e1d\de1m3913.out	S23e1o\oe1m3913.out	S23e3a\ae3m3f13.out	S23e3c\ce3m3f13.out	S23e3d\de3m3f13.out	S23e3o\oe3m3f13.out
S23e1a\ae1f6614.out	S23e1d\de1f6614.out	S23e1o\oe1f6614.out	S23e3a\ae3f7214.out	S23e3c\ce3f7214.out	S23e3d\de3f7214.out	S23e3o\oe3f7214.out
S23e1a\ae1f9214.out	S23e1d\de1f9214.out	S23e1o\oe1f9214.out	S23e3a\ae3fa414.out	S23e3c\ce3fa414.out	S23e3d\de3fa414.out	S23e3o\oe3fa414.out
S23e1a\ae1m0e14.out	S23e1d\de1m0e14.out	S23e1o\oe1m0e14.out	S23e3a\ae3m0e14.out	S23e3c\ce3m0e14.out	S23e3d\de3m0e14.out	S23e3o\oe3m0e14.out
S23e1a\ae1m3a14.out	S23e1d\de1m3a14.out	S23e1o\oe1m3a14.out	S23e3a\ae3m4014.out	S23e3c\ce3m4014.out	S23e3d\de3m4014.out	S23e3o\oe3m4014.out
S23e1a\ae1f6715.out	S23e1d\de1f6715.out	S23e1o\oe1f6715.out	S23e3a\ae3f7315.out	S23e3c\ce3f7315.out	S23e3d\de3f7315.out	S23e3o\oe3f7315.out
S23e1a\ae1f9315.out	S23e1d\de1f9315.out	S23e1o\oe1f9315.out	S23e3a\ae3fa515.out	S23e3c\ce3fa515.out	S23e3d\de3fa515.out	S23e3o\oe3fa515.out
S23e1a\ae1m0f15.out	S23e1d\de1m0f15.out	S23e1o\oe1m0f15.out	S23e3a\ae3m0f15.out	S23e3c\ce3m0f15.out	S23e3d\de3m0f15.out	S23e3o\oe3m0f15.out
S23e1a\ae1m3b15.out	S23e1d\de1m3b15.out	S23e1o\oe1m3b15.out	S23e3a\ae3m4115.out	S23e3c\ce3m4115.out	S23e3d\de3m4115.out	S23e3o\oe3m4115.out
S23e1a\ae1f6816.out	S23e1d\de1f6816.out	S23e1o\oe1f6816.out	S23e3a\ae3f7416.out	S23e3c\ce3f7416.out	S23e3d\de3f7416.out	S23e3o\oe3f7416.out
S23e1a\ae1f9416.out	S23e1d\de1f9416.out	S23e1o\oe1f9416.out	S23e3a\ae3fa616.out	S23e3c\ce3fa616.out	S23e3d\de3fa616.out	S23e3o\oe3fa616.out
S23e1a\ae1m1016.out	S23e1d\de1m1016.out	S23e1o\oe1m1016.out	S23e3a\ae3m1016.out	S23e3c\ce3m1016.out	S23e3d\de3m1016.out	S23e3o\oe3m1016.out
S23e1a\ae1m3c16.out	S23e1d\de1m3c16.out	S23e1o\oe1m3c16.out	S23e3a\ae3m4216.out	S23e3c\ce3m4216.out	S23e3d\de3m4216.out	S23e3o\oe3m4216.out
S23e1a\ae1f6917.out	S23e1d\de1f6917.out	S23e1o\oe1f6917.out	S23e3a\ae3f7517.out	S23e3c\ce3f7517.out	S23e3d\de3f7517.out	S23e3o\oe3f7517.out
S23e1a\ae1f9517.out	S23e1d\de1f9517.out	S23e1o\oe1f9517.out	S23e3a\ae3fa717.out	S23e3c\ce3fa717.out	S23e3d\de3fa717.out	S23e3o\oe3fa717.out
S23e1a\ae1m1117.out	S23e1d\de1m1117.out	S23e1o\oe1m1117.out	S23e3a\ae3m1117.out	S23e3c\ce3m1117.out	S23e3d\de3m1117.out	S23e3o\oe3m1117.out
S23e1a\ae1m3d17.out	S23e1d\de1m3d17.out	S23e1o\oe1m3d17.out	S23e3a\ae3m4317.out	S23e3c\ce3m4317.out	S23e3d\de3m4317.out	S23e3o\oe3m4317.out
S23e1a\ae1f6a18.out	S23e1d\de1f6a18.out	S23e1o\oe1f6a18.out	S23e3a\ae3f7618.out	S23e3c\ce3f7618.out	S23e3d\de3f7618.out	S23e3o\oe3f7618.out
S23e1a\ae1f9618.out	S23e1d\de1f9618.out	S23e1o\oe1f9618.out	S23e3a\ae3fa818.out	S23e3c\ce3fa818.out	S23e3d\de3fa818.out	S23e3o\oe3fa818.out
S23e1a\ae1m1218.out	S23e1d\de1m1218.out	S23e1o\oe1m1218.out	S23e3a\ae3m1218.out	S23e3c\ce3m1218.out	S23e3d\de3m1218.out	S23e3o\oe3m1218.out
S23e1a\ae1m3e18.out	S23e1d\de1m3e18.out	S23e1o\oe1m3e18.out	S23e3a\ae3m4418.out	S23e3c\ce3m4418.out	S23e3d\de3m4418.out	S23e3o\oe3m4418.out

