Technical Specification Group Services and System Aspects **TSGS#23(04)0075**
Meeting #23, Phoenix, Arizona (USA)
15-18 March 2004

3GPP TSG SA4#30 S4-040138
Malaga, Spain, 23-27 Feb 2004 Agenda Item: SA4 plenary

Title:	**Report from TSG SA WG4#30 to SA#23 plenary on SES codec selection**
Source:	**TSG SA WG4**
Agenda Item:	**7.4.3**
Document for:	**Information**
Contact:	**Kari Jarvinen**

Summary

This document provides a summary of the SES codec selection from SA4.

1 Introduction

SA4 has been working on the selection of a codec to recommend for Speech Enabled Services since October 2002 under the WID for SES [3]. The usual process of agreeing “design constrains” [4], “test and processing plan” [5] and “recommendation criteria” [6] was followed and completed before evaluating the candidates.

Two candidate codecs were proposed and evaluated:

1. ETSI Standard for the DSR Extended Advanced Front-end (ES 202 212) that can operate at both 8kHz and 16kHz
2. AMR and AMR-WB audio codec

Both candidates meet the design constraints.

The performance evaluations were conducted by two leading companies in the area of speech recognition, IBM and Scansoft. Results from these evaluations were presented at SA4#30 and are summarised here. The “recommendation criteria” have been applied and SA4 recommends the DSR codec for Speech Enabled Services.

2 ASR vendor evaluation results

ASR vendors IBM and Scansoft have completed evaluations according to the “Test and Processing plan” [5] and their results are presented in [7].

- At low data rate DSR provides an average of 36% relative reduction in word error rate compared to AMR4.75.
- At the high data rate at 8kHz DSR provides an average of 24% relative reduction in word error rate compared to AMR12.2.
- At the high data rate at 16kHz DSR provides an average of 31% relative reduction in word error rate compared to AMR-WB12.65.

According to the recommendation criteria [6]

- At the low data rate DSR is recommended.
- At the high data rate at 8kHz the result is in the “grey area”.
- At the high data rate at 16kHz DSR is recommended.

3 Informative Error Resilience Results

ASR vendors also provided informative results at 10% BLER in addition to those at 1% and 3% formally included in the recommendation criteria. The 10% BLER results are also included in [7]. These demonstrate that DSR is more robust to channel errors than AMR [17].

4 Informative Listening Tests

In LS from SA4 to SA on speech reconstruction [10] it was stated that "Based on the work done in ETSI Aurora [1,2], both the 8 and 16 kHz DSR codec versions are capable of reconstructing intelligible speech. Therefore, there is no need to carry out the intelligibility tests for the SES candidate codecs. Reconstruction quality of the SES codec candidates will be measured for informative purposes only." Accordingly Nokia and Ericsson have conducted listening quality tests for AMR and the DSR reconstruction.

ACR speech quality listening tests have been conducted in Finnish [11] and Chinese [12]. The results show that the quality of the DSR reconstruction is worse than AMR 4.75.

DCR tests were also conducted on the noisy speech samples; however, because of the presence of noise suppression in the DSR Advanced Front-end reconstruction the suitability of these tests is questionable. DCR tests are not appropriate for testing noisy speech samples when noise suppression is implemented.

5 Verification Plan

The verification plan has been agreed [17] and will be conducted by STMicroelectronics assisted by IBM. Verification is scheduled to be completed by 26th March.

6 Recommendation

According to the application of the SES recommendation criteria agreed at SA4 and SA [6]:

- **At the low data rate: DSR is recommended**
- At the high data rate at 8kHz the result is in the "grey area"
- **At the high data rate at 16kHz: DSR is recommended**

For the high data rate at 8kHz the DSR provides 24% improvement, which means that the results fall into the "grey area" (between 20% and 30% improvement). Since DSR is already selected at the low data rate at 8kHz it makes sense to also use DSR at the high data rate where it brings good performance improvement over AMR12.2 and also uses less than half the data rate (i.e. 5.6kbit/s for DSR cf 12.2kbit/s for AMR12.2).

It is therefore recommended that for Speech Enabled Services the DSR Extended Advanced Front-end should be used because it will bring substantially improved performance compared to using the voice channel.

AMR or AMR-WB may also be used for speech enabled services but the substantial performance advantages of DSR are noted.

For speech output back to the user in Speech Enabled Services then it is recommended that AMR or AMR-WB is used giving speech quality consistent with voice communications.

7 Conclusion

SA4 recommends that the DSR Extended Advanced Front-end should be used for Speech Enabled Services.

AMR or AMR-WB may also be used for these services but the substantial performance advantages of DSR are noted.

To update the release 6 specifications to include the DSR codec the following TS is brought to SA#23 for information and for approval at SA#24:

SP-040064 [S4-040054] “TS 26.243 ANSI-C code for the Fixed-Point Distributed Speech Recognition Extended Advanced Front-end” (note that this is a fixed point implementation of the ETSI Standard ES 202 212 [11])

S4-040137 is brought to SA#23 for information and contains information about the CRs to that will be brought to SA#24 for approval. These introduce optional Speech Enabled Services and the DSR codec that should be used and the AMR or AMR-WB that may be used.

S4-040136 CR to 26.235 for “Packet Switched Conversational Multimedia Applications: Default codecs”.

S4-040131 CR to 26.236 for “Packet switched conversational multimedia applications; Transport protocols”.

References

[1] TS 22.243 “Speech recognition framework for automated voice services; Stage 1”

[2] TR 22.977 “Feasibility study for speech enabled services”

[3] SP-020687 WID Codec Work to Support Speech Recognition Framework for Automated Voice Services (Rel-6)

[4] S4-030248 “Design Constraints for default codec for speech enabled services (SES)”, SA4

[5] S4-030395 “Test and Processing plan for default codec evaluation for speech enabled services (SES)”, SA4

[6] S4-030540 Recommendation Criteria for Default Codec for Speech Enabled Services (SES)”, SA4 & SP-030440 SA

[7] S4-040145 “SES Evaluation from ASR vendors (spreadsheet and informative data)”, ETSI

[8] S4-030708 “Complexity assessment of SES candidate codecs and justification of having met the design constraints”, Nokia, Ericsson

[9] S4-030710 “Fixed point complexity assessment and justification of having met the SES codec design constraints for the DSR Extended Advanced front-end candidate”, Motorola, France Telecom, Alcatel

[10] ETSI standard ES 202 050 “Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithm”, Oct 2002

[11] ETSI Standard ES 202 212 “Distributed Speech Recognition; Extended Advanced Front-end Feature Extraction Algorithm; Compression Algorithm, Back-end Speech Reconstruction Algorithm”, Nov 2003

[12] S4-040054 “Draft TS Software documentation for fixed-point DSR Extended Advanced Front-end”

[13] S4-030547 LS from SA4 to SA on “Assessment of the SES codecs ability to reconstruct speech”

[14] S4-040064 “Speech reconstruction assessment results”, Nokia

[15] S4-040069 “SES candidate codec speech reconstruction quality evaluation”, Ericsson

[16] S4-040153 “Draft SES Verification plan”, SA4

[17] S4-040152 “Results of error resilience for SES codec candidates**”,** France Telecom

[18] SP-040076 [S4-040137] “Proposed CRs from TSG SA WG4 to introduce SES to release 6 specifications”.

3GPP TSG SA4 meeting #25bis **S4-030248**
Berlin, Germany, 24. –28.Feb 2003 **Agenda Item: SES**

Title:	**Design Constraints for default codec for speech enabled services (SES)**
Source:	**SQ-SWG**
Contact:	**Bernhard Noé , Bernhard.Noe@alcatel.de**
Version:	**1.4**

Summary:

This document describes the design constraints for the codecs for speech enabled services

1. Sampling Rates

Sampling rates of 8 & 16kHz will be supported.

2. Complexity

The terminal side processing of the codec has to be able to be implemented within the resources of a typical mobile phone terminal. Accordingly the maximum complexity requirements for terminal side codec have been defined as shown in tables below. Table 1 shows complexity requirements for codec supporting 8kHz sampling rate and table 2 shows numbers for codec supporting 16 kHz sampling rate.

Measure	**Requirement**
WMOPS	Less than 25
ROM size	Less than 20 kwords
RAM size	Less than 7 kwords

Table 1: complexity and memory requirements for codec supporting 8 kHz sampling rate

Measure	**Requirement**
WMOPS	Less than 39
ROM size	Less than 34 kwords
RAM size	Less than 8 kwords

Table 2: complexity and memory requirements for codec supporting 16 kHz sampling rate The definition of the wMOPS measure and recommendations on how to estimate the computation and memory requirements can be found in ETSI Technical document [2]. A word is defined as 16bits. These complexity measures are for the front-end feature extraction and compression and the VAD.

ROM does not include program ROM.

3. Latency

The maximum codec latency requirement is 200ms, with the objective of 50 ms. This values contains the algorthmic delay introduced by the codec.

4. Data rate for the source codec

Voice enabled services need to be able to operate over a variety of channels. The following channels and datarates will at least be supported
a) For conversational class of service [4]:

- The GPRS single slot uplink (Coding scheme CS-1) channel. Here the maximum source data rate is 5.6 kbit/sec.
- The EGPRS single slot uplink (Coding scheme MCS -1) channel. Here the maximum source data rate is 6.4 kbit/sec.
- The Flexible Layer 1 (FLO) channel. Here the maximum data rate is expected to be between 6.4 and 8.4 kbit/sec.
- For UTRAN packet data channel the maximum source datarate is 24 kbit/sec.

It is assumed one 20ms frame within one RLC/MAC block.

b) For streaming and interactive class of service_

- For GPRS / EGPRS single slot uplink channel the maximum source datarate is 8 kbit/sec (assuming 10 frames per IP packet) or 7.5 kbit/sec (assuming 5 frames per IP packet) .
- For UTRAN packet data channel the maximum source datarate is 24 kbit/sec.

I

[1] ETSI SMG11 Tdoc SMG11 117/99, “Complexity verification report of the AMR codec, v2.0”, Alcatel, Philips, ST Microelectronics, Texas Instruments".
[2] ETSI SMG11 AMR-9 "AMR permanent document (AMR-9) Complexity and delay assessment v1.0", 23rd March 1998
[3] IETF RFC 3095: "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed".
[4] S4-030114.doc , TSG SA WG4 , Berlin, Germany, 24. –28.Feb 2003

3GPP TSG SA4 #26
Paris, France, 5-9 May 2003

S4-030395
Agenda Item: 7,9.1

Title:	**Test and processing plan for default codec evaluation for speech enabled services (SES)**
Source:	**SQ SWG**
Contact:	**David Pearce, bdp003@motorola.com**
Version:	1.10

Summary

This document proposes an update to the test & processing plan.

1. Introduction

Codec evaluation will be based on a framework which includes databases codecs and speech recognition engine. Evaluaters (as defined below) will be requested to use the same recognition engine for all codecs.

The following codecs have been submitted to the test:

1) AMR Codec and AMR WB Codec.
2) The ETSI DSR standard ES 202 050 for distributed speech recognition and its extension.

The evaluation framework for codec test is shown in Figure 1 and 2 below. Fig 1 applies for codecs with speech interface like a conventional speech codec and figure 2 applies for codecs with feature data interface like DSR optimised codecs.

The evaluation framework contains 2 processing stages:

- The candidate codec
- The speech recogniser from the evaluator

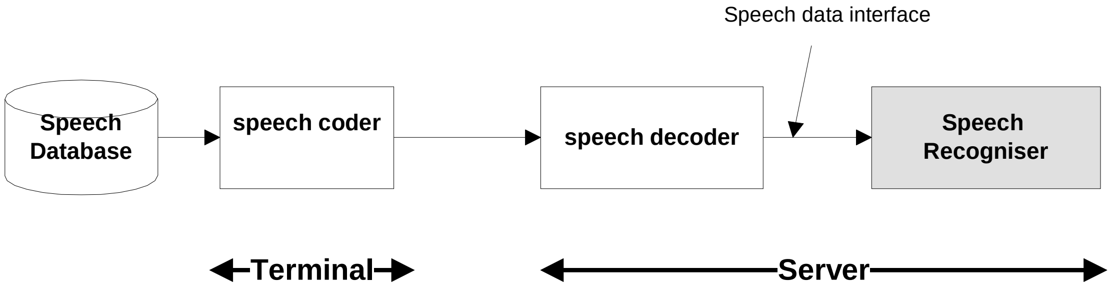


Figure 1: evaluation framework for speech codec (note that in this case the speech recognizer includes front-end and back-end decoder)

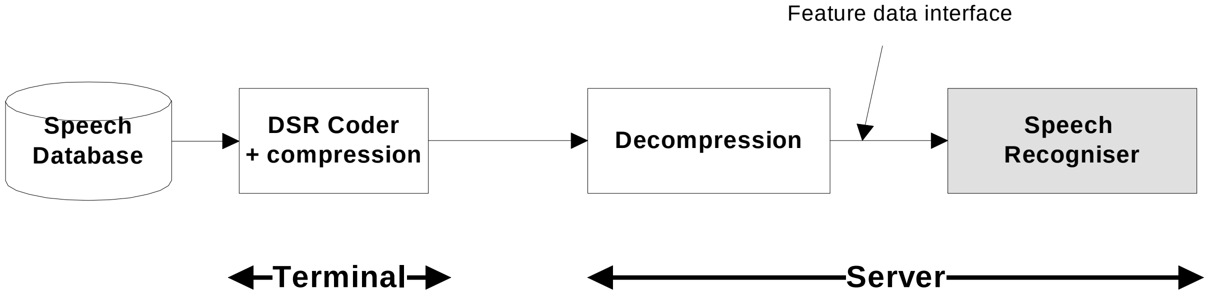


Figure 2: evaluation framework for DSR optimised codec (note that in this case the speech recogniser is back-end decoder only)

2. Recognition Engines

ASR vendors will perform the evaluations. Each ASR vendor will be provided with the database for the evaluation consisting of defined training and test sets (3GPP supplied databases). In addition ASR vendors proprietary databases will be used as well (ASR Vendor Supplied databases). Each ASR Vendor will run performance tests on these database considering both the AMR codec chain shown in figure 1 and the DSR optimised codec chain as shown in figure 2. ASR vendors have a free choice over the recogniser back-end configuration.

2.1 Recognizer for speech codecs based proposals

As AMR and AMR WB Codec can operate at several bitrates, a selection of bitrate has to be done for each test. Simulation of all AMR and AMR WB modes with all databases leads to practically unfeasible tests, therefore the number of Modes which are evaluated will be limited. For each selected bitrate the complete evaluation will be run on all databases. That means training and test is performed with that bitrate on the whole database. The following table shows the test conditions for AMR and AMR WB.

Bitrate	Codec	Sampling rate
4.75	AMR	8
12.2	AMR	8
12.65	AMR WB	16
23.85	AMR WB	16

Table 1: Test conditions for AMR and AMR WB Codec

2.1.1 Training & Testing

The training will be done using the coded & decoded speech data processed at the tested AMR bit rates as shown in the table above.

After speech decoder, any speech signal processing, e.g. compensating the coding artefacts or calculating the tonal language parameters, can be applied to the speech signal before calculating the actual recognition features.

2.2 Recognizer for DSR

Figure 2 shows the processing chain for a DSR front-end. The Advanced DSR Front-end (AFE) can operate with 8 or16kHz sampling rates. The feature extraction produces 12 mel-cepstral features (C1-C12), the zeroth order cepstral feature (C0) and log energy parameter (logE) at a 10ms frame rate. Recognisers may make use of either C0 or logE or both. The feature extraction is described in the ETSI standard document for ES 202 050 [1]. The static feature vector may be subject to further processing of the evaluators choice to produce dynamic features.. The software for the DSR standard contains an example implementing the recommended way of derivative calculation although evaluators are free to use their own alternatives.

In addition to the cepstral features the DSR AFE extension provides a pitch feature that may optionally be used as a feature to assist recognition when processing tonal languages. The raw pitch feature may be subject to further processing of the evaluators choice to produce tonal features to supplement the cepstral feature vector (e.g. smoothing or derivative calculation).

2.2.1 Training &Testing

Training should be performed with the features after compression and decompression with an error free channel. The same feature post-processing should be used for training as for recognition.

2.3 Usage of VAD for frame dropping

For the purpose of these performance evaluations no voice activity detector will be used for frame dropping either for discontinuous transmission at the terminal or at the recognition engine at the server.

3 Codec Evaluations

3.1 Recognition experiments under error-free channel

Testing has been arranged to cover a range of tasks as shown in the list below:

1. Connected digit recognition task

   Aurora-2
   Aurora-3
   Vendor 2 In-car Japanese, German, US English
   Vendor 1 US English in-car
   Vendor 1 Mandarin Embedded corpus (digits)

2. Sub-word trained model recognition task

   Nokia Mandarin Chinese name dialling (tone recognition ignored in performance scoring)
   Vendor 2 In-car

- Japanese,
- German,
- US English

Vendor 1 Mandarin Embedded Corpus (names /street names /organization names/commands)
Vendor 1 US English in car (commands, addresses, radio-controls, navigation, lifestyle information services and points-of-interest)

3. Tone confusability task

Nokia Mandarin Chinese name dialling (tone recognition taken into account in performance scoring)

4. Channel error task.

Aurora-3 Italian

Database Source	Database	Evaluator
3GPP supplied	Aurora-2	Vendor 2
	Aurora-3 German	Vendor 2
	Aurora-3 Spanish	Vendor 2
	Mandarin Name Dial	Vendor 1
	Aurora-2	Vendor 1
	Aurora-3 Spanish	Vendor 1
	Aurora-3 Italian	Vendor 1
ASR Vendor supplied	Mandarin Embedded PDA	Vendor 1
	US English In-Car	Vendor 1
	US English In-Car	Vendor 2
	German In-Car	Vendor 2
	Japanese In-Car	Vendor 2

Table 2: Table of databases for 8kHz Evaluations

Database Source	Database	Evaluator
3GPP Supplied		
	Aurora-3 Spanish	Vendor 2
	Mandarin Name Dial	Vendor 1
	Aurora-3 Spanish	Vendor 1
	Aurora-3 Italian	Vendor 1
ASR Vendor Supplied	Mandarin Embedded PDA	Vendor 1
	US English In-Car	Vendor 1
	US English In-Car	Vendor 2
	German In-Car	Vendor 2
	Japanese In-Car	Vendor 2

Table 3: Table of databases for 16kHz Evaluations

3.2 Recognition experiments under channel errors

For the purposes of testing under channel errors the Aurora-3 Italian database with the well-matched training and testing condition will be used.