EXP1-A	EXP1-D	EXP1-O	EXP3-A	EXP3-C	EXP3-D	EXP3-O
S23e1a\ae1f6b19.out	S23e1d\de1f6b19.out	S23e1o\oe1f6b19.out	S23e3a\ae3f7719.out	S23e3c\ce3f7719.out	S23e3d\de3f7719.out	S23e3o\oe3f7719.out
S23e1a\ae1f9719.out	S23e1d\de1f9719.out	S23e1o\oe1f9719.out	S23e3a\ae3fa919.out	S23e3c\ce3fa919.out	S23e3d\de3fa919.out	S23e3o\oe3fa919.out
S23e1a\ae1m1319.out	S23e1d\de1m1319.out	S23e1o\oe1m1319.out	S23e3a\ae3m1319.out	S23e3c\ce3m1319.out	S23e3d\de3m1319.out	S23e3o\oe3m1319.out
S23e1a\ae1m3f19.out	S23e1d\de1m3f19.out	S23e1o\oe1m3f19.out	S23e3a\ae3m4519.out	S23e3c\ce3m4519.out	S23e3d\de3m4519.out	S23e3o\oe3m4519.out
S23e1a\ae1f6c20.out	S23e1d\de1f6c20.out	S23e1o\oe1f6c20.out	S23e3a\ae3f7820.out	S23e3c\ce3f7820.out	S23e3d\de3f7820.out	S23e3o\oe3f7820.out
S23e1a\ae1f9820.out	S23e1d\de1f9820.out	S23e1o\oe1f9820.out	S23e3a\ae3faa20.out	S23e3c\ce3faa20.out	S23e3d\de3faa20.out	S23e3o\oe3faa20.out
S23e1a\ae1m1420.out	S23e1d\de1m1420.out	S23e1o\oe1m1420.out	S23e3a\ae3m1420.out	S23e3c\ce3m1420.out	S23e3d\de3m1420.out	S23e3o\oe3m1420.out
S23e1a\ae1m4020.out	S23e1d\de1m4020.out	S23e1o\oe1m4020.out	S23e3a\ae3m4620.out	S23e3c\ce3m4620.out	S23e3d\de3m4620.out	S23e3o\oe3m4620.out
S23e1a\ae1f6d21.out	S23e1d\de1f6d21.out	S23e1o\oe1f6d21.out	S23e3a\ae3f7921.out	S23e3c\ce3f7921.out	S23e3d\de3f7921.out	S23e3o\oe3f7921.out
S23e1a\ae1f9921.out	S23e1d\de1f9921.out	S23e1o\oe1f9921.out	S23e3a\ae3fab21.out	S23e3c\ce3fab21.out	S23e3d\de3fab21.out	S23e3o\oe3fab21.out
S23e1a\ae1m1521.out	S23e1d\de1m1521.out	S23e1o\oe1m1521.out	S23e3a\ae3m1521.out	S23e3c\ce3m1521.out	S23e3d\de3m1521.out	S23e3o\oe3m1521.out
S23e1a\ae1m4121.out	S23e1d\de1m4121.out	S23e1o\oe1m4121.out	S23e3a\ae3m4721.out	S23e3c\ce3m4721.out	S23e3d\de3m4721.out	S23e3o\oe3m4721.out
S23e1a\ae1f6e22.out	S23e1d\de1f6e22.out	S23e1o\oe1f6e22.out	S23e3a\ae3f7a22.out	S23e3c\ce3f7a22.out	S23e3d\de3f7a22.out	S23e3o\oe3f7a22.out
S23e1a\ae1f9a22.out	S23e1d\de1f9a22.out	S23e1o\oe1f9a22.out	S23e3a\ae3fac22.out	S23e3c\ce3fac22.out	S23e3d\de3fac22.out	S23e3o\oe3fac22.out
S23e1a\ae1m1622.out	S23e1d\de1m1622.out	S23e1o\oe1m1622.out	S23e3a\ae3m1622.out	S23e3c\ce3m1622.out	S23e3d\de3m1622.out	S23e3o\oe3m1622.out
S23e1a\ae1m4222.out	S23e1d\de1m4222.out	S23e1o\oe1m4222.out	S23e3a\ae3m4822.out	S23e3c\ce3m4822.out	S23e3d\de3m4822.out	S23e3o\oe3m4822.out
S23e1a\ae1f6f23.out	S23e1d\de1f6f23.out	S23e1o\oe1f6f23.out	S23e3a\ae3f7b23.out	S23e3c\ce3f7b23.out	S23e3d\de3f7b23.out	S23e3o\oe3f7b23.out
S23e1a\ae1f9b23.out	S23e1d\de1f9b23.out	S23e1o\oe1f9b23.out	S23e3a\ae3fad23.out	S23e3c\ce3fad23.out	S23e3d\de3fad23.out	S23e3o\oe3fad23.out