Each codec will be tested under error free channel and with average channel BLERs of 1%, 3% and 10%. The BLERs of 1% and 3% will be used as part of the recommendation criteria while 10% is for informative purposes.

Recognition tests will be conducted by SpeechWorks and IBM using the supplied test sets. Models for these tests will be trained on the error free training data.

Codec for SES will be used with PSS over UTRAN, EGPRS and GPRS channels.

EGPRS (/GPRS) channel:
Simulations for GPSR and EGPRS will be combined as coding schemes for CS1 ..CS4 and MCS1 .. MCS4 are equivalent. Thereby consideration of EGPRS channel is sufficient.
The following parameters will be used:

- Typical Urban condition
- Scenarios: pedestrian with 3 km/h speed
- no FH

- unacknowledged mode
- One 20msec Frame per RTP/UDP Packet
- One RTP/UDP Packet per RLC/MAC Block

3 BLER patterns for EGPRS will be provided namely EG_EP1, EG_EP2 and EG_EP3
EG_EP1 = error condition in very good channel (mean BLER ~ 1 %)
EG_EP2 = error condition in good channel.(mean BLER ~ 3 %)
EG_EP3 = error condition in bad channel.(mean BLER ~ 10 %)

<u>UTRAN Channel:</u>

Error situation for UTRAN channel will be better (fast power control) than in EGPRS channel. The UTRAN channel is here approximated using the EG_EP1 error mask of the EGPRS channel.

<u>Format of Error Pattern</u>

Error Pattern will be provided which contain one Flag per Block indicating the error status of the block. An error insertion device is used to skip the frame if the flag equals TRUE. The error mask is applied to the aligned coded speech data. That means with the first speech file the error mask is read from the beginning, At the end of the speech file a pointer showing to the position in the error mask file is stored. When the next speech file is processed the error mask is read from the position the pointer refers to. This continues till the end of the error mask file is reached. Then the error mask file is rewinded and whole process starts again.

Error patterns will be applied to the test database for each candidate where one 20ms frame (corresponding to one frame per block) is deleted as indicted by the binary file. It is the responsibility of each party submitting a codec candidate for speech enabled services to provide the error insertion device and create the test database set for each channel and apply error mitigation as appropriate.
Each party should be able to show how error masks were applied and allow verification of test database if required by others.

8kHz

	Error Free	EG_EP1	EG_EP 2	EG_EP 3
DSR	X	X	X	X
AMR 4.75	X	X	X	X
AMR 12.2	X	X	X	X

16kHz

	Error Free	EG_EP 1	EG_EP 2	EG_EP 3
DSR	X	X	X	X
AMR-WB 12.65	X	X	X	X
AMR-WB 23.85	X	X	X	X

4. List of evaluators:

Test will be made by two ASR Vendors namely IBM and Speech Works acting as testlabs.

5. Cost of databases

Aurora-2	250 Euro
Aurora-3	1000 Euro per language
Mandarin database from High-Tech 863 program	6000 US Dollars

Appendix 1: Description of Evaluation Databases

A1. Introduction

Several databases are used for the evaluation framework. The composition of the databases considers the real world situation and the requirements of the recommendation criteria. Databases contain several languages including tonal languages for tonal confusion tests. The environmental conditions are considered by including databases with real world noise. The application requirements are considered by including several tasks like digit task and a name dialling task. The Databases are selected from both the former ETSI STQ Aurora databases, from additional proposals of SA4 member companies and proprietary databases proposed by ASR vendors. In the following sections a short description is provided for all used databases.

A2. Aurora 2: Noisy TI Digits database

The original high quality TIDigits database has been prepared by downsampling to 8kHz, filtering with G712 (which has frequency response representative of GSM terminal characteristics) and the controlled addition of noise to cover a range of signal to noise ratios (clean, 20,15,10,5,0,-5dB) and 8 different noise conditions. The database consists of connected digit sequences for American English talkers and clean and multi-condition training sets are defined. A full description of the database and the test framework is given in reference [2].

There are 3 test sets; set A contains noises seen in the multi-condition training data, set B contains noises that have not been seen in the training data and set C uses M-IRS filtering and noise addition to test the combination of convolutional distortion and noise.

A3. Aurora 3: Multilingual Speechdat-Car Digits database

Over a period of 4 years the ETSI STQ-Aurora working group has developed a set of evaluation databases and test criteria. Their purpose has been to support the characterisation and selection of Distributed Speech Recognition (DSR) front-ends. The databases cover a range of environments (typical for mobile device users) and languages. These have been made publicly available and are widely used. More details are given are given in reports sited in the references. The databases and procedures have been used for the competitive selection of the Advanced DSR front-end standard ES 202 050 and is summarised in references [11, 13]. For ETSI members further information is available at the ETSI Aurora web site [12].

Tests with Aurora 3 database allow to evaluate the performance of the codec on data that has been collected from speakers in a noisy environment. It tests the performance of the front-end with well matched training and testing as well as its performance in mismatched conditions as are likely to be encountered in deployed DSR systems. It also serves to test the front-end on a variety of languages: Finnish [1], Italian, Spanish, German, and Danish [3,4,5,6,7]. It is a small vocabulary task consisting of the digits selected from a larger database collection called SpeechDat-Car. See reference [3] as an example of for descriptions of these databases for Finnish with baseline performances for the mfccFE. The databases each have 3 experiments consisting of training and test sets to measure performance with:

***A) Well matched training and testing* -** Train & test with the hands-free microphone over the range of vehicle speeds so that the training and test sets cover similar range of noise conditions.

***B) Moderate mismatch training and testing* -** Train on only of a subset of the range of noises present in the test set. For example, hands-free microphone for lower speed driving conditions for training and hands free microphone at higher vehicle speeds for testing.

C) High mismatch training and testing - Model training with speech from close talking microphone. Hands-free microphone at range of vehicle speeds for testing.

[1] An consistency check of all Aurora 3 databases showed that SDC Finnish seems to have some problems. Therefore this database will not be considered [15].

A3.1. Distribution and Availability of Aurora Databases

All of the Aurora databases have been made available publicly through the European Language Distribution Agency ELRA [8].

Note: These databases are now widely accepted and used by the international speech research communities. Two special sessions on Noise Robustness have been organised at international conferences where the Aurora-2 and Aurora-3 databases have been used for the purposes of comparing the performance of different research algorithms. At EuroSpeech 2001 held in Aalborg, Denmark in Sept 2001, 20 papers were presented at the session and at ICSLP held in Denver, USA in Sept 2002, 29 papers were presented with results on these databases.

A4. Mandarin Chinese Database (proposal from Nokia)

Training database: Mandarin Chinese database from Chinese 863 High-Tech Program

Training set: 100 female and 100 make speakers. The database consists of 4 groups of different sentences; each group has 500-600 sentences approximately. Each speaker pronounces one group. The whole data is about 115 hours of speech.

Test database: Nokia Tonal language database

Test set: 10 male + 10 female speakers, 512 full name utterances per speaker, 124 different names in the vocabulary (two names differing only in tone count as different ones)

Test conditions are clean speech and speech with background noise.

A4.1. Distribution and Availability of Chinese Database([14])

Mandarin Chinese database which is used for training and is a public database collected by Chinese High-Tech 863 Program. Contact person Miss Xie Ying (yxie@htrdc.com, +86 10 68339172).
The test database is available from Nokia under NDA agreement exclusively for this standardisation.

A5. Vendor 2 proprietary database

These databases are recorded in car simultaneously from a far-field and a near-field microphone. The corpora include digit strings, commands, and names (for voice dialing). Evaluations will be conducted for three languages: US English, German, and Japanese.

A6. Vendor 1 proprietary database

A6.1. US English In-Car Corpus

The database is used for Vendor 1’s research experiments in embedded speech recognition. The recordings were made in stationary (with engine and a/c on) and moving (30mph and 60mph) cars with AKG-Q400 microphones placed on the mirror and visor. The corpus includes digit strings, commands, names and general English text. The training corpus is balanced for gender, accents, and other variations and is comprised of a very large number of speakers.

The test set also includes a large collection of speakers recorded in stationary and moving cars with a AKG-Q400 microphone placed on the mirror. The test corpus covers seven different tasks, digit strings, commands, addresses, radio-controls, navigation, vindigo (lifestyle information services) and points of interest.

A6.2. Mandarin Embedded Corpus

The database is designed for Mandarin speech recognition on handheld devices. This corpus is balanced for gender and other variations and is comprised of a very large set of speakers. The tasks covered in the corpus include digit strings/names /street names /organization names/commands etc. The test corpus is very similar to the training corpus. All recordings are made with a Lucent SD1100 microphone embedded into a PDA made in a university dormitory under usual background noise conditions.

References

[1] ETSI standard ES 202 050 “Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithm”, Oct 2002

[2] H G Hirsch & D Pearce, “The AURORA Experimental Framework for the Performance Evaluations of Speech Recognition Systems under Noisy Conditions”, ISCA ITRW ASR2000 “Automatic Speech Recognition: Challenges for the Next Millennium”; Paris, France, September 18-20, 2000

[3] AU/225/00 “Baseline Results for subset of SpeechDat-Car Finnish Database for ETSI STQ WI008 Advanced Front-end Evaluation”, Nokia, Jan 2000

[4] AU/237/00 “Description and baseline results for the SpeechDat-Car Italian Database”, Alcatel, April 2000

[5] AU/271/00 “Spanish SDC-Aurora Database for ETSI STQ Aurora WI008 Advanced DSR Front-End Evaluation: Description and Baseline Results”, UPC, Nov 2000

[6] AU/273/00 “Description and Baseline Results for the Subset of the Speechdat-Car German Database used for ETSI STQ Aurora WI008 Advanced DSR Front-end Evaluation”, Texas Instruments, Dec 2001

[7] AU/378/01 “Danish SpeechDat-Car Digits Database for ETSI STQ-Aurora Advanced DSR”, Aalborg University, Jan 2001.

[8] http://www.icp.inpg.fr/ELRA/home.html the ELRA home page

[9] deleted reference

[10] Recommendation Criteria for default codec for speech enabled services (SES) S4-030075, 3GPP TSG SA4 meeting #25

[11] AU/372/01 “Overview of Evaluation Criteria for Advanced DSR front-ends, Version 8”, Motorola, Dec 2001

[12] ETSI STQ-Aurora document archive: http://docbox.etsi.org/STQ/stq-aurora

[13] David Pearce, “Developing the ETSI Aurora Advanced Distributed Speech Recognition Front-end & What Next?”, IEEE Automatic Speech Recognition and Understanding Workshop; ASRU 2001, Madonna di Campiglio, Italy, Dec 2001

[14] S4-020755 3GPP TSG SA4 meeting #25

[15] S4-030110 “Evaluation of usability of Aurora 3 databases”3GPP TSG SA4 meeting #25bis

[16] S4-030114 Reply LS on Transmission Aspects for Speech Enabled Services (SES)

3GPP TSG SA4 #27 **S4-030540**
Munich, Germany, 7-11 July 2003 **Agenda Item: SES**

Title: **Recommendation Criteria for Default Codec for Speech Enabled Services (SES)**
Source: **SQ SWG**
Contact: **David Pearce, bdp003@motorola.com**
Version: 1.0

Summary

This document provides the recommendation criteria for the default codec for speech enabled services (SES) as agreed at SQ SWG, SA4#27.

Updated to remove the 16kHz Mandarin Name dialling task and include agreed values for recommendations.

1. Introduction

This document defines recommendation criteria for the selection of the default codec for speech enabled services. These criteria are based on the design constrains [1] and performance evaluations described in the test and processing plan [2]. The recommendation is based on speech recognition performance and the details of the scoring system are described below.

2. Recognition performance

2.1 Overview

The set of databases used for the evaluations are defined in the Test and Processing Plan [2]. Each of these databases contains different types speech material covering a variety of tasks, environments and languages. Recommendation will be based on a score obtained from the recognition performance measured on each of these different databases. Section 2.3 describes how the scores from all the individual databases are combined using a weighting table (see also appendix 2).

2.2 Scoring on individual databases

For each database the reference performance is measured as the word error rate obtained from the ASR vendor’s system. This is the performance obtained from a state-of-the-art system from the ASR vendor assuming a transparent channel.

The performance (word error rate) on a given database is also measured with the ASR vendors system for a codec under test as described in the test and processing plan.

Scoring for tests performed with channel BLER described in section 3.1.2 of [2] will also be computed in a similar way. Note that only BLER of 1% and 3% are considered as part of the recommendation criteria.

2.3 Performance metric over all databases

The overall performance will be determined by averaging the absolute word error rate using the weightings presented in tables A2.1 for 8kHz sampling rate and A2.2 for 16kHz sampling rate of Appendix 2. The result of this weighted average is an overall measure of the average word error rate for each codec. This metric is called the “average word error rate”.

2.4 Comparisons between codecs

2.4.1 Low data-rate codec comparison

The two codecs under consideration at low data-rate are AMR 4.75 and DSR AFE with extension (5.6kbit/s). Only 8kHz sampling rate is considered since there is no AMR-WB codec at low data rate.

Table A2.1 in Appendix 2 shows the list of databases that will be tested and the weightings to be given to the scores obtained for each of these databases.

2.4.2 High data-rate codec comparison

At high data-rates the comparisons are made separately at 8kHz and 16kHz sampling rates.

2.4.2.1 8kHz sampling rate

The two codecs under consideration at high data-rate at 8kHz sampling are AMR 12.2 & DSR AFE and extension (5.6kbit/s).

Table A2.1 in Appendix 2 shows the list of databases that will be tested and the weightings to be given to the scores obtained for each of these databases.

2.4.2.2 16kHz sampling rate

The two codecs under consideration at high data-rate at 16kHz sampling are AMR-WB 12.65, & DSR AFE (5.6kbit/s).

Table A2.2 in Appendix 2 shows the list of databases that will be tested and the weightings to be given to the scores obtained for each of these databases.

3. Recommendation criteria

The recommendation procedure will consist of the following:

1. Candidates not compliant with all Design Constraints will be excluded from further consideration. (For the selection meeting, all candidates must provide justification document for meeting the Design Constraints.)

2. For the low data-rate comparison:
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR 4.75kbps codec is more than 35% then the DSR codec and its extension will be recommended.
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR 4.75kbps codec is less than 20% then the DSR codec will not be recommended.
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR 4.75kbps codec is less than 20% then AMR will be recommended.
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR 4.75kbps codec is between 20% and 35% then the performance results will be further considered by SA4 and if there is no consensus the results will be passed to SA for decision on what recommendation to make.

3. For the high data-rate comparison at 8kHz:
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR 12.2kbps codec is more than 30% then the DSR codec and its extension will be recommended.
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR 12.2kbps codec is less than 20% then the DSR codec will not be recommended.
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR 12.2kbps codec is less than 20% then AMR will be recommended.
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR 12.2kbps codec is between 20% and 30% then the performance results will be further considered by SA4 and if there is no consensus the results will be passed to SA for decision on what recommendation to make.

4. For the high data-rate comparison at 16kHz:
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR-WB codec is more than 25% then the DSR codec and its extension will be recommended.
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR-WB codec is less than 15% then the DSR codec will not be recommended.
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR-WB codec is less than 15% then AMR-WB will be recommended.
   - If the relative reduction in average word error rate for the DSR AFE codec and its extension compared to the AMR-WB codec is between 15% and 25% then the performance results will be further considered by SA4 and if there is no consensus the results will be passed to SA for decision on what recommendation to make.

References

[1] Design Constraints for default codec for speech enabled services (SES)
Tdoc S4-030248
3GPP TSG SA4 meeting #25bis, Berlin, Germany, 24-28 Feb 2003

[2] Test and Processing plan for default codec evaluation for speech enabled services (SES),
Tdoc S4-030395
3GPP TSG SA4 meeting #26, Paris, France, 5-9 May 2003

Appendix 1: Weighting scheme for results on each database

Each database in the test and processing plan [2] produces a set of results for different training conditions and test sets. The weighting scheme to be used to combine the different results to give a single average performance on each database is defined below

1. 3GPP supplied databases

1.1 Aurora 2

Database	Aurora 2		
Test Set	Set A	Set B	Set C
Weight of the test set	40 %	40 %	20 %

Table A1: Weighting scheme within the databases Aurora 2

Multicondition and clean trained results to be weighted equally.

2.2 Aurora 3

For the Aurora 3 databases there are three test sets, well matched, medium mismatch and high mismatch. These will be weighted equally.

2. ASR vendor supplied databases

Test sets within the ASR vendor supplied databases will be weighted equally.

Appendix 2: Weighting of evaluation databases

Task	**Database**	**Evaluator**	**Task Weight**	**Database Weight**
Digits	Aurora-3 German	Vendor 2	3/10	1/11
	Aurora-3 Spanish	Vendor 2		1/11
	Aurora-2	Vendor 2		1/11
	Aurora-3 Italian	Vendor 1		1/11
	Aurora-3 Spanish	Vendor 1		1/11
	Aurora-2	Vendor 1		1/11
	US English In-Car (digit test)	Vendor 2		1/11
	German In-Car (digit test)	Vendor 2		1/11
	Japanese In-Car (digit test)	Vendor 2		1/11
	US English In-Car (digit test)	Vendor 1		1/11
	Mandarin Embedded PDA (digit test set)	Vendor 1		1/11
subword	Mandarin Embedded PDA **(**names /street names /organization names/commands)	Vendor 1	4/10	1/6
	US English In-Car (commands, addresses, radio-controls, navigation, lifestyle information services and points-of-interest)	Vendor 1		1/6
	US English In-Car	Vendor 2		1/6
	German In-Car	Vendor 2		1/6
	Japanese In-Car	Vendor 2		1/6
	Mandarin Name dialling (baseform test)	Vendor 1		1/6
Tone confusability	Mandarin Name dialling (tone confusable test)	Vendor 1	1/10	1
Channel errors	1% BLER	Vendor 1	2/10	¼
	3% BLER	Vendor 1		¼
	1% BLER	Vendor 2		¼
	3% BLER	Vendor 2		¼

Table A2.1: Weighting of evaluation databases at 8kHz

Task	**Database**	**Evaluator**	**Task Weight**	**Database Weight**
Digits			3.5/10	
	Aurora-3 Spanish	Vendor 2		1/8
	Aurora-3 Italian	Vendor 1		1/8
	Aurora-3 Spanish	Vendor 1		1/8
	US English In-Car (digit test)	Vendor 2		1/8
	German In-Car (digit test)	Vendor 2		1/8
	Japanese In-Car (digit test)	Vendor 2		1/8
	US English In-Car (digit test)	Vendor 1		1/8
	Mandarin Embedded PDA (digit test set)	Vendor 1		1/8
subword	Mandarin Embedded PDA **(**names /street names /organization names/commands)	Vendor 1	4.5/10	1/5
	US English In-Car (commands, addresses, radio-controls, navigation, lifestyle information services and points-of-interest)	Vendor 1		1/5
	US English In-Car	Vendor 2		1/5
	German In-Car	Vendor 2		1/5
	Japanese In-Car	Vendor 2		1/5
Channel errors	1% BLER	Vendor 1	2/10	¼
	3% BLER	Vendor 1		¼
	1% BLER	Vendor 2		¼
	3% BLER	Vendor 2		¼

Table A2.2: Weighting of evaluation databases at 16kHz

Appendix 3: Illustration of recommendation based on relative improvement

					Relative improvement					
AMR error rate	**10.0**	**15.0**	**20.0**	**25.0**	**30.0**	**35.0**	**40.0**	**50.0**	**60.0**	**70.0**
40	36.0	34.0	32.0	30.0	28.0	26.0	24.0	20.0	16.0	12.0
35	31.5	29.8	28.0	26.3	24.5	22.8	21.0	17.5	14.0	10.5
30	27.0	25.5	24.0	22.5	21.0	19.5	18.0	15.0	12.0	9.0
25	22.5	21.3	20.0	18.8	17.5	16.3	15.0	12.5	10.0	7.5
20	18.0	17.0	16.0	15.0	14.0	13.0	12.0	10.0	8.0	6.0
18	16.2	15.3	14.4	13.5	12.6	11.7	10.8	9.0	7.2	5.4
16	14.4	13.6	12.8	12.0	11.2	10.4	9.6	8.0	6.4	4.8
14	12.6	11.9	11.2	10.5	9.8	9.1	8.4	7.0	5.6	4.2
12	10.8	10.2	9.6	9.0	8.4	7.8	7.2	6.0	4.8	3.6
10	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.0	4.0	3.0
9	8.1	7.7	7.2	6.8	6.3	5.9	5.4	4.5	3.6	2.7
8	7.2	6.8	6.4	6.0	5.6	5.2	4.8	4.0	3.2	2.4
7	6.3	6.0	5.6	5.3	4.9	4.6	4.2	3.5	2.8	2.1
6	5.4	5.1	4.8	4.5	4.2	3.9	3.6	3.0	2.4	1.8
5	4.5	4.3	4.0	3.8	3.5	3.3	3.0	2.5	2.0	1.5
4	3.6	3.4	3.2	3.0	2.8	2.6	2.4	2.0	1.6	1.2
3	2.7	2.6	2.4	2.3	2.1	2.0	1.8	1.5	1.2	0.9
2	1.8	1.7	1.6	1.5	1.4	1.3	1.2	1.0	0.8	0.6
1	0.9	0.85	0.80	0.75	0.70	0.65	0.6	0.50	0.40	0.30
0.5	0.5	0.43	0.40	0.38	0.35	0.33	0.3	0.25	0.20	0.15
0.1	0.1	0.09	0.08	0.08	0.07	0.07	0.1	0.05	0.04	0.03

TSG-SA4#27 meeting
July 7-11, 2003, Munich, Germany

Tdoc S4 (03)0547
Agenda Item: 8, 10

Title:	LS on "**Assessment of the SES codecs ability to reconstruct speech**"
Response to:	Reply to the task assigned from SA#20 to SA4
Source:	SA4
To:	SA#21
Cc:	-

Contact Person:

Name:	Olli Viikki
Tel. Number:	+358 7180 08000
E-mail Address:	olli.viikki@nokia.com

Attachments: none

1. Overall Description:

There are two aspects to be considered when assessing the speech reconstruction capability of the SES (Speech Enabled Services) codecs. *Intelligibility* and *quality* are two separate quantities which can be determined in the different type of listening experiments. The capability to reconstruct intelligible speech is regarded as a basic requirement for any codec to be used in real-world systems. Quality is measured as it influences the pleasantness of user experience.

The following codecs have been submitted as the candidate codecs for SES:

- AMR Codec and AMR WB Codec.
- The ETSI DSR standard ES 202 050 for distributed speech recognition and its extension.

2. Conclusions:

Based on the work done in ETSI Aurora [1,2], both the 8 and 16 kHz DSR codec versions are capable of reconstructing intelligible speech. Therefore, there is no need to carry out the intelligibility tests for the SES candidate codecs. Reconstruction quality of the SES codec candidates will be measured for informative purposes only.

3. Date of Next SA4 Meetings:

SA4#29	24th – 28th November 2003 TBD

References:

[1] S4-030544, "SES DSR intelligibility testing DRT report by Dynastat"
[2] S4-030545, "Extended advanced DSR front-end reconstruction intelligibility tests by ETSI"

3GPP TSG-SA4#29 meeting
November 24-28, 2003, Tampere, Finland

Tdoc S4-030708

Source: Nokia

Title: Complexity assessment of SES candidate codecs and justification of having met the design constraints

Agenda item: **7**

---

1. Introduction

SES work item work plan requires the codec complexity assessment and justification of having met the design constraints given for all the candidate codecs by October 31 2003 [4].

This document provides the complexity assessment of AMR and AMR-WB codecs for SES services. In addition, the justification of having met the design constraints is given.

2. Design constraints

The permanent document S4-030248 contains the design constraints for default codec for speech enabled services (SES) [3].

2.1 Complexity

The following table summarises the constraints and lists the corresponding figures for fixed-point implementation of AMR and AMR-WB speech codecs.

The full analysis and characterisation of AMR and AMR-WB codecs are available in Technical Reports TR 26.975 [1] and TS 26.976 [2], respectively.

Measure	**Requirement**	**AMR**
WMOPS	Less than 25	15,33 (worst case)
ROM size	Less than 20 kwords	19,807
RAM size	Less than 7 kwords	5,280

Table 1: complexity and memory requirements for codec supporting 8 kHz sampling rate

Measure	**Requirement**	**AMR-WB**
WMOPS	Less than 39	38,97 (worst case)
ROM size	Less than 34 kwords	13,109
RAM size	Less than 8 kwords	7,101

Table 2: complexity and memory requirements for codec supporting 16 kHz sampling rate

Conclusion: Complexity requirements are met.

2.2 Latency

Requirement:

The maximum codec latency SES is 200 ms, with the objective of 50 ms.

Candidate codec latency:

The algorithmic delay of both AMR and AMR-WB codecs is 25 ms.

Conclusion: Latency requirement and objective is met.

2.3 Data rate for the source codec

Requirements:

Voice enabled services need to be able to operate over a variety of channels.
The following channels and data rates will at least be supported

a) For conversational class of service:

The GPRS single slot uplink (Coding scheme CS-1) channel.
Here the maximum source data rate is 5.6 kbit/sec.

The EGPRS single slot uplink (Coding scheme MCS -1) channel.
Here the maximum source data rate is 6.4 kbit/sec.

The Flexible Layer 1 (FLO) channel. Here the maximum data rate is expected to be between 6.4 and 8.4 kbit/sec.

For UTRAN packet data channel the maximum source data rate is 24 kbit/sec.

It is assumed one 20ms frame within one RLC/MAC block.

b) For streaming and interactive class of service

For GPRS / EGPRS single slot uplink channel the maximum source data rate is 8 kbit/sec (assuming 10 frames per IP packet) or 7.5 kbit/sec (assuming 5 frames per IP packet).

For UTRAN packet data channel the maximum source data rate is 24 kbit/sec.

Candidate codec data rates:

AMR codec has data rates ranging from 4.75 to 12.2 kbit/s. Using the RTP payload defined in IETF RFC 3267 [5] the lowest source data rate for the 4.75 mode with 20 ms packets is 5.6 kbit/s.

Conclusion: AMR codec can be used all the channels mentioned above.

AMR-WB codec has data rates ranging from 6.6 to 23.85 kbit/s.

Conclusion: AMR-WB codec can be used in UTRAN packet data channel.

3. Justification

As stated in Section 2, AMR and AMR-WB codecs meet all the design constraints. Hence, both AMR and AMR-WB are fully compliant for the SES service.

4. References

[1] TR 26.975 “Performance characterization of the AMR peech codec”

[2] TR 26.976 “Performance characterization of the Adaptive Multi-Rate Wideband (AMR-WB) speech codec”

[3] S4-030248 “Design Constraints for default codec for speech enabled services (SES)”

[4] S4-030542 “SES workplan Version 7.0”

[5] IET RFC 3267 “Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs”

3GPP TSG-SA4 meeting TBD **Tdoc S4-030710**

Source: Alcatel, France Telecom, Motorola

Title: Fixed point complexity assessment and justification of having met the SES codec design constraints for the DSR Extended Advanced front-end candidate.

1. Introduction

This document provides the fixed point complexity estimate and the justification of having met the SES design constraints for the DSR Extended Advanced front-end (X-AFE).

2. Design constraints

The permanent document S4-030248 contains the design constraints for default codec for speech enabled services (SES) [1].

2.1 Complexity

The following table summarises the constraints and lists the corresponding figures for fixed-point implementation.

Measure	Requirement	DSR X-AFE at 8kHz
WMOPS	Less than 25	24.07
ROM size	Less than 20 kwords	7091
RAM size	Less than 7 kwords	6665

Table 1: complexity and memory requirements for codec supporting 8 kHz sampling rate

Measure	Requirement	DSR X-AFE at 16kHz
WMOPS	Less than 39	30.79
ROM size	Less than 34 kwords	7482
RAM size	Less than 8 kwords	7595

Table 2: complexity and memory requirements for codec supporting 16 kHz sampling rate

RAM/ROM figures were obtained using the methodology outlined in [4]. ROM is data only and therefore doesn’t include code. Both RAM/ROM numbers are expressed as 16-bit word.

WMOPS are based on the use of the set of basic ETSI fixed-point operators and associated weights. The speech data files used for this assessment were the 2640 files of the Aurora-3 Finnish database.

Conclusion: Complexity requirements are met.

2.2 Latency

Requirement:

The maximum codec latency SES is 200 ms, with the objective of 50 ms.

Candidate codec latency*:*

The algorithmic delay of the DSR Advanced front-end and the DSR Extended Advanced DSR front end is 62.5 ms.

Conclusion: Latency requirement is met.

2.3 Data rate for the source codec

Requirements:

Voice enabled services need to be able to operate over a variety of channels.
The following channels and data rates will at least be supported

a) For conversational class of service:

The GPRS single slot uplink (Coding scheme CS-1) channel.
Here the maximum source data rate is 5.6 kbit/sec.

The EGPRS single slot uplink (Coding scheme MCS -1) channel.
Here the maximum source data rate is 6.4 kbit/sec.

The Flexible Layer 1 (FLO) channel. Here the maximum data rate is expected to be between 6.4 and 8.4 kbit/sec.

For UTRAN packet data channel the maximum source data rate is 24 kbit/sec.

It is assumed one 20ms frame within one RLC/MAC block.

b) For streaming and interactive class of service

For GPRS / EGPRS single slot uplink channel the maximum source data rate is 8 kbit/sec (assuming 10 frames per IP packet) or 7.5 kbit/sec (assuming 5 frames per IP packet).

For UTRAN packet data channel the maximum source data rate is 24 kbit/sec.

Candidate codec data rates:

The DSR Advanced front-end has a source data rate of 4.8kbit/sec (12 bytes per 20ms frame pair) and the Extended Advanced DSR front-end **5.6kbit/sec** (14 bytes per 20ms frame pair). The data rate is the same for both 8 and 16kHz sampling rates.

Note that the complete DSR codec can be run over any of the channels (i.e. The optimal recognition performance is obtained whether over GPRS, EGPRS or UTRANS.) This is likely to be important to deliver similar experience to customers for SES services whether this is over GPRS or UMTS.

Conclusion: The Advanced DSR front-end and the DSR Extended Advanced front-end can be used over any of the above channels.

3. Justification

The DSR Advanced Front-end and the DSR Extended Advanced front-end codecs meet all the design constraints.

4. References

[1] S4-030248 “Design Constraints for default codec for speech enabled services (SES)”

[2] ETSI standard ES 202 050 “Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithm”, Oct 2002

[3] ETSI standard ES 202 212 “Distributed Speech Recognition; Extended Advanced Front-end Feature Extraction Algorithm; Compression Algorithm”, publication expected Nov 2003

[4] AMR permanent document (AMR-9), “Complexity and Delay Assessment”, SMG11 AM/98

3GPP TSG SA4#30 **S4-040054**
Malaga, Spain, 23-27 Feb 2004 **Agenda Item: SES**

Title:	**Draft TS Software documentation for fixed-point DSR Extended Advanced Front-end**
Source:	**Motorola, France Telecom, Alcatel**
Contact:	**David Pearce, bdp003@motorola.com**
Version:	1

Summary

The companion document in the same zip file contains the draft of the TS for the Software documentation for the fixed-point DSR Extended Advanced Front-end.

3GPP TS 26.243 V1.0.0 (2004-03)

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
ANSI-C code for the Fixed-Point Distributed Speech Recognition Extended Advanced Front-end;
(Release 6)

GLOBAL SYSTEM FOR
MOBILE COMMUNICATIONS

The present document has been developed within the 3rd Generation Partnership Project (3GPP $^{\text{TM}}$) and may be further elaborated for the purposes of 3GPP.

The present document has not been subject to any approval process by the 3GPP Organizational Partners and shall not be implemented.
This Specification is provided for future development work within 3GPP only. The Organizational Partners accept no liability for any use of this Specification.
Specifications and reports for implementation of the 3GPP $^{\text{TM}}$ system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

AMR, CODEC, Adaptive Multi-Rate, Wideband
speech coder

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Contents

1 Scope

The present document contains an electronic copy of the ANSI-C code for DSR Extended Advanced Front-end. The ANSI-C code is necessary for a bit exact implementation of DSR Extended Advanced Front-end.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

[1] ETSI standard ES 202 050 “Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithm”, Oct 2002

[2] ETSI Standard ES 202 212 “Distributed Speech Recognition; Extended Advanced Front-end Feature Extraction Algorithm; Compression Algorithm, Back-end Speech Reconstruction Algorithm”, Nov 2003

3 Definitions and abbreviations

3.1 Definitions

Definition of terms used in the present document, can be found in [1], [2]

3.2 Abbreviations

For the purpose of the present document, the following abbreviations apply:

ANSI	American National Standards Institute
I/O	Input/Output
RAM	Random Access Memory
ROM	Read Only Memory
AFE	Advanced Front-end
X-AFE	eXtended Advanced Front-end
DSR	Distributed Speech Recognition

4 C code structure

This clause gives an overview of the structure of the bit-exact C code and provides an overview of the contents and organization of the C code attached to this document.

The C code has been verified on the following systems:

- Sun Microsystems workstations and GNU gcc compiler
- IBM PC compatible computers with Linux operating system and GNU gcc compiler.

ANSI-C was selected as the programming language because portability was desirable.

4.1 Contents of the C source code

The distributed files with suffix "c" contain the source code and the files with suffix "h" are the header files.

Makefiles are provided for the platforms in which the C code has been verified (listed above).

4.2 Program execution

There are separate executables for the FrontEnd and Vector Quantization, with and without Extensions. The command line options are described below.

<> - indicates parameters for the given option for running the executable
() – indicates default parameter.

FrontEnd w/ Extension:
USAGE: bin/ExtAdvFrontEnd infile HTK_outfile pitch_outfile class_outfile [options]
OPTIONS:

-q	Quiet Mode (FALSE)
-F format	Input file format <*NIST,HTK,RAW*> (NIST)
-fs freq	Sampling frequency in kHz <*8,16*> (8)
-swap	Change input byte ordering (Native)
-noh	No HTK header to output file (FALSE)
-noc0	No c0 coefficient to output feature vector (FALSE)
-nologE	No logE component to output feature vector (FALSE)

-skip_header_bytes n - Skip header, first n bytes (Only for -F RAW)

-noh, -noc0, -nologE and –skip_header_bytes are not used and should not be changed.

FrontEnd w/o Extension:
USAGE: bin/AdvFrontEnd infile HTK_outfile [options]
OPTIONS: - Same as FrontEnd w/ Extension

Vector Quantization w/ Extension:
Usage: extcoder htk_file_in pitch_file_in class_file_in bitstream_file_out pitch_file_out txt_file_out -freq x -VAD/No_VAD

htk_file_in	Input mel-frequency cepstral coefficient file in HTK MFCC format.
pitch_file_in	Input pitch period file.
class_file_in	Input classification file.
bit_file_out	Output binary bitstream.
pitch_file_out	Output quantised pitch period file.
txt_file_out	Vector quantiser output in text format.
-freq x	Sampling frequency in kHz (8 or 16).
-VAD	Use voice activity detector data. Voice activity input file must have same name as htk_file, but extension .vad
-No_VAD	Do not incorporate voice activity detector information in output bitstream.