S23e1a\ae1m1723.out	S23e1d\de1m1723.out	S23e1o\oe1m1723.out	S23e3a\ae3m1723.out	S23e3c\ce3m1723.out	S23e3d\de3m1723.out	S23e3o\oe3m1723.out
S23e1a\ae1m4323.out	S23e1d\de1m4323.out	S23e1o\oe1m4323.out	S23e3a\ae3m4923.out	S23e3c\ce3m4923.out	S23e3d\de3m4923.out	S23e3o\oe3m4923.out
S23e1a\ae1f7024.out	S23e1d\de1f7024.out	S23e1o\oe1f7024.out	S23e3a\ae3f7c24.out	S23e3c\ce3f7c24.out	S23e3d\de3f7c24.out	S23e3o\oe3f7c24.out
S23e1a\ae1f9c24.out	S23e1d\de1f9c24.out	S23e1o\oe1f9c24.out	S23e3a\ae3fae24.out	S23e3c\ce3fae24.out	S23e3d\de3fae24.out	S23e3o\oe3fae24.out
S23e1a\ae1m1824.out	S23e1d\de1m1824.out	S23e1o\oe1m1824.out	S23e3a\ae3m1824.out	S23e3c\ce3m1824.out	S23e3d\de3m1824.out	S23e3o\oe3m1824.out
S23e1a\ae1m4424.out	S23e1d\de1m4424.out	S23e1o\oe1m4424.out	S23e3a\ae3m4a24.out	S23e3c\ce3m4a24.out	S23e3d\de3m4a24.out	S23e3o\oe3m4a24.out
S23e1a\ae1f7125.out	S23e1d\de1f7125.out	S23e1o\oe1f7125.out	S23e3a\ae3f7d25.out	S23e3c\ce3f7d25.out	S23e3d\de3f7d25.out	S23e3o\oe3f7d25.out
S23e1a\ae1f9d25.out	S23e1d\de1f9d25.out	S23e1o\oe1f9d25.out	S23e3a\ae3faf25.out	S23e3c\ce3faf25.out	S23e3d\de3faf25.out	S23e3o\oe3faf25.out
S23e1a\ae1m1925.out	S23e1d\de1m1925.out	S23e1o\oe1m1925.out	S23e3a\ae3m1925.out	S23e3c\ce3m1925.out	S23e3d\de3m1925.out	S23e3o\oe3m1925.out
S23e1a\ae1m4525.out	S23e1d\de1m4525.out	S23e1o\oe1m4525.out	S23e3a\ae3m4b25.out	S23e3c\ce3m4b25.out	S23e3d\de3m4b25.out	S23e3o\oe3m4b25.out
S23e1a\ae1f7226.out	S23e1d\de1f7226.out	S23e1o\oe1f7226.out	S23e3a\ae3f7e26.out	S23e3c\ce3f7e26.out	S23e3d\de3f7e26.out	S23e3o\oe3f7e26.out
S23e1a\ae1f9e26.out	S23e1d\de1f9e26.out	S23e1o\oe1f9e26.out	S23e3a\ae3fb026.out	S23e3c\ce3fb026.out	S23e3d\de3fb026.out	S23e3o\oe3fb026.out
S23e1a\ae1m1a26.out	S23e1d\de1m1a26.out	S23e1o\oe1m1a26.out	S23e3a\ae3m1a26.out	S23e3c\ce3m1a26.out	S23e3d\de3m1a26.out	S23e3o\oe3m1a26.out
S23e1a\ae1m4626.out	S23e1d\de1m4626.out	S23e1o\oe1m4626.out	S23e3a\ae3m4c26.out	S23e3c\ce3m4c26.out	S23e3d\de3m4c26.out	S23e3o\oe3m4c26.out
S23e1a\ae1f7327.out	S23e1d\de1f7327.out	S23e1o\oe1f7327.out	S23e3a\ae3f7f27.out	S23e3c\ce3f7f27.out	S23e3d\de3f7f27.out	S23e3o\oe3f7f27.out
S23e1a\ae1f9f27.out	S23e1d\de1f9f27.out	S23e1o\oe1f9f27.out	S23e3a\ae3fb127.out	S23e3c\ce3fb127.out	S23e3d\de3fb127.out	S23e3o\oe3fb127.out
S23e1a\ae1m1b27.out	S23e1d\de1m1b27.out	S23e1o\oe1m1b27.out	S23e3a\ae3m1b27.out	S23e3c\ce3m1b27.out	S23e3d\de3m1b27.out	S23e3o\oe3m1b27.out
S23e1a\ae1m4727.out	S23e1d\de1m4727.out	S23e1o\oe1m4727.out	S23e3a\ae3m4d27.out	S23e3c\ce3m4d27.out	S23e3d\de3m4d27.out	S23e3o\oe3m4d27.out