Vector Quantization w/o Extension:
Usage: coder htk_file_in bitstream_file_out txt_file_out -freq x -VAD/No_VAD

htk_file_in	Input mel-frequency cepstral coefficient file in HTK MFCC format.
bit_file_out	Binary output bitstream.
txt_file_out	Vector quantiser output in text format.
-freq x	Sampling frequency in kHz (8 or 16).
-VAD	Use voice activity detector data. Voice activity input file must have same name as htk_file, but extension .vad
-No_VAD	Do not incorporate voice activity detector information in output bitstream.

File extension descriptions as generated by the sample script:
.cep – Binary file containing cepstral features in HTK format. Output from the FrontEnd, input to the vector quantizer.
.pitch – Binary file containing pitch information. Output from the FrontEnd, input to the vector quantizer. Only used for Extension.
.class – Ascii file containing class information. Output from the FrontEnd, input to the vector quantizer. Only used for Extension.
.bs – Binary file containing the bitstream. Output from the vector quantizer.
.log – Log files from the different executables.

4.3 Code hierarchy

Tables 1 to 3 are call graphs that show the functions used for AFE (table 1), VQ (table 2), and Extension (table 3).

Each column represents a call level and each cell a function. The functions contain calls to the functions in rightwards neighboring cells. The time order in the call graphs is from the top downwards as the processing of a frame advances. All standard C functions: printf(), fwrite(), etc. have been omitted. Also, no basic operations (add(), L_add(), mac(), etc.) or double precision extended operations (e.g. L_Extract()) appear in the graphs.

The basic operations are not counted as extending the depth, therefore the deepest level in this software is level 7.

Table 1: AFE call structure

main()				
	AdvProcessInit_B()			
		DoNoiseSupInit_B()		
		DoWaveProcInit_B()		
		DoCompCepsInit_B()		
		DoPostProcInit_B()		
		DoVADInit_F()		
		Do16kProcInit_B()		
			QMF_FIR_Init_B()	
				fir_initialization_B()
				DP_HP_filters_B()
		BufIn32Alloc()		
			BufIn32Alloc()	
	AdvProcessAlloc_B()			
		DoNoiseSupAlloc_B()		
		DoWaveProcAlloc_B()		
		DoCompCepsAlloc_B()		
		DoPostProcAlloc_B()		
		DoVADAlloc_F()		
		Do16kProcAlloc_B()		
	FlushAdvProcess_B()			
		DoVADFlush_F()		
		CvFeatInt2Float()		
	AdvProcessDelete_B()			
		DoNoiseSupDelete_B()		
		DoWaveProcDelete_B()		
		DoCompCepsDelete_B()		
		DoPostProcDelete_B()		
		DoVADDelete_B()		
	DoAdvProcess_B()			
		Do16kProcessing_B()		
		DoNoiseSup_B()		
			Get16k_p_bufferData16k_B()	
			Get16k_bufData16kSize_B()	
			Get16k_p_BandsForCoding16k_B()	
			Get16k_p_CodeForBands16k_B()	
			Get16k_dataHP_B()	
			VAD_F()	
			DoSigWindowing16_F1()	
			DoSigWindowing16_F2()	
			ff4NBFix32_B()	
			FFTtoPSD_F()	
			Get16k_BFC_dec_B()	
			GetBandsForCoding16k_B()	
			PSDMean_F()	
			NoiseEstimation_F1()	
			NoiseEstimation_F2()	
			FilterCalc_F()	
			SpeechQVar()	
			FilterBank16()	
			SpeechQSpec()	
			SpeechQMel()	
			DoGainFact_F1()	
			DoGainFact_F2()	
			DoMelIDCT_F16()	
			ApplyWF()	
			Get16k_dec1()	
			Get16k_dec2()	
			Get16k_dec3()	
			DoSigWindowing16_F3()	
			ff4NBFix32_B()	
			FFTtoPSD_F()	
			DoMelFB_B()	
			CodeBands16k_B()	
			DoSpecSub16k_B()	
			DCOffsetFil_F()	
			Get16k_hpBandsSize_B()	
			Get16k_p_hpBands_B()	
			Get16k_p_bufferCodeForBands16k_B()	
			Get16k_p_CodeForBands16k_B()	
			Get16k_p_bufferCodeWeights_B()	
			Get16k_p_codeWeights_B()	
			Set16k_hpBands_dec_B()	
		DoWaveProc_B()		
			TeagerEng()	
			GetTeagerFilter()	
				GetMaximaPositions()
		DoCompCeps_B()		
			CepsCompute()	
				Get16k_p_bufferCodeWeights_B()
				Get16k_p_bufferCodeForBands16k_B()
				PreEmphHamm()
				ff4NB16_B()
				GetBandsForDecoding16k_B()
				DecodeBands16k_B()
				FilterBank()
				Get16k_hpBands_dec_B()
				Get16k_p_hpBands_B()
				MergeSSandCoded_B()
				CorrectEnergy_B()
				CosInv16Khz()
				cosInv() (only for 8kHz)
		DoPostProc_B()		

DoVADProc_F()	
	focalpoint()

Table 2: VQ call structure

main()			
	quantize_and_print()		
		get_best_dataframe()	
			best_centroid()
		quant_pitch_abs()	
		get_class_bit()	
		quant_pitch_diff()	
		get_class_bit()	
		mfcc_crc_encode()	
		pc_crc_encode()	

Table 3: Extension call structure

main()							
	RVC_ConstructPitchRom_be()						
	RVC_ConstructPitchMeter_be()						
		Allocate_InterpolatedDft_be()					
		RVC_ResetPitchMeter_be()					
	RVC_DestructPitchRom_be()						
	RVC_DestructPitchMeter_be()						
		Deallocate_InterpolatedDft_be()					
	DoAdvProcess_B()						
		DoPitchExtract()					
			FilterBank()				
			dsr_afe_vad()				
				get_vm()			
					fnLog2()		
			IsLowBandNoise()				
			get_zcm()				
			pre_process()				
				iir_d()			
				iir_s()			
			RVC_MeasurePitch_be()				
				ClearPitch_be()			
				DirichletInterpolation_be()			
				IsLowLevelInput_be()			
				Finalize_be()			
					IsContinuousPitch_be()		
						Mpy_lw_sw()	
				Mpy_lw_sw()			
				PrepareSpectralPeaks_be()			
					CalcSpectrum_be()		
						Mpy_lw_sw()	
						Mpy_lw_sw_Add()	
					FindPeaks_be()		
					Prelim_ScaleDownAmpsOfHighFreqPeaks_be()		
					qsort_be()*		
						swap()	
					CompareIpointAmp_be()		
					RefineSpectralPeaks_be()		
						sqrt_l_fix()	
					Final_ScaleDownAmpsOfHighFreqPeaks_be()		
					Mpy_lw_sw()		
				FindPitchCandidates_be()			
					NormalizeAmplitudes_be()		
					CalcUtilityFunction_be()		
						CreatePieceWiseConstantFunction_be()	
							L_Extract()
							Mpy_32_16()
						qsort_be()*	
							swap()
						Compare_ARRAY_OF_XPOINTS_be()	
						LinkArrayOfPoints_be()	

			AddSortedArrayOfPoints_be()	
				LinkArrayOfPoints_be()
			ConvertLinkedListOfDiffPointsToUtilFunc_be()	
		FindDominantLocalMaximaInUtilityFunction_be()		
			Mpy_lw_sw()	
		UtilityFunctionAtGivenPitchFreq_be()		
		qsort_be()*		
			swap()	
		ComparePitchFreqAscending_be()		
		SelectTopPitchCandidates_be()		
			Mpy_lw_sw()	
		compute_pcorr_be()		
			interpolate_be()	
				Mpy_lw_sw()
				Mpy_lw_lw()
				sqrt_l_fix()
			find_most_energetic_window_be()	
			accumulate_be()	
			find_most_energetic_window2_be()	
			Mpy_lw_sw()	
	SelectFinalPitch_be()			
		qsort_be()*		
			swap()	
		ComparePitchFreqDescending_be()		
		ClearPitch_be()		
		GOOD_ENOUGH_be()		
		CLOSELY_LOCATED_be()		
			Mpy_lw_sw()	
		BETTER_be()		
		IsContinuousPitch_be()		
			Mpy_lw_sw()	
	CalculateDoubleWindowDft_be()			
classify_frame()				

`* qsort_be() is a recursive function`

4.4 Variables, constants and tables

The data types of variables and tables used in the fixed point implementation are signed integers in 2's complement representation, defined by:

- **Word16** 16 bit variable;
- **Word32** 32 bit variable.

4.4.1 Description of constants used in the C-code

Table 5a: Global constants for AFE

Constant	Value	Description
NS_SPEC_ORDER_16K	64	Noise suppression Array length
NS_HANGOVER_16K	15	Noise suppression hangover count
NS_MIN_SPEECH_FRAME_HANGOVER_16K	4	Noise suppression minmum speech frame hangover count
NS_ANALYSIS_WINDOW_16K	80	Noise suppression analysis window
PERC_CODED	0.7	lambda merge (empirically set constant)
LAMBDA_NSE16k	0.99	Noise estimation Lambda
NS_NB_FRAME_THRESHOLD_NSE	100	Noise suppression number of frame threshold used for NSE
LENGTH_QMF	118	QMF filter length
f24	1	multiplier for QMF filter coefficients
SHFF_H	8	shift to get higher value
L_H	16	shift to get lower value
HP16k_MEL_USED	3	Higher frequnecy band Mel used
NB_LP_BANDS_CODING	3	Lower frequency band used in coding
NE16k_FRAMES_THRESH	100	Noise estimation frames threshold
NB_TOPOSTPROC	12	Number of coefficients to postprocess
CEP_FRAME_LENGTH	200	Frame length for cepstral coefficients
CEP_NB_COEF	13	Number of cepstral coefficients (including c0)
CEP_NB_CHANNELS	23	Number of filters used for cepstral coefficients
CEP_FFT_LENGTH	256	FFT length for cepstral coefficients
FRAME_BUF_SIZE	241	Denoised Output buffer size
FRAME_SHIFT	80	WaveProcessing input frame shift
FRAME_LENGTH	200	WaveProcessing frame size
NS_SPEC_ORDER	65	Noise suppression array length (8khz)
NS_BUFFER_SIZE	180	Noise suppression past frame size
NS_FRAME_SHIFT	80	Noise suppression input frame shift
NS_HALF_FILTER_LENGTH	8	Noise suppression filter half size
NS_NB_FRAME_THRESHOLD_LTE	10	Noise suppression long term energy forgetting factor threshold (in frames)
NS_NB_FRAME_THRESHOLD_NSE	100	Noise suppression spectrum estimate forgetting factor threshold (in frames)
NS_MIN_FRAME	10	Number of frame threshold to update average energy for Nosie suppression VAD
NS_FFT_LENGTH	256	FFT length for noise suppression
WF_MEL_ORDER	25	Noise suppression Wiener filter order
SHFT_NOISE	14	shift applied to noise spectrum estimate
SHFT_FACT_MUL	14	shift applied to gain coefficient (nosie suppression gain factoriization)
IDCT_ORDER	25	Noise suppression idct order
NS_BETA	0.98	Noiseless signal suppression factor
NS_RSB_MIN	0.079432823	Minimum a priori SNR
NS_LAMBDA_NSE	0.99	Forgetting factor for noise spectrum estimate
NS_LOG_SPEC_FLOOR	-10.0	average energy minimum threshold
NS_SNR_THRESHOLD_VAD	15	SNR threshold for noise suppression VAD
NS_SNR_THRESHOLD_UPD_LTE	20	Long term energy update threshold for noise suppression VAD
NS_ENERGY_FLOOR	80	Energy Minimum threshold for noise suppression VAD
MaxPos	10	Maximum number of maxima in waveprocessing
WP_EPS	0.2	weithing value added or substracted for waveprocessing

Table 5b: Global constants for VQ

Constant	Value	Description
MIN_PERIOD	1245184	Minimum pitch period allowed
MAX_PERIOD	9175040	Maximum pitch period allowed
NUM_MULTI_LEVELS_1	26	number of levels in pitch quantization
NUM_MULTI_LEVELS_2	24	number of levels in pitch quantization
UNVOICED_CODE	0	init value for Qpindex

Table 5c: Global constants for Extension

Constant	Value	Description
HISTORY_LEN	100	History length - past samples for pitch extraction
DOWN_SAMP_FACTOR	4	Down-sampling factor - used in computing correlation
NO_OF_DFT_POINTS	128	Number of DFT points
BREAK_POINT	12	Break point - marks the end of low frequency band
LBN_HIST_WEIGHT	32440	Low band noise history weight
LBN_CURR_WEIGHT	328	Low band noise current weight (32768 - LBN_HIST_WEIGHT)
LBN_MAX_THR	124518	Low band noise maximum threshold
LBN_LOW_ENR_LEVEL_MANT	32000	Low band noise low energy level mantissa
LBN_LOW_ENR_LEVEL_SHFT	22	Low band noise low energy level shift
RVC_OK	0	Return code for success
RVC_ERR	-1	Return code for unspecified error
RVC_ERR_NOT_ENOUGH_MEMORY	-2	Return code for not enough memory
RVC_ERR_ILLEGAL_ARGUMENT	-3	Return code for an illegal input / output argument
RVC_ERR_IO_FAILED	-4	Return code for failed input / output to a file
RVC_ERR_BAD_FILE_FORMAT	-5	Return code for a bad file header
RVC_ERR_NOT_INITIALIZED	-6	Return code for failure due to improper initialization
RVC_ERR_ILLEGAL_USAGE	-7	Return code for illegal usage of a function
RVC_ERR_NOT_ENOUGH_SAMPLES	-8	Return code for insufficient number of samples
RVC_ERR_NOT_IMPLEMENTED	-9	Return code for an unimplemented function
RVC_ERR_FAIL_OPEN_FILE	-10	Return code for failure to open a file

UB_ENRG_FRAC	59	Upper band energy fraction
ZCM_THLD	87	Zero crossing measure threshold
SQRT_ONE_HALF	0x5A82	Square root of 0.5 (0.707)
FRAME_LEN_DS	50	Frame length downsampled (200/4)
FRAME_LEN_DS_BY_2	25	Frame length downsampled divided by 2
HISTORY_LEN_DS	25	History length downsampled (100/4)
WINDOW_LENGTH	18	Window length used in computing correlation
INV_WINDOW_LENGTH	1820	Inverse of window length (1/18 = 0.05556)
NUM_CHAN	23	Number of channels or Mel-frequency bands
MIN_CH_ENRG_MANTISSA	20000	Minimum channel energy mantissa
MIN_CH_ENRG_SHIFT	25	Minimum channel energy shift
INIT_SIG_ENRG_MANTISSA	30518	Initial signal energy mantissa
INIT_SIG_ENRG_SHIFT	8	Initial signal energy shift
CE_SM_FAC	18022	Channel energy smoothing factor
CE_SM_FAC_COMPL	14746	Channel energy smoothing factor complement
CNE_SM_FAC	3277	Channel noise energy smoothing factor
CNE_SM_FAC_COMPL	29491	Channel noise energy smoothing factor complement
LO_GAMMA	22938	Low gamma value
LO_GAMMA_COMPL	9830	Low gamma value complement
HI_GAMMA	29491	High gamma value
HI_GAMMA_COMPL	3277	High gamma value complement
LO_BETA	31130	Low beta value
HI_BETA	32702	High beta value
INIT_FRAMES	10	Initial number of frames (considered to be noise frames)
SINE_START_CHAN	4	Sine start channel (for sine wave detection)
PEAK_TO_AVE_THLD	10	Peak to average threshold
DEV_THLD	1523942	Deviation threshold
HYSTER_CNT_THLD	9	Hysteresis count threshold
F_UPDATE_CNT_THLD	500	Forced update count threshold
NON_SPEECH_THLD	32	Non-speech threshold
FIX_34	24576	(short) (32768.0 * 3.0/4.0)
FIX_18	4096	(short) (32768.0 * 1.0/8.0)
FIX_INVSQRT2	-23170	1 / sqrt(2)
swTHIRD_REF_BANDWIDTH	85	One third of the reference bandwidth
swTWO_THIRDS_REF_BANDWIDTH	171	Two thirds of the reference bandwidth
MIN_ENERGY_MANTISSA	25600	Minimum energy mantissa
MIN_ENERGY_SHIFT	18	Minimum energy shift
swREF_SAMPLE_RATE_Q0	0x1F40	Reference sampling rate in Q0 format
swCLOSE_FACTOR_Q14	0x4CCD	Closeness factor in Q14 format
swFD_SCORE_THLD1_Q15	0x63D7	Frequency domain score threshold 1 in Q15 format
swFD_SCORE_THLD2_Q15	0x570A	Frequency domain score threshold 2 in Q15 format
swCORR_THLD_Q15	0x651F	Correlation threshold in Q15 format
swSUM_THLD_Q14	0x6667	Sum threshold in Q14 format
lwCRIT0_OFFSET_Q15	0x0000170A	Offset for finding a better pitch candidate in Q15 format
swCANDCORR_THLD1_Q15	0x799A	Pitch candidate correlation threshold 1 in Q15 format
swCANDCORR_THLD2_Q15	0x599A	Pitch candidate correlation threshold 2 in Q15 format
swCANDCORR_THLD3_Q15	0x6CCD	Pitch candidate correlation threshold 3 in Q15 format
swCANDAMP_THLD3_Q15	0x68F6	Pitch candidate amplitude threshold 3 in Q15 format
swSTARTFREQ_COEFF	0x553F	Start frequency coefficient (for candidate search)
swENDFREQ_COEFF	0x4666	End frequency coefficient (for candidate search)
DIRICHLET_KERNEL_SPAN	8	Direchlet kernal span (for interpolation)
REF_SAMPLE_RATE	8000	Reference sampling rate
REF_BANDWIDTH	4000	Reference bandwidth
lwTHIRD_REF_BANDWIDTH	87381333	One third of the reference bandwidth
lwTWO_THIRDS_REF_BANDWIDTH	174762667	Two thirds of the reference bandwidth
swCENTER_WEIGHT	0x5000	Center weight
swSIDE_WEIGHT	0x1800	Side weight
swAMP_SCALE_DOWN1	0x5333	Amplitude scale down factor 1
swAMP_SCALE_DOWN2	0x399A	Amplitude scale down factor 2
swAMP_SCALE_DOWN2b	0x7333	Amplitude scale down factor 2b
swUDIST1	-4160	Utility function distance 1
swUDIST2	-6400	Utility function distance 2
swUSTEP	-16384	Utility function step
swFREQ_MARGIN1	0x4AE1	Frequency margin 1
swAMP_MARGIN1	0x07AE	Amplitude margin 1
swAMP_MARGIN2	0x07AE	Amplitude margin 2
MIN_STABLE_FRAMES	6	Minimum number of stable frames
MAX_TRACK_GAP_FRAMES	2	Maximum pitch track gap frames
swSTABLE_FREQ_UPPER_MARGIN	0x4E14	Stable frequency upper margin
swSTABLE_FREQ_LOWER_MARGIN	0x68EB	Stable frequency lower margin
UNVOICED	0	Pitch frequency of an unvoiced frame
lwMAX_PITCH_FREQ	0x01A40000L	Maximum pitch frequency
lwMIN_PITCH_FREQ	0x00340000L	Minimum pitch frequency
MAX_PITCH_FREQ	420	Maximum pitch frequency in Hz
MIN_PITCH_FREQ	52	Minimum pitch frequency in Hz
HIGHPASS_CUTOFF_FREQ	300	Highpass cut-off frequency in Hz
NO_OF_FRACS	77	Number of fractions in the frations table
lwSHORT_WIN_START_FREQ	0x00C80000L	Short window start frequency
lwSHORT_WIN_END_FREQ	0x01A40000	Short window end frequency
lwSINGLE_WIN_START_FREQ	0x00640000L	Single window start frequency
lwSINGLE_WIN_END_FREQ	0x00D20000L	Single window end frequency
lwDOUBLE_WIN_START_FREQ	0x00340000	Double window start frequency
lwDOUBLE_WIN_END_FREQ	0x00780000L	Double window end frequency
MAX_LOCAL_MAXIMA_ON_SPECTRUM	70	Maximum number of local maxima on the spectrum
MAX_PEAKS_FOR_SORT	30	Maximum number peaks for sorting
MAX_PEAKS_PRELIM	7	Maximum number of peaks (preliminary)
MIN_PEAKS	7	Minimum number of peaks
MAX_PEAKS_FINAL	20	Maximum number of peaks (final)
MAX_PRELIM_CANDS	4	Maximum number of preliminary candidates (pitch)
CREATE_PIECEWISE_FUNC_LOOP_LIM_SH	20	Create Piecewise function loop limit for short window
CREATE_PIECEWISE_FUNC_LOOP_LIM_SNG	30	Create Piecewise function loop limit for single window
CREATE_PIECEWISE_FUNC_LOOP_LIM_DBL	60	Create Piecewise function loop limit for double window
swSUM_FRACTION	0x799A	Sum fraction

swAMP_FRACTION	0x33F8	Amplitude fraction
MAX_BEST_CANDS	2	Maximum number of best candidates (pitch)
N_OF_BEST_CANDS_SHORT	2	Number of best candidates for short window
N_OF_BEST_CANDS_SINGLE	2	Number of best candidates for single window
N_OF_BEST_CANDS_DOUBLE	2	Number of best candidates for double window
N_OF_BEST_CANDS	6	Number of best candidates for all windows
SIZE_SCRATCH_DOPITCH	1090	Scratch memory size for DoPitch() function (This is the actual size required. The declared size in C simulation is 1632)
SIZE_SCRATCH_ADVPROCESS	825	Scratch memory size for DoAdvProcess() function (This is the actual size required. The declared size in C simulation is 1100)
RVC_PITCH_ROM_SIG	11031	Signature for RVC_PITCH_ROM structure
RVC_PITCH_METER_SIG	21053	Signature for RVC_PITCH_METER structure

4.4.2 Description of fixed tables used in the C-code

This section contains a listing of all fixed tables sorted by source file name and table name. All table data is declared as **Word16**.

Table 6a: Fixed tables for AFE

File	Table Name	Length	Description
16kHzProcessing_B.c	table_pow2	33	Table for square root
	LambdaNSEx2	100	Table used to compute first 100 LambdaNSE
	dp02_h	59	MSB of QMF filter coefficients
	dp02_l	43	LSB of QMF filter coefficients
PostProc_B.c	targetLMS16	12	Target for blind equalization
ComCeps_B.c	HalfHamming16	100	Hamming window coefficients
	CosMatrix16	144	Inverse cosinus coefficients at 8Khz (not used at 16khz)
	CosMatrix16_16khz	156	Inverse cosinus coefficients at 16Khz
	pondMelFilter	309	Mel bank coefficients
ff4nrFix16_B.c	tabSin	64	Sine table
	tabCos	64	Cosine table
ff4nrFix32_B.c	tabSin	64	Sine table
	tabCos	64	Cosine table
MathFunc.c	tbInt0	48	Coefficients for computation of square root
ExtNoiseSup_B.c	lambda_1divX	20	Computation of 1/N
	Hann_sh32_hi	100	MSB of hanning window coefficients (32 bits)
	Hann_sh32_lo	100	LSB of hanning window coefficients (32 bits)
	Hann_sh24_hi	100	MSB of hanning window coefficients (24 bits)
	Hann_sh24_lo	100	LSB of hanning window coefficients (24 bits)
	pondMelFilterNoise	157	Mel-frequency scale coefficients (applied to the Wiener filter)
	idctMel16	234	Mel-warped inverse DCT coefficients
	pondMelFilter16k	134	Filter bank coefficients at 16Khz
	M1_LamdaLTE	8	Computation of 1/N
	M1_LambdaNSEx2	100	Computation of 2/N
	M1_LamdaNSE	9	Computation of 1/N
	mInvLambda16	10	Comutation od 2/N

Table 6b: Fixed tables for VQ

File	Table Name	Length	Description
coder_VAD.c	quantizer16kHz_0_1	128	vq table
	quantizer16kHz_2_3	128	vq table
	quantizer16kHz_4_5	128	vq table
	quantizer16kHz_6_7	128	vq table
	quantizer16kHz_8_9	128	vq table
	quantizer16kHz_10_11	64	vq table
	quantizer16kHz_12_13	512	vq table
	quantizer8kHz_0_1	128	vq table
	quantizer8kHz_2_3	128	vq table
	quantizer8kHz_4_5	128	vq table
	quantizer8kHz_6_7	128	vq table
	quantizer8kHz_8_9	128	vq table
	quantizer8kHz_10_11	64	vq table
	quantizer8kHz_12_13	512	vq table
	weight16kHz_c0_shift	1	vq weights
	weight16kHz_c0_norm	1	vq weights
	weight16kHz_logE	1	vq weights
	weight8kHz_c0_shift	1	vq weights
	weight8kHz_c0_norm	1	vq weights
	weight8kHz_logE	1	vq weights
	plwQuantLevels[127]	127*2	vq tables for pitch/class quantization
	ppplwQuantSections[8][3]	24*2	vq tables for pitch/class quantization
	plwQuantLevels[31]	31*2	vq tables for pitch/class quantization
	pplwQuantSections[4][3]	12*2	vq tables for pitch/class quantization
	pswRatioThld_1[4][6]	24	vq tables for pitch/class quantization
	piMultiLevelIndex[4]	4	vq tables for pitch/class quantization
	pswRatioThld_2[4][8]	32	vq tables for pitch/class quantization
	piMultiLevelIndex_2[4]	4	vq tables for pitch/class quantization
	swAlpha1	1	pitch/class constants
	swAlpha2	1	pitch/class constants

Table 6c: Fixed Tables for Extension

File	Table name	Length	Description
ExtNoiseSup_B.c	pswPePower	129	Coefficients to compute the pre-emphasis power spectrum
preProc_B.c	pswHpfCoef	15	High pass filter coefficients
preProc_B.c	pswLpfCoef	15	Low pass filter coefficients
preProc_B.c	pswLfeCoef	3	Low frequency emphasis filter coefficients
dsrAfeVad_B.c	piBurstConst	20	Burst length constants for different SNR's
dsrAfeVad_B.c	piHangConst	20	Hang length constants for different SNR's
dsrAfeVad_B.c	piVADThld	20	VAD voice metric thresholds for different SNR's
dsrAfeVad_B.c	piVMTable	90	Voice metric table as a function of SNR index
dsrAfeVad_B.c	piSigThld	20	Signal threshold table as a function of SNR
dsrAfeVad_B.c	piUpdateThld	20	Update threshold table as a function of SNR
dsrAfeVad_B.c	pswShapeTable	23	Spectral shape correction table
fix_mathlib.c	coeff_sqrt5_58	5	Coefficients for computation of square root
fix_mathlib.c	coeff_sqrt5_78	5	Coefficients for computation of square root
rvc_pitch_init_B.h	ROM_astFrac	312	Fractions table

rvc_pitch_init_B.h	ROM_pstWindowshiftTable	514	Complex exponents table for time shifting in frequency domain
rvc_pitch_init_B.h	ROM_aswDirichletImag	8	Imaginary part of the Dirichlet kernel

4.4.3 Static variables used in the C-code:

In this section two tables that specify the static variables for the AFE, VQ, and Extension respectively are shown.

Table 7a: AFE static variables

Struct Name	Variable	Type[Length]	Description
QMF_FIR			
	lengthQMF	Word32	QMF Filter length
	*dp_l	Word16	QMF filter low frequency Coeff
	*dp_h	Word16	QMF filter high frequency Coeff
	*T	Word16	Temporary QMF filter buffer
	T_dec	Word16	Multiplier for T
DataFor16kProc_B			
	FrameLength	Word32	Input Frame length
	FrameShift	Word32	Shift value for the frame
	numFramesInBuffer	Word32	Number of frames in buffer
	SamplingFrequency	Word32	Sampling frequency (8/16)
	Do16kHzProc	BOOLEAN	Flag to enable 16kHz processing
	*hpBands_B	Word32	Buffer for HP bands
	hpBandsSize	Word32	hpBands_B buffer size
	CodeForBands16k_B	Word32[9]	HP coding buffer
	bufferCodeForBands16k_B	Word32[27]	buffer used for HP coding
	codeWeights_B	Word16[3]	code Weights buffer
	bufferCodeWeights_B	Word16[9]	buffer used for code Weights
	* pQMF_Fir	QMF_FIR	Pointer to QMF_FIR structure
	*bufferData16k_B	Word32	temporary buffer to carry QMF LP data
	bufData16kSize	Word32	16k data buffer size
	*FirstWindow16k	MelFB_Window	pointer to MelFB_Window structure
	noiseSE16k_B	Word32[3]	noise spectrul energy variable
	noise_dec	Word16	Multiplier for noiseSE16k_B
	BandsForCoding16k_B	Word32[9]	buffer for storing Bands for Coding
	vadCounter16k	Word32	vad flag counter
	vad16k	Word32	vad flag
	nbSpeechFrames16k	Word32	number of speech frames counter
	hangOver16k	Word32	hang over used for VAD
	meanEn16k	Word32	mean Energy variable
	nb_frame_threshold_nse	Word32	threshold NSE for frame
	lambda_nse	Word16	lambda NSE variable
	*dataHP_B	Word32	buffer stores QMF HP value
	dec_16k	Word16[5]	Multiplier for dataHP_B buffer
	BFC_dec	Word16[1]	Multiplier for computing bands for coding
	fb16k_dec	Word16[3]	Buffer is used to store multiplier for current and pervious two frames
PostProcStructX			
	weightLMS	Word32[12]	Current LMS weight
CompCepsStructX			
	FFTLength	Word32	FFT size
	Do16khzProc	Word16	Flag to enable 16kHz processing
	*pData16k	Word32	Pointer to data for 16Khz processing
WaveProcStructX			
	*TeagerFilter16	Word32	Pointer to teager filter
	*TeagerWindow32	Word32	Pointer to teager window
	TeagerOnset	Word32	Unused
	FrameLength	Word32	Input frame length
ns_var_F			
	SampFreq	Word16	Sampling frequency (8/16)
	Do16khzProc	Word16	Flag to enable 16kHz processing
	buffers.nbFramesInFirstStage	Word32	number of frames in first stage
	buffers.nbFramesInFirstStage	Word32	number of frames in second stage
	buffers. nbFramesOutSecondStage	Word32	number of frames out og second stage
	buffers. FirstStageIn16Buffer	Word16[180]	First stage buffer
	buffers.SecondStageInBuffer32	Word32[180]	Second stage buffer
	buffers. SecondDecalSig	Word16[4]	Shift factor for each sub-frame of second stage buffer
	prevSamples32.lastSampleIn32	Word32	Last input sample of DC offset compensation
	prevSamples32.lastDCOut32	Word32	last output sample of DC offset compensation
	prevSamples32. oldShift	Word16	lprevious window shift factor of DC offset compensation
	spectrum.indexBuffer1	Word16	Where to enter new PSD for first stage, alternatively 0 and 1
	spectrum.indexBuffer2	Word16	Where to enter new PSD for second stage, alternatively 0 and 1
	spectrum.noiseSE1_32	Word32[65]	Noise spectrum estimate for first stage
	spectrum.noiseSE1_dec	Word16[65]	Shift factor for Noise spectrum estimate (first sage)
	spectrum.noiseSE2_32	Word32[65]	Noise spectrum estimate for second stage
	spectrum.noiseSE2_dec	Word16[65]	Shift factor for Noise spectrum estimate (second sage)
	spectrum.PSDMeanAntBuffer1	Word32[65]	1st stage PSD Mean buffer for precedent frame
	spectrum.nSigSE1Ant_dec	Word16[65]	Shift factor for PSD Mean buffer for precedent frame (1rst stage)
	spectrum.PSDMeanAntBuffer2	Word32[65]	2nd stage PSD Mean bufferfor precedent frame
	spectrum.nSigSE2Ant_dec	Word16[65]	Shift factor for PSD Mean buffer for precedent frame (2nd stage)
	spectrum.denSigSE1_32	Word32[65]	1st stage PSD Mean buffer
	spectrum. nSigSE1Cur_dec	Word16[65]	Shift factor for PSD Mean buffer (1rst stage)
	spectrum. denSigSE2_32	Word32[65]	2nd stage PSD Mean buffer
	spectrum. nSigSE2Cur_dec	Word16[65]	Shift factor for PSD Mean buffer (2nd stage)
	vad_data_ns_F. nbFrame	Word16[2]	Nubmer of frames (for the 2 stages)
	vad_data_ns_F. flagVAD	Word16	Vad Flag (1 = SPEECH, 0 = NON SPEECH)
	vad_data_ns_F.hangOver	Word16	hangover
	vad_data_ns_F. nbSpeechFrames	Word16	Number of speech frames (used to set hangover)
	vad_data_ns_F.meanEn32	Word32	Mean energy for VAD
	vad_data_ca. flagVAD	Word16	Vad Flag (1 = SPEECH, 0 = NON SPEECH)
	vad_data_ca.hangOver	Word16	hangover
	vad_data_ca. nbSpeechFrames	Word16	Number of speech frames (used to set hangover)
	vad_data_ca.meanEn32	Word32	Mean energy for VAD
	vad_data_fd.MelMean	Word16	SpeechQMel (for frame dropping)
	vad_data_fd.VarMean	Word32	SpeechQVar (for frame dropping)

	vad_data_fd.AccTest	Word32	SpeechQSpec (for frame dropping)
	vad_data_fd.AccTest2	Word32	
	vad_data_fd.SpecMean	Word32	SpecMean (for frame dropping)
	vad_data_fd.MelValues	Word16[2]	SpeechQMel (for frame dropping)
	vad_data_fd.SpecValues	Word32	SpeechQSpec (for frame dropping)
	vad_data_fd.SpeechInVADQ	Word16	Flag (for frame dropping)
	vad_data_fd.SpeechInVADQ2	Word16	Flag (for frame dropping)
	gainFact.logDenEn1_32	Word32[3]	Denoise frame energy for gain factorization
	gainFact.lowSNRtrack32	Word32	Low SNR level for gain factorization
	gainFact. alfaGF16	Word16	Wiener filter gain factorization coefficient
VADStructX_F			
	Focus	Word16	Position of circular buffe
	HangOver	Word16	Hangover length
	FlushFocus	Word16	Position in circular buffer when emptying at end
	H_CountDown	Word16	Main hangover countdown
	V_CountDown	Word16	Short hangover countdown
	**OutBuffer	Word32	outBuffer pointer pointer
	*OutBuffer	Word32[7]	outBuffer pointer
	OutBuffer	Word16[7x15]	outBuffer

Table 7b: VQ static variables

Struct Name	Variable	Type [Length]	Description
coder_VAD.c	four_frames[27]	Word16[27]	Previous frames used to build multiframe
	plwQPHistory[3]	Word32[3]	History of Pitch
	IReliableFlag	Word16	Pitch reliability flag

Table 7c: Extension static variables

Struct Name	Variable	Type[Length]	Description
	iFirstFrameFlag	Word16	First frame flag
	pswUBSpeech	Word16[200]	Upper band speech
	pswDownSampledProcSpeech	Word16[75]	Down-sampled processed speech
	lwCritMax	Word32	Maximum power ratio
	iOldPitchPeriod	Word16	Old pitch period value
	iOldFrameNo	Word16	Old frame number
PCORR_STATE_be	s_be		
	lwX1_X1	Word32	X1*X1
	lwZ1_Z1	Word32	Z1*Z1
	lwZ2_Z2	Word32	Z2*Z2
	lwX1_Z1	Word32	X1*Z1
	lwX1_Z2	Word32	X1*Z2
	lwZ1_Z2	Word32	Z1*Z2
	swX1_Sum	Word16	Sum of X1
	swZ1_Sum	Word16	Sum of Z1
	swZ2_Sum	Word16	Sum of Z2
	iBurstConst	Word16	Burst constant
	iBurstCount	Word16	Burst count
	iHangConst	Word16	Hang constant
	iHangCount	Word16	Hang count
	iVADThld	Word16	VAD threshold
	iFrameCount	Word16	Frame count
	iFUpdateFlag	Word16	Forced update flag
	iHysterCount	Word16	Hysteresis count
	iLastUpdateCount	Word16	Last update count
	iSigThld	Word16	Signal threshold
	iUpdateCount	Word16	Update count
	iChanEnrgShift	Word16	Channel energy shift
	iChanNoiseEnrgShift	Word16	Channel noise energy shift
	pswChanEnrg	Word16[23]	Channel energy
	pswChanNoiseEnrg	Word16[23]	Channel noise energy
	swBeta	Word16	Beta value
	swSnr	Word16	SNR value
NormSw	pnsLogSpecEnrgLong		
	swMantissa	Word16[23]	Mantissa
	iShift	Word16[23]	Shift
	swC0	Word16	C0 value
	swC1	Word16	C1 value
	swC2	Word16	C2 value
	pswHpfXState	Word16[6]	High pass filter input state
	pswHpfYState	Word16[12]	High pass filter output state
	pswLpfXState	Word16[6]	Low pass filter input state
	pswLpfYState	Word16[12]	Low pass filter output state
	pswLfeXState	Word16	Low frequency emphasis filter input state
	pswLfeYState	Word16[2]	Low frequency emphasis filter output state

5 File formats

This section describes the file formats used by the AFE, VQ & Extension programs.