EXP1-A	EXP1-D	EXP1-O	EXP3-A	EXP3-C	EXP3-D	EXP3-O
S23e1a\ae1f7428.out	S23e1d\de1f7428.out	S23e1o\oe1f7428.out	S23e3a\ae3f8028.out	S23e3c\ce3f8028.out	S23e3d\de3f8028.out	S23e3o\oe3f8028.out
S23e1a\ae1fa028.out	S23e1d\de1fa028.out	S23e1o\oe1fa028.out	S23e3a\ae3fb228.out	S23e3c\ce3fb228.out	S23e3d\de3fb228.out	S23e3o\oe3fb228.out
S23e1a\ae1m1c28.out	S23e1d\de1m1c28.out	S23e1o\oe1m1c28.out	S23e3a\ae3m1c28.out	S23e3c\ce3m1c28.out	S23e3d\de3m1c28.out	S23e3o\oe3m1c28.out
S23e1a\ae1m4828.out	S23e1d\de1m4828.out	S23e1o\oe1m4828.out	S23e3a\ae3m4e28.out	S23e3c\ce3m4e28.out	S23e3d\de3m4e28.out	S23e3o\oe3m4e28.out
S23e1a\ae1f7529.out	S23e1d\de1f7529.out	S23e1o\oe1f7529.out	S23e3a\ae3f8129.out	S23e3c\ce3f8129.out	S23e3d\de3f8129.out	S23e3o\oe3f8129.out
S23e1a\ae1fa129.out	S23e1d\de1fa129.out	S23e1o\oe1fa129.out	S23e3a\ae3fb329.out	S23e3c\ce3fb329.out	S23e3d\de3fb329.out	S23e3o\oe3fb329.out
S23e1a\ae1m1d29.out	S23e1d\de1m1d29.out	S23e1o\oe1m1d29.out	S23e3a\ae3m1d29.out	S23e3c\ce3m1d29.out	S23e3d\de3m1d29.out	S23e3o\oe3m1d29.out
S23e1a\ae1m4929.out	S23e1d\de1m4929.out	S23e1o\oe1m4929.out	S23e3a\ae3m4f29.out	S23e3c\ce3m4f29.out	S23e3d\de3m4f29.out	S23e3o\oe3m4f29.out
S23e1a\ae1f7630.out	S23e1d\de1f7630.out	S23e1o\oe1f7630.out	S23e3a\ae3f8230.out	S23e3c\ce3f8230.out	S23e3d\de3f8230.out	S23e3o\oe3f8230.out
S23e1a\ae1fa230.out	S23e1d\de1fa230.out	S23e1o\oe1fa230.out	S23e3a\ae3fb430.out	S23e3c\ce3fb430.out	S23e3d\de3fb430.out	S23e3o\oe3fb430.out
S23e1a\ae1m1e30.out	S23e1d\de1m1e30.out	S23e1o\oe1m1e30.out	S23e3a\ae3m1e30.out	S23e3c\ce3m1e30.out	S23e3d\de3m1e30.out	S23e3o\oe3m1e30.out
S23e1a\ae1m4a30.out	S23e1d\de1m4a30.out	S23e1o\oe1m4a30.out	S23e3a\ae3m5030.out	S23e3c\ce3m5030.out	S23e3d\de3m5030.out	S23e3o\oe3m5030.out
S23e1a\ae1f7731.out	S23e1d\de1f7731.out	S23e1o\oe1f7731.out	S23e3a\ae3f8331.out	S23e3c\ce3f8331.out	S23e3d\de3f8331.out	S23e3o\oe3f8331.out
S23e1a\ae1fa331.out	S23e1d\de1fa331.out	S23e1o\oe1fa331.out	S23e3a\ae3fb531.out	S23e3c\ce3fb531.out	S23e3d\de3fb531.out	S23e3o\oe3fb531.out
S23e1a\ae1m1f31.out	S23e1d\de1m1f31.out	S23e1o\oe1m1f31.out	S23e3a\ae3m1f31.out	S23e3c\ce3m1f31.out	S23e3d\de3m1f31.out	S23e3o\oe3m1f31.out
S23e1a\ae1m4b31.out	S23e1d\de1m4b31.out	S23e1o\oe1m4b31.out	S23e3a\ae3m5131.out	S23e3c\ce3m5131.out	S23e3d\de3m5131.out	S23e3o\oe3m5131.out
S23e1a\ae1f7832.out	S23e1d\de1f7832.out	S23e1o\oe1f7832.out	S23e3a\ae3f8432.out	S23e3c\ce3f8432.out	S23e3d\de3f8432.out	S23e3o\oe3f8432.out
S23e1a\ae1fa432.out	S23e1d\de1fa432.out	S23e1o\oe1fa432.out	S23e3a\ae3fb632.out	S23e3c\ce3fb632.out	S23e3d\de3fb632.out	S23e3o\oe3fb632.out