5.1 Speech file

Speech files read by the X-AFE and written by the Extension consist of 16-bit words. The byte order depends on the host architecture (e.g. MSByte first on SUN workstations, LSByte first on PCs etc)

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
02-2004					Version 1.0.0 provided for Information		

3GPP TSG-SA4#30 meeting
February 23-27, 2004, Malaga, Spain

Tdoc S4-040064

Source: Nokia
Title: Speech reconstruction assessment results
Agenda item: 7

1. Introduction

As part of the codec selection for the SES service, the capability of candidate codecs to reconstruct speech was assessed in subjective listening tests. This document presents the speech reconstruction assessment test results.

2. Test overview

The test was split into three experiments concerning the different use conditions. AMR 12.2 and 4.75 kbit/s modes and ES 202 050 (with extension) were tested in each experiment.

Exp. No.	Test type	Title
1	ACR test	AMR and ES 202 050 (with extension) in clean speech in clean and error prone channel (8 kHz sampling)
2	DCR test	AMR and ES 202 050 (with extension) in speech with background noise (babble) speech in clean and error prone channel (8 kHz sampling)
3	DCR test	AMR and ES 202 050 (with extension) in speech with background noise (car) speech in clean and error prone channel (8 kHz sampling)

Table 1. Speech reconstruction assessment experiments

2.1 Test environment

The tests were conducted in Nokia listening test facilities in a quiet environment; 30dBA Hoth Spectrum (as defined by ITU-T, Recommendation P.800 [2], Annex A, section A.1.1.2.2.1 Room Noise, with table A.1 and Figure A.1) measured at the head position of the subject.

2.2 Listeners

All the listeners were native Finnish speakers and naïve with the listening tests. Altogether 58 listeners conducted this test (24 per experiment). Some of the listeners conducted two experiments. In that case they first conducted experiment 1 and then either experiment 2 or 3.

2.3 Input source material

Source material was balanced Finnish sentences recorded according to ITU-T recommendation P.800 [2]. A Corpus of balanced sentences was produced according to [3] by the Department of Phonetics in University of Helsinki. An external high quality studio performed all the recordings.

The packet loss simulation was done by corrupting the codec bit streams with an error pattern file consisting of lost packet flags. The error pattern files with 1 and 3% packet error loss rates were the same to those used in the 3GPP speech recognition experiments.

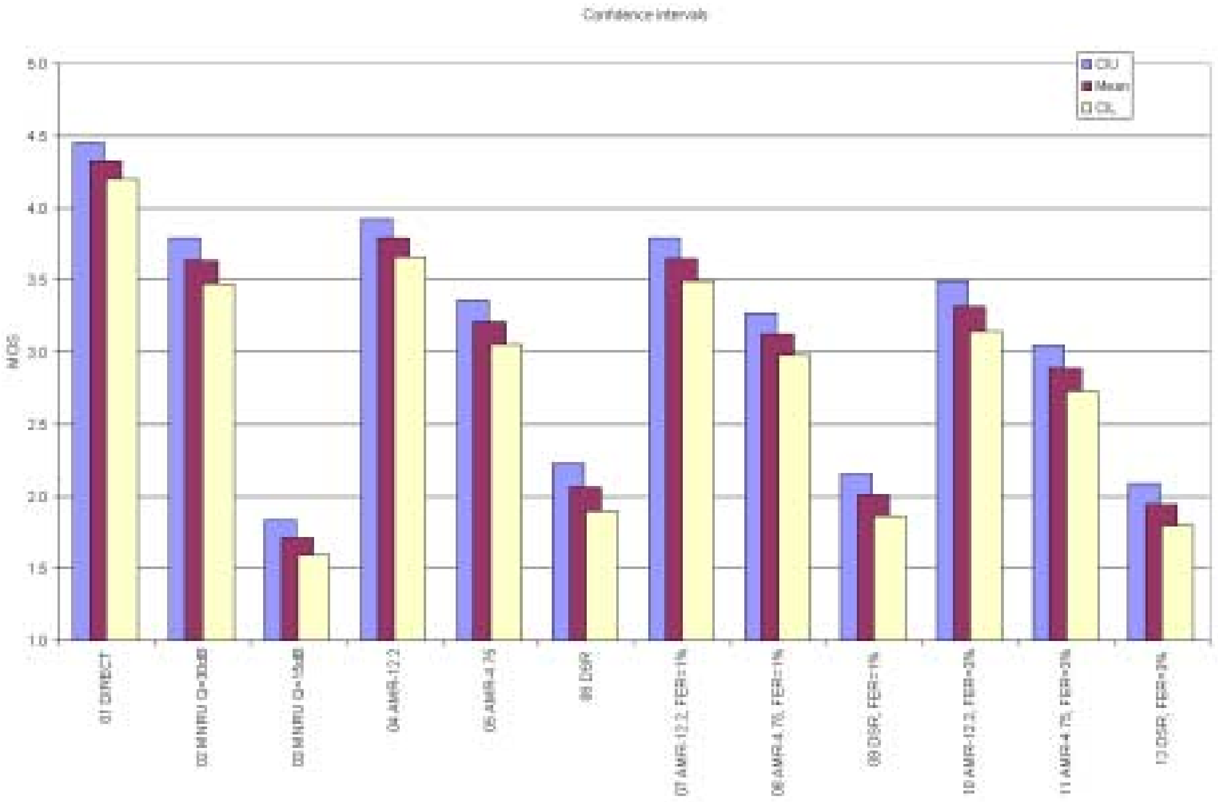
2.4 Processing

All the speech material was processed in Nokia according to test plan [1]. AMR speech coding was done also in Nokia. DSR coded material was processed by Motorola.

3. Test results

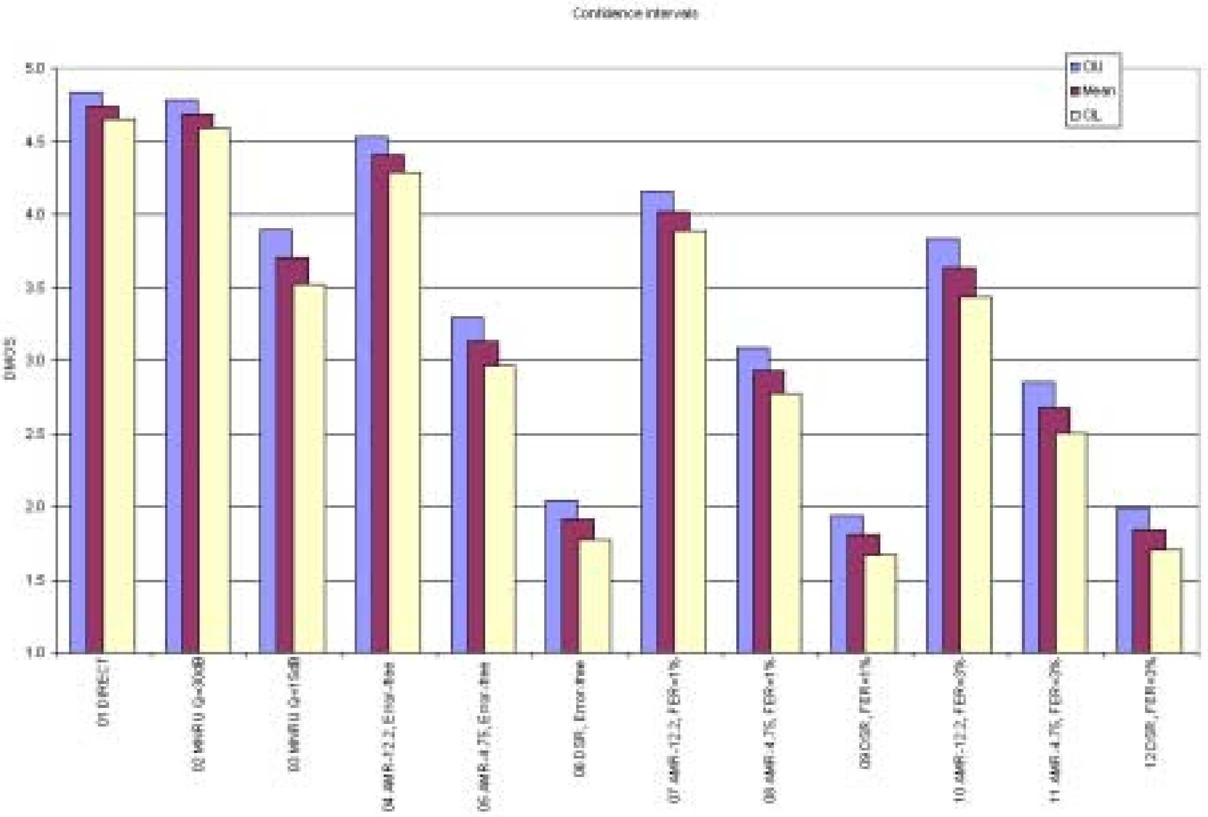
The test results are presented below in Figures 1, 2 and 3. The diagrams contain the average MOS (experiment 1) and DMOS (experiments 2 and 3) values as well as 95% confidence intervals. The diagrams contain the score for each condition including the reference MNRU items.

3.1 Experiment 1



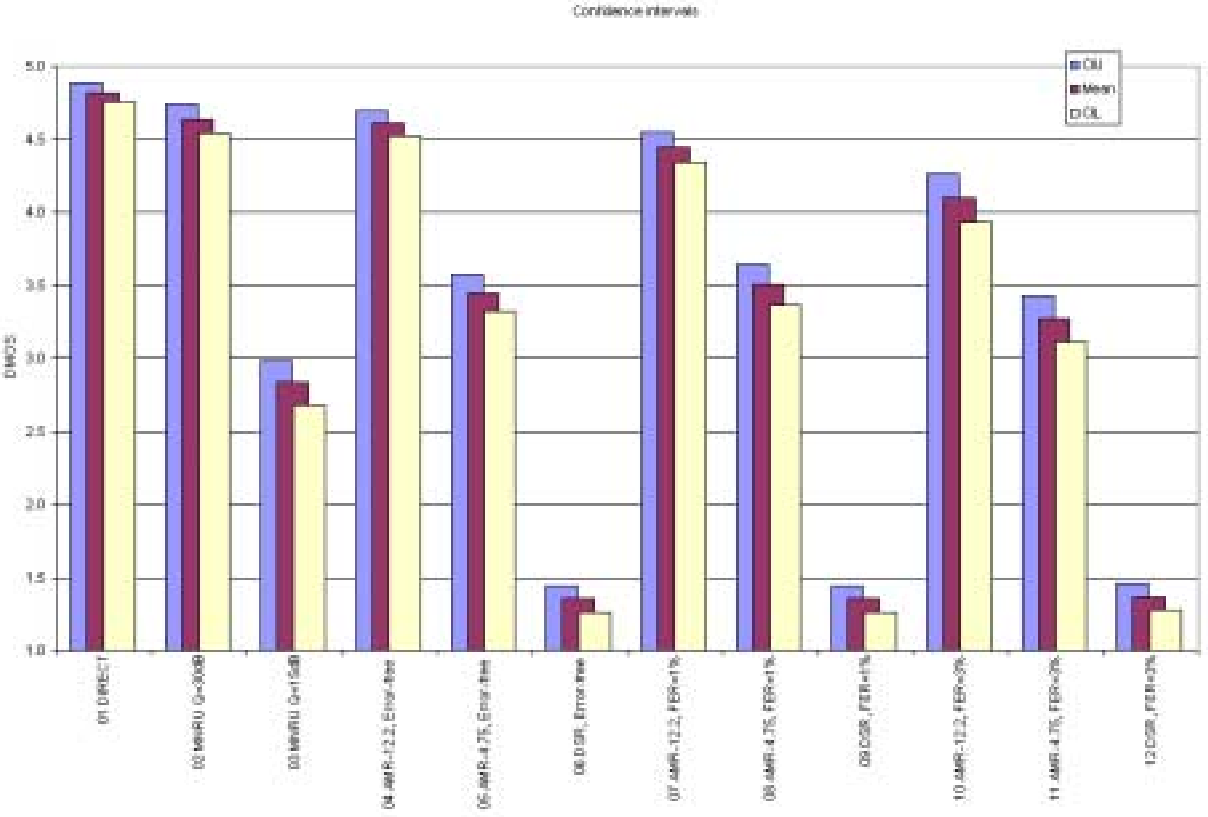
Picture 1. ACR test results in clean speech in clean and error prone channel

3.2 Experiment 2



Picture 2. DCR test results in babble background noise in clean and error prone channel

3.3 Experiment 3



Picture 3. DCR test results in car background noise in clean and error prone channel

4. Conclusion

The results obtained clearly indicate that only the AMR candidate codec is capable of producing high quality speech. Speech quality of the played messages is important for the user to conveniently interact with the speech recognition application and it thus impacts the user experience for the speech enabled service.

5. References

[1] S4-030539 “Assessment of the codec capability to reconstruct speech”

[2] ITU-T; Recommendation P.800; “Methods for Subjective Determination of Transmission Quality”

3GPP TSG-SA4#30 meeting **Tdoc S4-040069**
February 23-27, 2004, Malaga, Spain

Title: **SES candidate codec speech reconstruction quality evaluation**

Source: **Ericsson**

Document for: Information

Agenda Item: 7

1. Introduction

On mandate of SA speech quality evaluations have been carried out testing the speech reconstruction quality of the SES candidate codecs. This document presents for information to be considered in the SES codec selection results of quality evaluations carried out by RCDCT Laboratories[1] on behalf of Ericsson.

2. Experiments

RCDCT Laboratories performed three listening assessments in Chinese comparing the codec speech reconstruction quality of the candidate codec for SES, the AMR speech codec (3GPP TS 26.073) and DSR.

The evaluations were done in accordance with the test plan specified in [1]. The following three experiments were performed:

Exp. No.	Title
1	ACR test: AMR and DSR in clean speech in clean and error prone channel (8 kHz sampling)
2	DCR test: AMR and DSR in speech with background noise (babble) speech in clean and error prone channel (8 kHz sampling)
3	DCR test: AMR and DSR in speech with background noise (car) speech in clean and error prone channel (8 kHz sampling)

The experiments were carried out using a subset of the Chinese speech material available in the NTT Speech Database. Twenty-four distinct native speakers of the Chinese language performed as subjects for each of the three experiments, which were nominally balanced for gender. In total, 72 subjects were used. The raw data collected was used to derive Mean Opinion Scores and and standard deviation statistics for each experiment.

2.1. Source Material

2.1.1 Speech source material

The experiments were performed using a subset of the Chinese speech material available in the NTT Speech Database. Six sentence pairs from three male and three female Chinese-speaking talkers i.e. a total of 36 were selected.

[1] Contact: ShengHui Zhao
Research Center of Digital Communications Technology (RCDCT)
Tel: +86-10-6891-1841
E-mail: shzhao@bit.edu.cn

2.1.2 Background noise material

The background noise signals for experiment 2 and 3 were taken from the NTT noise database.

2.1.3 Channel error patterns

The same channel error patterns for 1% and 3% frame loss rate were used as in the speech recognition experiments. Alcatel provided the error patterns.

2.2. Processing

2.2.1 Preprocessing

The speech source material was MSIN filtered and level adjusted to an active speech level of -26 dBov. For the experiments with background noise, MSIN filtered background noise was adjusted to an RMS level of –36 dBov and then added to the speech files giving noisy speech with the required SNR of 10 dB. The pre-processing was done by Ericsson.

2.2.2 Main-processing

The processing for the AMR codec conditions was done using executables built from 3GPP TS 26.073. For the conditions with frame losses a frame loss device was used discarding codec frames depending on the contents of the error pattern file. This processing was done with concatenated speech files, according to the test plan.

The processing for the DSR codec was done by Motorola using concatenated speech files.

3. Listening Sessions

3.1 Listener groups, randomization and presentation order

For each experiment, the test subjects were divided in eight groups of three subjects and each group used its own material randomization and a unique random presentation order.

3.2 Listeners

Each of the three subjective assessments was carried out using 24 listeners (nominally balanced between male and female), divided into eight groups of three listeners each. In total, 72 different native speakers of Chinese performed as test subjects.

3.3 Lab setup

The processed speech material was presented to groups of listeners, seated at separate, visually screened listening stations contained within an acoustically conditioned sound room meeting the requirements recommended by ITU-T P.800. The room had a HOTH noise spectrum at 30 dBA level. The presentations were made monaurally.

4. Results

Figures 1 to 3 display the MOS, respectively, DMOS scores obtained in the 3 experiments. The error bars show the 95% confidence intervals for the scores obtained in a statistical analysis of the results.

As can be concluded from the results of experiment 1, the speech quality of the AMR codec is scored in a range from good to fair, depending on the AMR codec mode and the error condition. The speech reconstruction quality of the DSR codec is scored from slightly better than poor to poor, depending on the error condition.

The degradation scores obtained for the AMR codec in the experiments with background noise range from almost not perceivable to slightly annoying, depending on the AMR mode and the error condition. The speech quality of the DSR codec is rated between annoying and very annoying.

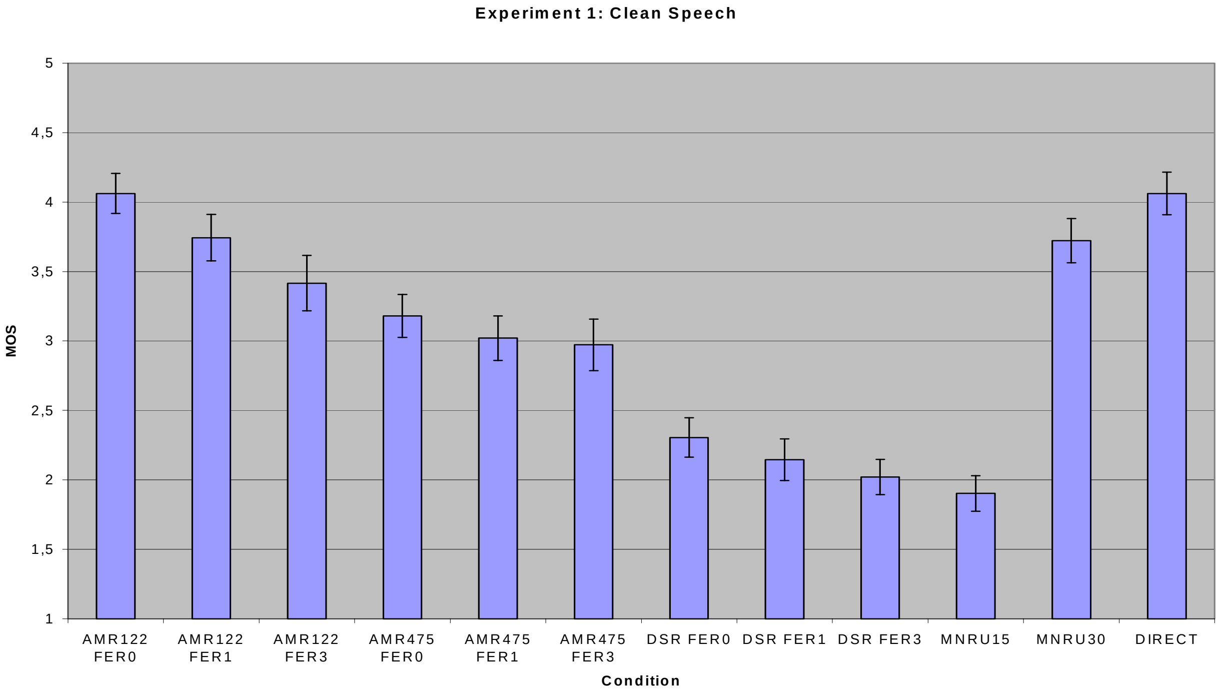


Figure 1: Results of experiment 1 - Clean speech performance under clean and degraded channel conditions

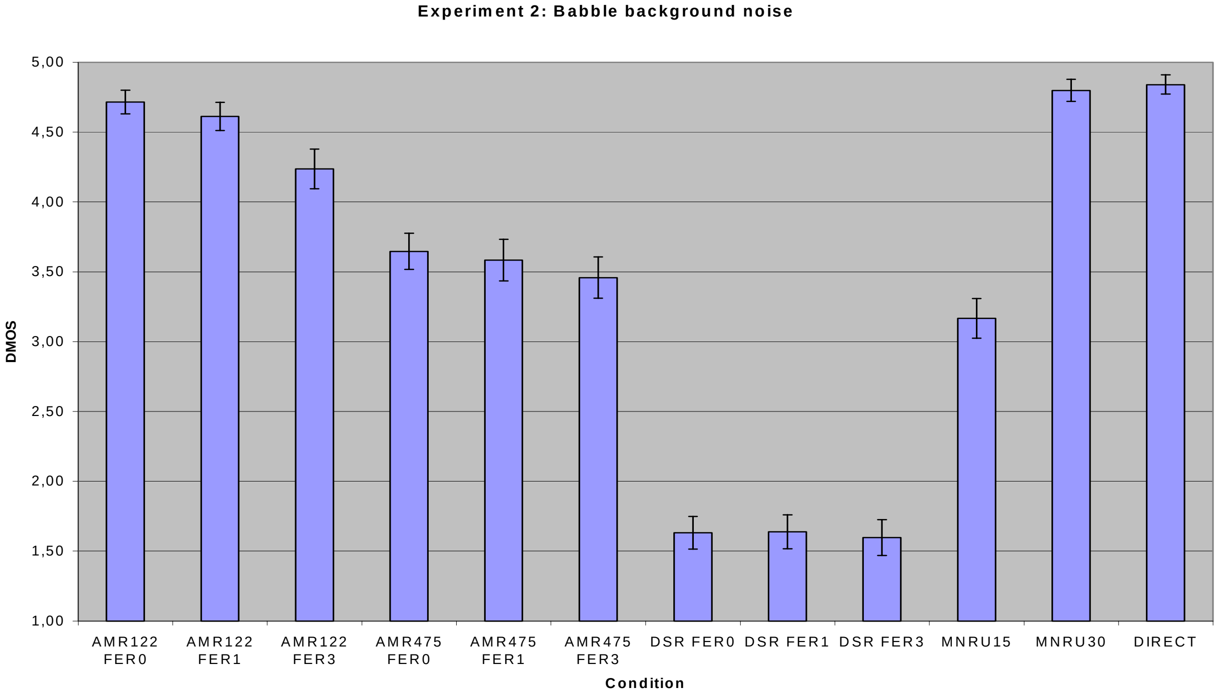


Figure 2: Results of experiment 2 – Noisy speech performance (babble) under clean and degraded channel conditions

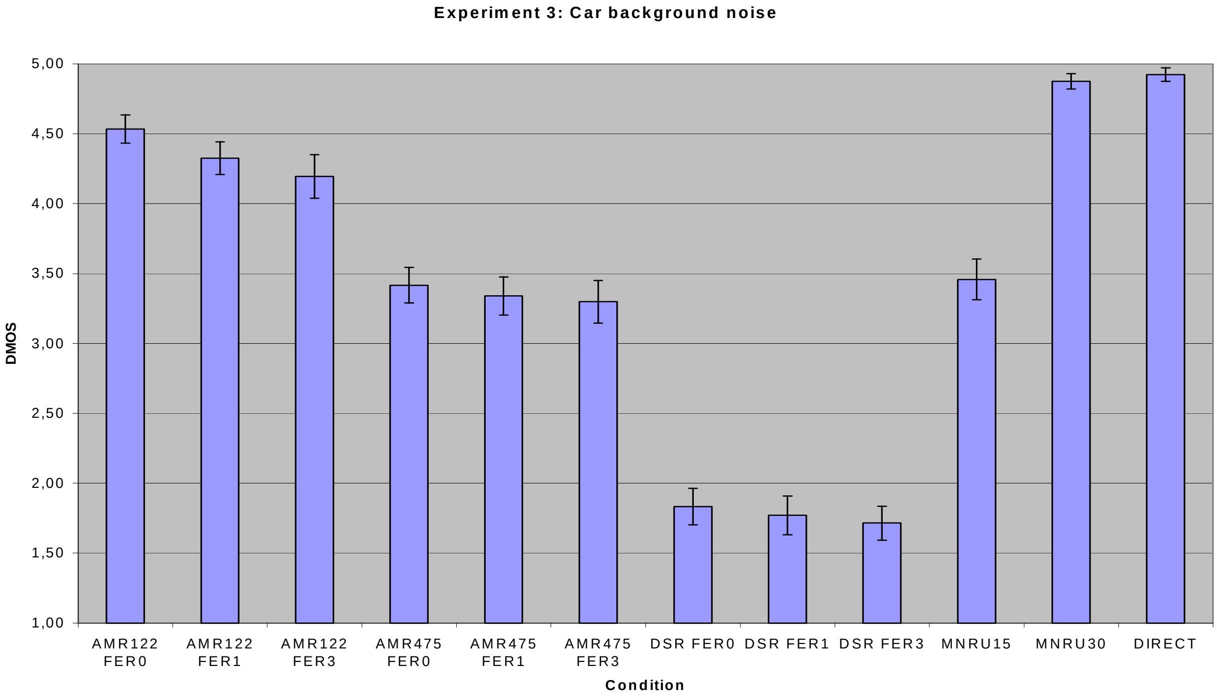


Figure 3: Results of experiment 3 – Noisy speech performance (car) under clean and degraded channel conditions

4. Conclusion

As can be concluded from the listening test results, the speech quality obtained with the DSR codec is under all tested conditions considerably degraded compared to the worst-case quality of the AMR codec. The quality offered by the DSR codec can hardly be considered acceptable, particularly if operated in a context where listening to longer periods of the reconstructed speech is required.

References

[1] S4-030539 “Assessment of the codec capability to reconstruct speech”

[2] ITU-T; Recommendation P.800; “Methods for Subjective Determination of Transmission Quality”

3GPP TSG-SA4#30 meeting
February 23-27, 2004, Malaga, Spain

Tdoc S4-040145

Proprietary Databases

US English In-Car

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	*16.4*
	Other tasks *	w/o coding	*12*
16kHz	Digits	w/o coding	*NA*
		AMR-WB 23.85	*NA*
	Other tasks *	w/o coding	*NA*
		AMR-WB 23.85	*NA*

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	2.98
	Other tasks *	w/o coding	3.01
16kHz	Digits	w/o coding	1.57
		AMR-WB 23.85	1.79
	Other tasks *	w/o coding	2.03
		AMR-WB 23.85	2.18

* Note: for “other tasks” the performance is the average word error rate over the test vocabularies other than the digits.

German In-Car

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	*9*
	Other tasks *	w/o coding	*10.5*
16kHz	Digits	w/o coding	*NA*
		AMR-WB 23.85	*NA*
	Other tasks *	w/o coding	*NA*
		AMR-WB 23.85	*NA*

* Note: for “other tasks” the performance is the average word error rate over the test vocabularies other than the digits.

Mandarin Embedded Corpus

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	2.22
	Other tasks *	w/o coding	2.82
16kHz	Digits	w/o coding	1.56
		AMR-WB 23.85	1.63
	Other tasks	w/o coding	1.97
		AMR-WB 23.85	2.28

* Note: for “other tasks” the performance is the average word error rate over the test vocabularies other than the digits.

Japanese In-Car

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	*9.6*
	Other tasks *	w/o coding	*16.3*
16kHz	Digits	w/o coding	*NA*
		AMR-WB 23.85	*NA*
	Other tasks *	w/o coding	*NA*
		AMR-WB 23.85	*NA*

* Note: for “other tasks” the performance is the average word error rate over the test vocabularies other than the digits.

3GPP “supplied” databases

Aurora-2

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	from 30.28 to 1.59

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	*13.8*

Note that for Aurora-2 the average results should be computed using the ETSI Aurora spreadsheet:

- average over SNRs from 0, 5, 10 15 and 20 dB,
- average over test sets A, B & C
- average of multicondition and clean training conditions

Aurora-3 German

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	*14.5*

Note: Word error rate taken as an average for the three conditions: well matched, medium mismatch and high mismatch.

Aurora-3 Spanish

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	from 3.07 to 21.62
16kHz	Digits	w/o coding	from 2,37 to 14.74
		AMR-WB 23.85	from 2.62 to 13.89

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	*8.3*
16kHz	Digits	w/o coding	*NA*
		AMR-WB 23.85	*NA*

Note: For Aurora-3 word error rate taken as an average for the three conditions: well matched, medium mismatch and high mismatch.

Aurora-3 Italian

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	from 3.25 to 37.38
16kHz	Digits	w/o coding	from 2.02 to 37.24
		AMR-WB 23.85	from 2.43 to 34.23

Note: For Aurora-3 word error rate taken as an average for the three conditions: well matched, medium mismatch and high mismatch.

Aurora-3 Italian under channel errors

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	3.25
		AMR-NB 4.75	4.60
		AMR-NB 4.75 @ 10% BLER	9.73
		AMR-NB 12.2	3.45
		AMR-NB 12.2 @ 10% BLER	10.62
		DSR at 8kHz	2.18

		DSR 8kHz @ 10% BLER	2.67
16kHz	Digits	w/o coding	2.02
		AMR-WB 12.65	2.43
		AMR-WB 12.65 @ 10% BLER	5.53
		AMR-WB 23.85	2.43
		AMR-WB 23.85 @ 1% BLER	2.43
		AMR-WB 23.85 @ 3% BLER	3.15
		AMR-WB 23.85 @ 10% BLER	5.62
		DSR at 16kHz	1.80
		DSR 16kHz @ 10% BLER	2.05

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Digits	w/o coding	*6.4*
		AMR-NB 4.75	*9.4*
		AMR-NB 4.75 @ 10% BLER	*NA*
		AMR-NB 12.2	*6.6*
		AMR-NB 12.2 @ 10% BLER	*NA*
		DSR at 8kHz	*6.5*
		DSR 8kHz @ 10% BLER	*NA*
16kHz	Digits	w/o coding	*NA*
		AMR-WB 12.65	*7.2*
		AMR-WB 12.65 @ 10% BLER	*NA*
		AMR-WB 23.85	*NA*
		AMR-WB 23.85 @ 1% BLER	*NA*
		AMR-WB 23.85 @ 3% BLER	*NA*
		AMR-WB 23.85 @ 10% BLER	*NA*
		DSR at 16kHz	*4.7*
		DSR 16kHz @ 10% BLER	*NA*

Note: For Aurora-3 under channel errors word error rate is for the well-matched condition.

Mandarin name dialling

s-rate	vocabs	codec	Spreadsheet cell
8kHz	Name dialing baseform test	w/o coding	0.56
8kHz	Name dialing tone confusion test	w/o coding	3.56

Low Data Rate comparison

Sampling rate = 8kHz
AMR mode = AMR-NB 4.75

		word error rate		
		AMR-NB 4.75	**DSR**	**Relative Improvement**
Digits	Aurora-2 (result B)	11.73	9.62	17.99%
	Aurora-2 (result A)	16.1	12.4	22.98%
	Aurora-3 German	18.27	13.83	24.30%
	Aurora-3 Spanish (Result A)	9.23	4.86	47.35%
	Aurora-3 Spanish (Result B)	13.93	4.86	65.11%
	Aurora-3 Italian	21.68	6.15	71.63%
	US English In-Car (digits test)	19	12	36.84%
	German In-Car (digit test)	11.4	8.3	27.19%
	Japanese In-Car (digit test)	16.2	9	44.44%
	US English In-Car (digits test)	4.49	2.44	45.66%
	Mandarin Embedded PDA (digit test)	2.57	1.66	35.41%
0.3	**Average improvement on digits tasks**			**39.90%**
Subword	Mandarin Embedded PDA	4.09	2.52	38.39%
	US English In-Car	4.25	2.78	34.59%
	US English In-Car	14.2	9.5	33.10%
	German In-Car	12	10.1	15.83%
	Japanese In-Car	18	13	27.78%
	Mandarin Name dialling (baseform test)	0.83	0.58	30.12%
0.4	**Average improvement on subword tasks**			**29.97%**
Tone Confusability	Mandarin Name dialling (tone confusion test)	3.59	3.06	14.76%
0.1	**Average improvement on tone confusability**			**14.76%**
Channel errors	1% BLER (result A)	5.67	2.39	57.85%
	1% BLER (result B)	9.4	6.7	28.72%
	3% BLER (result A)	6.51	2.38	63.44%
	3% BLER (result B)	17.6	6.8	61.36%
0.2	**Average improvement with channel errors**			**52.84%**

OVERALL RELATIVE REDUCTION IN WORD ERROR RATE **36%**

High Data Rate comparison at 8kHz

Sampling rate = 8kHz
AMR mode = AMR-NB 12.2

		word error rate		
		AMR-NB 12.2	DSR	Relative Improvement
Digits	Aurora-2 (result B)	10.28	9.62	6.42%
	Aurora-2 (result A)	14.2	12.4	12.68%
	Aurora-3 German	15.9	13.83	13.02%
	Aurora-3 Spanish (Result A)	7.7	4.86	36.88%
	Aurora-3 Spanish (Result B)	11.95	4.86	59.33%
	Aurora-3 Italian	19.04	6.15	67.70%
	US English In-Car (digits test)	15.6	12	23.08%
	German In-Car (digit test)	8.6	8.3	3.49%
	Japanese In-Car (digit test)	11	9	18.18%
	US English In-Car (digits test)	3.37	2.44	27.60%
	Mandarin Embedded PDA (digit test)	2.57	1.66	35.41%
0.3	**Average improvement on digits tasks**			**27.62%**
Subword	Mandarin Embedded PDA	3.14	2.52	19.75%
	US English In-Car	3.29	2.78	15.50%
	US English In-Car	12.9	9.5	26.36%
	German In-Car	9.7	10.1	-4.12%
	Japanese In-Car	12.8	13	-1.56%
	Mandarin Name dialling (baseform test)	0.84	0.58	30.95%
0.4	**Average improvement on subword tasks**			**14.48%**
Tone Confusability	Mandarin Name dialling (tone confusion test)	3.81	3.06	19.69%
0.1	**Average improvement on tone confusability**			**19.69%**
Channel errors	1% BLER (result A)	4.73	2.39	49.47%
	1% BLER (result B)	7.1	6.7	5.63%
	3% BLER (result A)	6.33	2.38	62.40%
	3% BLER (result B)	12.6	6.8	46.03%
0.2	**Average improvement with channel errors**			**40.88%**

OVERALL RELATIVE REDUCTION IN WORD ERROR RATE **24%**

High Data Rate comparison at 16kHz

Sampling rate = 16kHz
AMR mode = AMR-WB 12.65

		word error rate		
		AMR-WB	**DSR**	**Relative Improvement**
Digits	Aurora-3 Spanish (Result A)	7.5	4.6	38.67%
	Aurora-3 Spanish (Result B)	7.39	3.47	53.04%
	Aurora-3 Italian	14.77	5.62	61.95%
	US English In-Car (digits test)	17.8	12.3	30.90%
	German In-Car (digit test)	9.2	7.3	20.65%
	Japanese In-Car (digit test)	11.3	8.4	25.66%
	US English In-Car (digits test)	2.04	1.78	12.75%
	Mandarin Embedded PDA (digit test)	1.8	1.14	36.67%