S23e1a\ae1m2032.out	S23e1d\de1m2032.out	S23e1o\oe1m2032.out	S23e3a\ae3m2032.out	S23e3c\ce3m2032.out	S23e3d\de3m2032.out	S23e3o\oe3m2032.out
S23e1a\ae1m4c32.out	S23e1d\de1m4c32.out	S23e1o\oe1m4c32.out	S23e3a\ae3m5232.out	S23e3c\ce3m5232.out	S23e3d\de3m5232.out	S23e3o\oe3m5232.out
S23e1a\ae1f7933.out	S23e1d\de1f7933.out	S23e1o\oe1f7933.out	S23e3a\ae3f8533.out	S23e3c\ce3f8533.out	S23e3d\de3f8533.out	S23e3o\oe3f8533.out
S23e1a\ae1fa533.out	S23e1d\de1fa533.out	S23e1o\oe1fa533.out	S23e3a\ae3fb733.out	S23e3c\ce3fb733.out	S23e3d\de3fb733.out	S23e3o\oe3fb733.out
S23e1a\ae1m2133.out	S23e1d\de1m2133.out	S23e1o\oe1m2133.out	S23e3a\ae3m2133.out	S23e3c\ce3m2133.out	S23e3d\de3m2133.out	S23e3o\oe3m2133.out
S23e1a\ae1m4d33.out	S23e1d\de1m4d33.out	S23e1o\oe1m4d33.out	S23e3a\ae3m5333.out	S23e3c\ce3m5333.out	S23e3d\de3m5333.out	S23e3o\oe3m5333.out
S23e1a\ae1f7a34.out	S23e1d\de1f7a34.out	S23e1o\oe1f7a34.out	S23e3a\ae3f8634.out	S23e3c\ce3f8634.out	S23e3d\de3f8634.out	S23e3o\oe3f8634.out
S23e1a\ae1fa634.out	S23e1d\de1fa634.out	S23e1o\oe1fa634.out	S23e3a\ae3fb834.out	S23e3c\ce3fb834.out	S23e3d\de3fb834.out	S23e3o\oe3fb834.out
S23e1a\ae1m2234.out	S23e1d\de1m2234.out	S23e1o\oe1m2234.out	S23e3a\ae3m2234.out	S23e3c\ce3m2234.out	S23e3d\de3m2234.out	S23e3o\oe3m2234.out
S23e1a\ae1m4e34.out	S23e1d\de1m4e34.out	S23e1o\oe1m4e34.out	S23e3a\ae3m5434.out	S23e3c\ce3m5434.out	S23e3d\de3m5434.out	S23e3o\oe3m5434.out
S23e1a\ae1f7b35.out	S23e1d\de1f7b35.out	S23e1o\oe1f7b35.out	S23e3a\ae3f8735.out	S23e3c\ce3f8735.out	S23e3d\de3f8735.out	S23e3o\oe3f8735.out
S23e1a\ae1fa735.out	S23e1d\de1fa735.out	S23e1o\oe1fa735.out	S23e3a\ae3fb935.out	S23e3c\ce3fb935.out	S23e3d\de3fb935.out	S23e3o\oe3fb935.out
S23e1a\ae1m2335.out	S23e1d\de1m2335.out	S23e1o\oe1m2335.out	S23e3a\ae3m2335.out	S23e3c\ce3m2335.out	S23e3d\de3m2335.out	S23e3o\oe3m2335.out
S23e1a\ae1m4f35.out	S23e1d\de1m4f35.out	S23e1o\oe1m4f35.out	S23e3a\ae3m5535.out	S23e3c\ce3m5535.out	S23e3d\de3m5535.out	S23e3o\oe3m5535.out
S23e1a\ae1f7c36.out	S23e1d\de1f7c36.out	S23e1o\oe1f7c36.out	S23e3a\ae3f8836.out	S23e3c\ce3f8836.out	S23e3d\de3f8836.out	S23e3o\oe3f8836.out
S23e1a\ae1fa836.out	S23e1d\de1fa836.out	S23e1o\oe1fa836.out	S23e3a\ae3fba36.out	S23e3c\ce3fba36.out	S23e3d\de3fba36.out	S23e3o\oe3fba36.out
S23e1a\ae1m2436.out	S23e1d\de1m2436.out	S23e1o\oe1m2436.out	S23e3a\ae3m2436.out	S23e3c\ce3m2436.out	S23e3d\de3m2436.out	S23e3o\oe3m2436.out
S23e1a\ae1m5036.out	S23e1d\de1m5036.out	S23e1o\oe1m5036.out	S23e3a\ae3m5636.out	S23e3c\ce3m5636.out	S23e3d\de3m5636.out	S23e3o\oe3m5636.out

EXP1-A	EXP1-D	EXP1-O	EXP3-A	EXP3-C	EXP3-D	EXP3-O
S23e1a\ae1f7d37.out	S23e1d\de1f7d37.out	S23e1o\oe1f7d37.out	S23e3a\ae3f8937.out	S23e3c\ce3f8937.out	S23e3d\de3f8937.out	S23e3o\oe3f8937.out
S23e1a\ae1fa937.out	S23e1d\de1fa937.out	S23e1o\oe1fa937.out	S23e3a\ae3fbb37.out	S23e3c\ce3fbb37.out	S23e3d\de3fbb37.out	S23e3o\oe3fbb37.out
S23e1a\ae1m2537.out	S23e1d\de1m2537.out	S23e1o\oe1m2537.out	S23e3a\ae3m2537.out	S23e3c\ce3m2537.out	S23e3d\de3m2537.out	S23e3o\oe3m2537.out
S23e1a\ae1m5137.out	S23e1d\de1m5137.out	S23e1o\oe1m5137.out	S23e3a\ae3m5737.out	S23e3c\ce3m5737.out	S23e3d\de3m5737.out	S23e3o\oe3m5737.out
S23e1a\ae1f7e38.out	S23e1d\de1f7e38.out	S23e1o\oe1f7e38.out	S23e3a\ae3f8a38.out	S23e3c\ce3f8a38.out	S23e3d\de3f8a38.out	S23e3o\oe3f8a38.out
S23e1a\ae1faa38.out	S23e1d\de1faa38.out	S23e1o\oe1faa38.out	S23e3a\ae3fbc38.out	S23e3c\ce3fbc38.out	S23e3d\de3fbc38.out	S23e3o\oe3fbc38.out
S23e1a\ae1m2638.out	S23e1d\de1m2638.out	S23e1o\oe1m2638.out	S23e3a\ae3m2638.out	S23e3c\ce3m2638.out	S23e3d\de3m2638.out	S23e3o\oe3m2638.out
S23e1a\ae1m5238.out	S23e1d\de1m5238.out	S23e1o\oe1m5238.out	S23e3a\ae3m5838.out	S23e3c\ce3m5838.out	S23e3d\de3m5838.out	S23e3o\oe3m5838.out
S23e1a\ae1f7f39.out	S23e1d\de1f7f39.out	S23e1o\oe1f7f39.out	S23e3a\ae3f8b39.out	S23e3c\ce3f8b39.out	S23e3d\de3f8b39.out	S23e3o\oe3f8b39.out
S23e1a\ae1fab39.out	S23e1d\de1fab39.out	S23e1o\oe1fab39.out	S23e3a\ae3fbd39.out	S23e3c\ce3fbd39.out	S23e3d\de3fbd39.out	S23e3o\oe3fbd39.out
S23e1a\ae1m2739.out	S23e1d\de1m2739.out	S23e1o\oe1m2739.out	S23e3a\ae3m2739.out	S23e3c\ce3m2739.out	S23e3d\de3m2739.out	S23e3o\oe3m2739.out
S23e1a\ae1m5339.out	S23e1d\de1m5339.out	S23e1o\oe1m5339.out	S23e3a\ae3m5939.out	S23e3c\ce3m5939.out	S23e3d\de3m5939.out	S23e3o\oe3m5939.out
S23e1a\ae1f8040.out	S23e1d\de1f8040.out	S23e1o\oe1f8040.out	S23e3a\ae3f8c40.out	S23e3c\ce3f8c40.out	S23e3d\de3f8c40.out	S23e3o\oe3f8c40.out
S23e1a\ae1fac40.out	S23e1d\de1fac40.out	S23e1o\oe1fac40.out	S23e3a\ae3fbe40.out	S23e3c\ce3fbe40.out	S23e3d\de3fbe40.out	S23e3o\oe3fbe40.out
S23e1a\ae1m2840.out	S23e1d\de1m2840.out	S23e1o\oe1m2840.out	S23e3a\ae3m2840.out	S23e3c\ce3m2840.out	S23e3d\de3m2840.out	S23e3o\oe3m2840.out
S23e1a\ae1m5440.out	S23e1d\de1m5440.out	S23e1o\oe1m5440.out	S23e3a\ae3m5a40.out	S23e3c\ce3m5a40.out	S23e3d\de3m5a40.out	S23e3o\oe3m5a40.out
S23e1a\ae1f8141.out	S23e1d\de1f8141.out	S23e1o\oe1f8141.out	S23e3a\ae3f8d41.out	S23e3c\ce3f8d41.out	S23e3d\de3f8d41.out	S23e3o\oe3f8d41.out
S23e1a\ae1fad41.out	S23e1d\de1fad41.out	S23e1o\oe1fad41.out	S23e3a\ae3fbf41.out	S23e3c\ce3fbf41.out	S23e3d\de3fbf41.out	S23e3o\oe3fbf41.out
S23e1a\ae1m2941.out	S23e1d\de1m2941.out	S23e1o\oe1m2941.out	S23e3a\ae3m2941.out	S23e3c\ce3m2941.out	S23e3d\de3m2941.out	S23e3o\oe3m2941.out
S23e1a\ae1m5541.out	S23e1d\de1m5541.out	S23e1o\oe1m5541.out	S23e3a\ae3m5b41.out	S23e3c\ce3m5b41.out	S23e3d\de3m5b41.out	S23e3o\oe3m5b41.out
S23e1a\ae1f8242.out	S23e1d\de1f8242.out	S23e1o\oe1f8242.out	S23e3a\ae3f8e42.out	S23e3c\ce3f8e42.out	S23e3d\de3f8e42.out	S23e3o\oe3f8e42.out
S23e1a\ae1fae42.out	S23e1d\de1fae42.out	S23e1o\oe1fae42.out	S23e3a\ae3fc042.out	S23e3c\ce3fc042.out	S23e3d\de3fc042.out	S23e3o\oe3fc042.out
S23e1a\ae1m2a42.out	S23e1d\de1m2a42.out	S23e1o\oe1m2a42.out	S23e3a\ae3m2a42.out	S23e3c\ce3m2a42.out	S23e3d\de3m2a42.out	S23e3o\oe3m2a42.out
S23e1a\ae1m5642.out	S23e1d\de1m5642.out	S23e1o\oe1m5642.out	S23e3a\ae3m5c42.out	S23e3c\ce3m5c42.out	S23e3d\de3m5c42.out	S23e3o\oe3m5c42.out
S23e1a\ae1f8343.out	S23e1d\de1f8343.out	S23e1o\oe1f8343.out	S23e3a\ae3f8f43.out	S23e3c\ce3f8f43.out	S23e3d\de3f8f43.out	S23e3o\oe3f8f43.out
S23e1a\ae1faf43.out	S23e1d\de1faf43.out	S23e1o\oe1faf43.out	S23e3a\ae3fc143.out	S23e3c\ce3fc143.out	S23e3d\de3fc143.out	S23e3o\oe3fc143.out
S23e1a\ae1m2b43.out	S23e1d\de1m2b43.out	S23e1o\oe1m2b43.out	S23e3a\ae3m2b43.out	S23e3c\ce3m2b43.out	S23e3d\de3m2b43.out	S23e3o\oe3m2b43.out
S23e1a\ae1m5743.out	S23e1d\de1m5743.out	S23e1o\oe1m5743.out	S23e3a\ae3m5d43.out	S23e3c\ce3m5d43.out	S23e3d\de3m5d43.out	S23e3o\oe3m5d43.out
S23e1a\ae1f8444.out	S23e1d\de1f8444.out	S23e1o\oe1f8444.out	S23e3a\ae3f9044.out	S23e3c\ce3f9044.out	S23e3d\de3f9044.out	S23e3o\oe3f9044.out
S23e1a\ae1fb044.out	S23e1d\de1fb044.out	S23e1o\oe1fb044.out	S23e3a\ae3fc244.out	S23e3c\ce3fc244.out	S23e3d\de3fc244.out	S23e3o\oe3fc244.out
S23e1a\ae1m2c44.out	S23e1d\de1m2c44.out	S23e1o\oe1m2c44.out	S23e3a\ae3m2c44.out	S23e3c\ce3m2c44.out	S23e3d\de3m2c44.out	S23e3o\oe3m2c44.out
S23e1a\ae1m5844.out	S23e1d\de1m5844.out	S23e1o\oe1m5844.out	S23e3a\ae3m5e44.out	S23e3c\ce3m5e44.out	S23e3d\de3m5e44.out	S23e3o\oe3m5e44.out
			S23e3a\ae3f9145.out	S23e3c\ce3f9145.out	S23e3d\de3f9145.out	S23e3o\oe3f9145.out
			S23e3a\ae3fc345.out	S23e3c\ce3fc345.out	S23e3d\de3fc345.out	S23e3o\oe3fc345.out
			S23e3a\ae3m2d45.out	S23e3c\ce3m2d45.out	S23e3d\de3m2d45.out	S23e3o\oe3m2d45.out
			S23e3a\ae3m5f45.out	S23e3c\ce3m5f45.out	S23e3d\de3m5f45.out	S23e3o\oe3m5f45.out

EXP1-A	EXP1-D	EXP1-O	EXP3-A	EXP3-C	EXP3-D	EXP3-O
			S23e3a\ae3f9246.out	S23e3c\ce3f9246.out	S23e3d\de3f9246.out	S23e3o\oe3f9246.out
			S23e3a\ae3fc446.out	S23e3c\ce3fc446.out	S23e3d\de3fc446.out	S23e3o\oe3fc446.out
			S23e3a\ae3m2e46.out	S23e3c\ce3m2e46.out	S23e3d\de3m2e46.out	S23e3o\oe3m2e46.out
			S23e3a\ae3m6046.out	S23e3c\ce3m6046.out	S23e3d\de3m6046.out	S23e3o\oe3m6046.out
			S23e3a\ae3f9347.out	S23e3c\ce3f9347.out	S23e3d\de3f9347.out	S23e3o\oe3f9347.out
			S23e3a\ae3fc547.out	S23e3c\ce3fc547.out	S23e3d\de3fc547.out	S23e3o\oe3fc547.out
			S23e3a\ae3m2f47.out	S23e3c\ce3m2f47.out	S23e3d\de3m2f47.out	S23e3o\oe3m2f47.out
			S23e3a\ae3m6147.out	S23e3c\ce3m6147.out	S23e3d\de3m6147.out	S23e3o\oe3m6147.out
			S23e3a\ae3f9448.out	S23e3c\ce3f9448.out	S23e3d\de3f9448.out	S23e3o\oe3f9448.out
			S23e3a\ae3fc648.out	S23e3c\ce3fc648.out	S23e3d\de3fc648.out	S23e3o\oe3fc648.out
			S23e3a\ae3m3048.out	S23e3c\ce3m3048.out	S23e3d\de3m3048.out	S23e3o\oe3m3048.out
			S23e3a\ae3m6248.out	S23e3c\ce3m6248.out	S23e3d\de3m6248.out	S23e3o\oe3m6248.out
			S23e3a\ae3f9549.out	S23e3c\ce3f9549.out	S23e3d\de3f9549.out	S23e3o\oe3f9549.out
			S23e3a\ae3fc749.out	S23e3c\ce3fc749.out	S23e3d\de3fc749.out	S23e3o\oe3fc749.out
			S23e3a\ae3m3149.out	S23e3c\ce3m3149.out	S23e3d\de3m3149.out	S23e3o\oe3m3149.out
			S23e3a\ae3m6349.out	S23e3c\ce3m6349.out	S23e3d\de3m6349.out	S23e3o\oe3m6349.out
			S23e3a\ae3f9650.out	S23e3c\ce3f9650.out	S23e3d\de3f9650.out	S23e3o\oe3f9650.out
			S23e3a\ae3fc850.out	S23e3c\ce3fc850.out	S23e3d\de3fc850.out	S23e3o\oe3fc850.out
			S23e3a\ae3m3250.out	S23e3c\ce3m3250.out	S23e3d\de3m3250.out	S23e3o\oe3m3250.out
			S23e3a\ae3m6450.out	S23e3c\ce3m6450.out	S23e3d\de3m6450.out	S23e3o\oe3m6450.out

A.3 Speech files provided for validation with variable delay

These speech files are in uncompressed format, 16-bit linear PCM, Intel byte ordering, at 8 kHz sample rate.

A.4 Conformance data sets

The data sets for the conformance tests are as follows.

Table A.1/P.563 – Conformance data sets

Test	Number of files	Data set	Type of test
1	1328	Downsampled by factor 2 from Supplement 23 using ITU-T Software Tools Library (version 2000, release 3)	Mandatory
2	n.a.	No data set defined. This test is open-ended, based on general, unknown data.	Mandatory

A.5 Conformance requirements

The test requirements are summarized in Table A.2. The requirements are based on the absolute difference in P.563 score between the implementation under test and the ANSI-C reference implementation, calculated for each file.

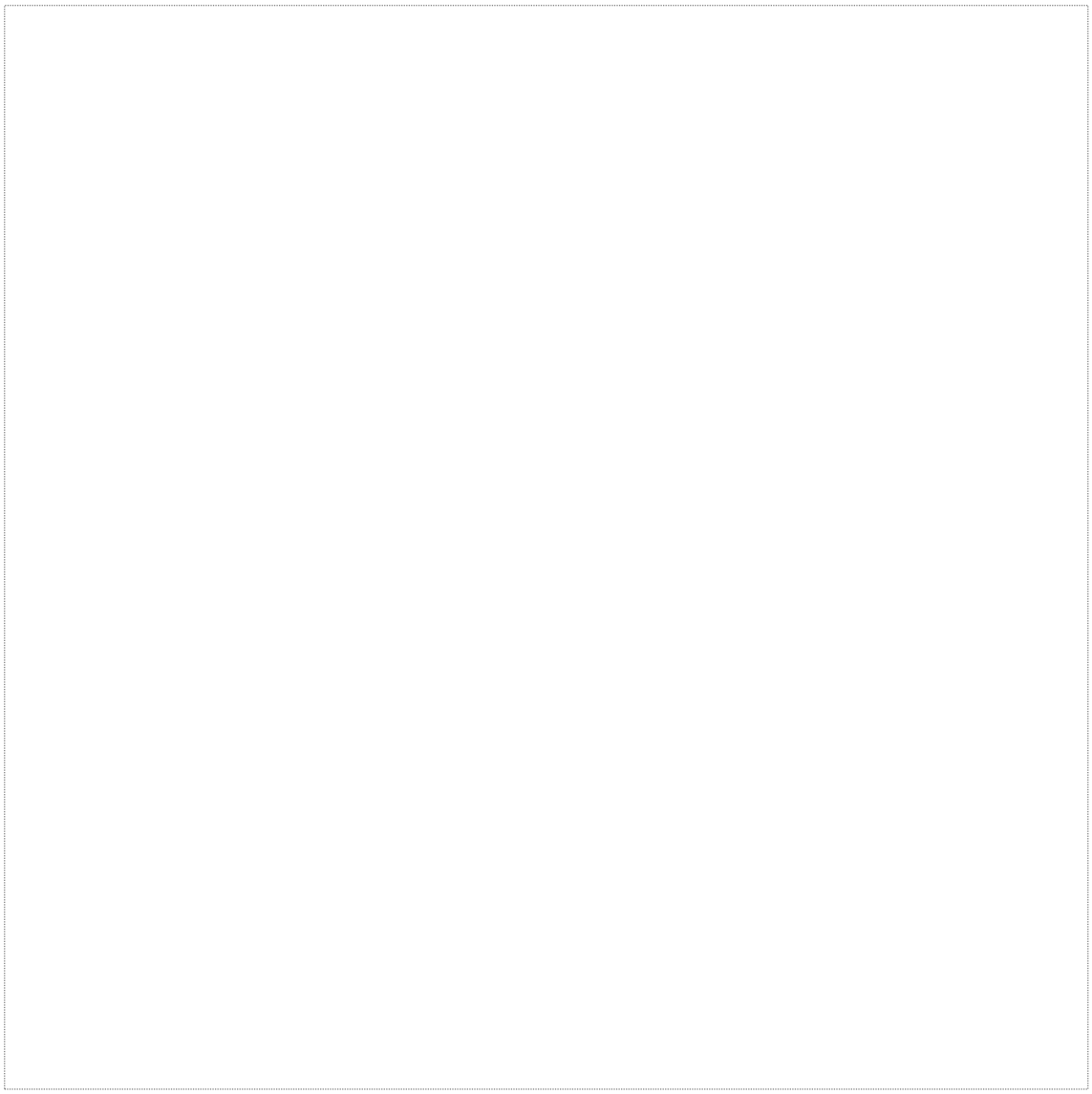
Table A.2/P.563 – Conformance requirements

Test	Number of file pairs	Lower threshold	Upper threshold	Type of test
1	1328	Difference may exceed 0.05 in not more than 13 files (1%).	Difference may not exceed 0.5 in any case.	Mandatory
2	No data set defined	Difference may exceed 0.05 in not more than 2% of cases.	Difference may not exceed 0.5 in any case.	Lower threshold is advisory. Upper threshold is mandatory.

NOTE – The double precision of P.563 implementation provided has been tested on Windows and Linux platforms and passes these conformance requirements.

A.6 Conformance test on unknown data

To prevent implementers from specifically tailoring an algorithm to conform to requirements for the files described above, a further test is available. An implementation of P.563 that conforms to this Recommendation must in at least 98% of cases give an output score that is within 0.05 of the model score given by the ANSI-C reference implementation. These cases must be based on speech files covering a representative sample of reasonable telephone network conditions, and must lie within the scope of P.563.



SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	**Telephone transmission quality, telephone installations, local line networks**
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure, Internet protocol aspects and Next Generation Networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2005