0.35 **Average improvement on digits tasks** **35.04%**

Subword	Mandarin Embedded PDA	2.29	1.63	28.82%
	US English In-Car	2.35	2.31	1.70%
	US English In-Car	13.2	7.8	40.91%
	German In-Car	10.7	7.1	33.64%
	Japanese In-Car	12.3	10.8	12.20%

0.45 **Average improvement on subword tasks** **23.45%**

Channel errors	1% BLER (result A)	2.74	1.84	32.85%
	1% BLER (result B)	7.4	4.8	35.14%
	3% BLER (result A)	3.44	1.84	46.51%
	3% BLER (result B)	10.9	5	54.13%

0.2 **Average improvement with channel errors** **42.16%**

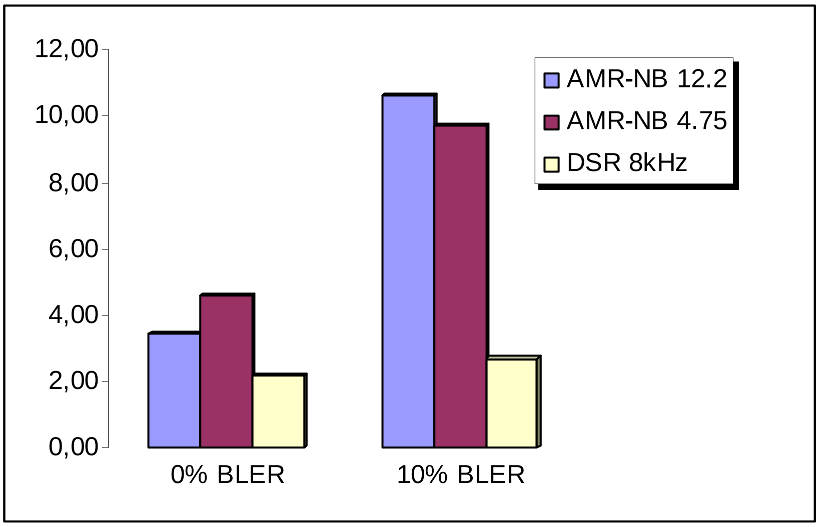
OVERALL RELATIVE REDUCTION IN WORD ERROR RATE **31%**

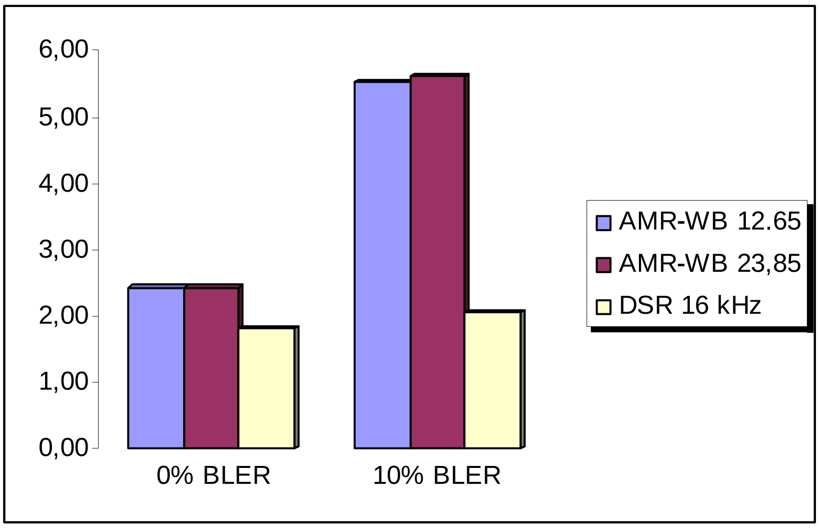
3GPP TSG SA4#30 **S4-040152**
Malaga, Spain, 23-27 Feb 2004 **Agenda Item: SES**

Title: **Results of error resilience for SES codec candidates**
Source: **France Telecom**
Contact: **Denis Jouvet**

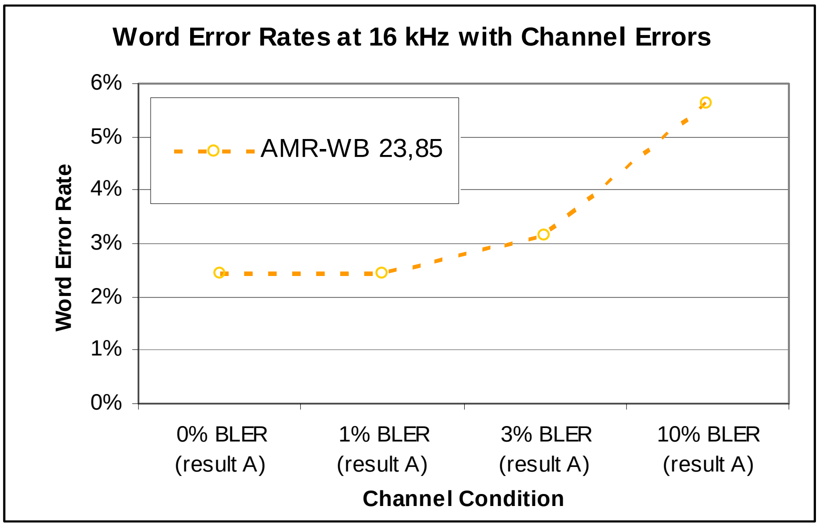
The following figures displays the behaviour of the speech recognition word error rates under channel error. The results are extracted from [1].

This word file provides the word error rates at 0 and 10% BLER, both for 8 kHz and 16 kHz.





At 16 kHz, for AMR-WB 23.85, the intermediate results at 1% and 3% are also provided. They are reported in the following figure.



Conclusion

DSR is robust to channel errors, whereas AMR based speech recognition performances degrades significantly with channel errors.

Moreover, the results at 16 kHz show that the performances degradation significantly gets larger as the channel errors increase.

Apprendix

Results from information results (extracted from [1]):
Aurora-3 Italian under channel errors

s-rate	vocabs	codec	Word Error Rate
8kHz	Digits	w/o coding	3.25
		AMR-NB 4.75	4.60
		AMR-NB 4.75 @ 10% BLER	9.73
		AMR-NB 12.2	3.45
		AMR-NB 12.2 @ 10% BLER	10.62
		DSR at 8kHz	2.18
		DSR 8kHz @ 10% BLER	2.67
16kHz	Digits	w/o coding	2.02
		AMR-WB 12.65	2.43
		AMR-WB 12.65 @ 10% BLER	5.53
		AMR-WB 23.85	2.43
		AMR-WB 23.85 @ 1% BLER	2.43
		AMR-WB 23.85 @ 3% BLER	3.15
		AMR-WB 23.85 @ 10% BLER	5.62
		DSR at 16kHz	1.80
		DSR 16kHz @ 10% BLER	2.05

s-rate	vocabs	codec	Word Error Rate
8kHz	Digits	w/o coding	*6.4*
		AMR-NB 4.75	*9.4*
		AMR-NB 4.75 @ 10% BLER	*NA*
		AMR-NB 12.2	*6.6*
		AMR-NB 12.2 @ 10% BLER	*NA*
		DSR at 8kHz	*6.5*
		DSR 8kHz @ 10% BLER	*NA*
16kHz	Digits	w/o coding	*NA*
		AMR-WB 12.65	*7.2*
		AMR-WB 12.65 @ 10% BLER	*NA*
		AMR-WB 23.85	*NA*
		AMR-WB 23.85 @ 1% BLER	*NA*
		AMR-WB 23.85 @ 3% BLER	*NA*
		AMR-WB 23.85 @ 10% BLER	*NA*
		DSR at 16kHz	*4.7*
		DSR 16kHz @ 10% BLER	*NA*

Note: For Aurora-3 under channel errors word error rate is for the well-matched condition.

Reference

[1] "Results for information from ASR vendors" – word file in S4-040100.

3GPP TSG-SA4#30 meeting
February 23-27, 2004, Málaga, Spain

Tdoc S4-040153

Source: STMicroelectronics[1]
Title: Draft verification plan v0.3
Agenda item: **7**

1. Introduction

This document provides a verification plan for the SES codec selection. The SES candidate selected during SA4#30 (February 23-27, 2004) will be brought to TSG-SA for approval (TSG-SA#23, March 15-17, 2003). Some critical items (as listed in [1]) will be verified by volunteering organizations before the candidate is brought to TSG-SA.

The codecs under consideration are the AFE/X-AFE codec (Advanced DSR front-end and its extension, cf. [3,4]), the AMR-NB codec and the AMR-WB codec. In case of the AMR-NB and AMR-WB codecs are selected then the independent complexity assessment results that are already available from earlier standardisation efforts will be used to verify the complexity. In the case of the AFE/X-AFE codec the fixed-point implementation will be verified.

In the case that SA4 passes decision to TSG-SA because the performance falls in the “grey area” of the recommendation criteria (cf. [11]) and SA4 is unable to reach consensus then verification will also be performed before it is brought to TSG-SA.

2. Verification of bit-exactness

2.1 Motivation

The motivation is to check that the executable used by the ASR vendors corresponds to the executable built from the source code of the selected candidate. A test of "bit-exactness" is used to verify the match of the output bitstreams of the compiled version of the source code of the selected candidate and the executables provided to the two test laboratories for selection testing. Output files from both versions are compared with respect to the bit-exactness.

2.2 Definition

The verification laboratories will make use of:

1. Executables obtained by compiling the source code of the candidate
2. Executables used for selection testing
3. A subset of the samples used for the selection phase.

During the evaluation phase of the AFE/X-AFE algorithm conducted by the testing laboratories, two sampling rates were used, one for the narrowband case (`T8`) and one for the wideband case (`T16`). The binaries were delivered for two different platforms: I386/linux RH7.3 (resp. `T8_linux` and `T16_linux`) and AIX (resp. `T8_AIX` and `T16_AIX`).

Source codes will be provided to the verification laboratories. The executables compiled from the source code are the reference executables to be run at the different sampling rates (resp. `B8` and `B16`).

[1] **Stéphan Tassart**
STMicroelectronics,
Email: stephan.tassart@st.com

Bit exactness will be checked with the VAD flag off since ASR vendors did not use VAD in their evaluations [section 2.3 of 10].

The bit-exactness verification will be made on a subset of the samples used for the selection phase:

Acronym	Description	Duration	Bandwidth	Owner
A3I8	Aurora 3 Italian	8h	8kHz	Alcatel
A3I16	Aurora 3 Italian	8h	16kHz	Alcatel
MND8	Mandarin name dialling	5h	8kHz	Nokia

Table 1: complexity requirements for the SES candidate

2.3 Task

2.3.1 Narrowband verification

The verification laboratory tests the bit-exactness of the output bitstream of the candidate `B8` vs. the output bitstream of the executable `T8_linux` or `T8_AIX` provided to the testing laboratories.

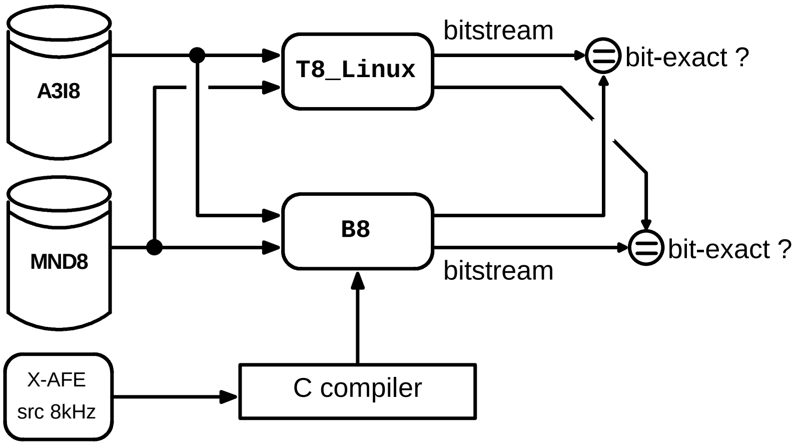


Figure 1: Verification of the bit-exactness of the narrowband candidate

The platform used for verifying the bit-exactness of the candidate is not relevant because the source code of the narrowband candidate is platform independent (i.e. bit-exact on any supported platform). The verification laboratories can use any supported platform for verifying `T8_linux` or `T8_AIX`, i.e. the executable used by the test laboratories.

2.3.2 Wideband verification

The verification laboratory tests the bit-exactness of the output bitstream of the candidate `B16_linux` vs. the output bitstream of the executable `T16_linux` provided to the testing laboratories.

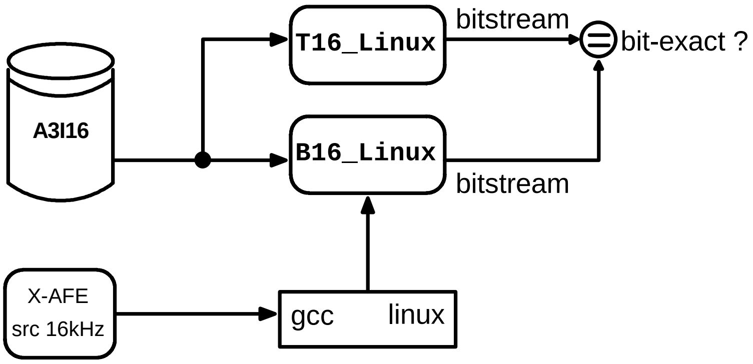


Figure 2: Verification of the bit-exactness of the wideband candidate

Motorola notified to the committee that 6 lines of the code delivered to the testing laboratories were incorrect. Only the wideband case (`T16_linux` and `T16_AIX`) is affected (cf. [5]). The code delivered to the testing laboratories contains a processing block using the floating-point arithmetic (cf. [6]). The verification laboratory checks that the compilation of the source code with the floating-point arithmetic mentioned by Motorola and the executables delivered to the testing laboratories generate identical bitstreams. However, since the IEEE floating-point arithmetic is not bit-exact (cf. [7,8]), the verification of the binaries can be conducted only on a similar platform (same hardware, same compiler, same compilation options).

3. WMOPS Complexity verification

3.1 Motivation

The compiled version of the fixed-point ANSI-C source code must meet the design constraints (cf. [9]). The WMOPS complexity of the candidate will be estimated in the framework of the worst observed frame on a subset of the samples used for the selection phase.

Bandwidth	WMOPS design constraint
narrowband	≤25 WMOPS
wideband	≤39 WMOPS

Table 2: complexity requirements for the SES candidate

3.2 Source-code verification

The source code is used to verify the complexity of the codec. The verification laboratory checks that the C-code has been correctly implemented with basic operators and that the C-code correctly implements the instrumentation that generates a maximum WMOPS score for each sample file.

3.3 Complexity verification

3.3.1 Task

The verification laboratories compile the C-code on one of the supported platforms (gcc on AIX, i386/linux RH7, Sun Solaris 8 or possibly VC++ on win32) and build an executable to be run at the different sampling rates (resp. A8 for the narrowband and A16 for the wideband) (Note: the versions A8 and B8 are identical).

The verification laboratories check that the complexity of the VAD processing is included in the WMOPS complexity verification as indicated in [9].

The executable generates a log file with the maximum observed WMOPS score for each sample file. The verification laboratories process all the files from the selected subset and evaluate the maximum observed WMOPS score. The maximum observed

WMOPS score is evaluated by selecting the maximum WMOPS score from every sample file. The obtained maximum observed WMOPS score is compared with the design constraints (cf. Table 2).

3.3.2 Database selection

The verification of the WMOPs complexity is made on a subset of the samples used for the selection phase (cf. Table 1).

3.3.3 Narrowband verification

The verification laboratory processes the selected databases (**A3I8** and **MND8**) through the A8 executable and produces the maximum WMOPS score.

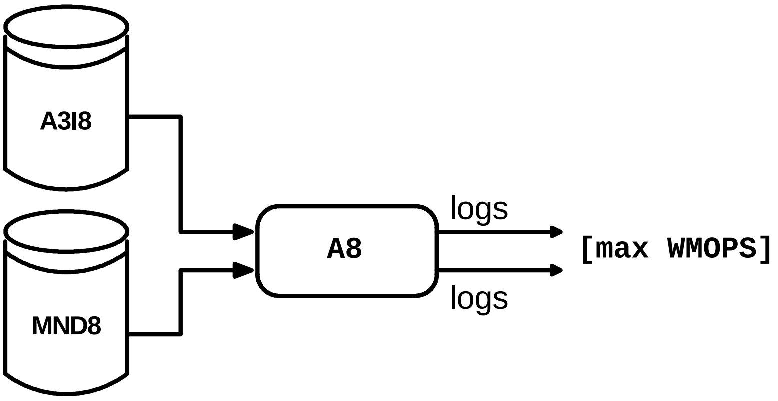


Figure 3: Verification of the complexity of the narrowband candidate

The platform used for the complexity verification is not relevant for the purpose of the WMOPS complexity verification.

3.3.4 Wideband verification

The verification laboratory processes the selected databases (**A3I16**) through the A16 executable and produces the maximum WMOPS score.

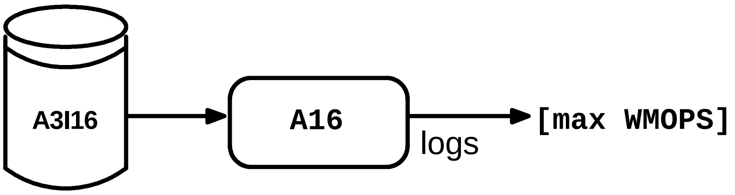


Figure 4: Verification of the complexity of the wideband candidate

The platform used for the complexity verification is not relevant for the purpose of the WMOPS complexity verification.

4. RAM and ROM Complexity verification

4.1 Motivation

The memory used by the fixed-point ANSI-C source code must meet the design constraints (cf. [9]). The memory complexity of the candidate will be estimated from the source code.

Bandwidth	ROM design constraint	RAM design constraint
narrowband	≤20 kwords	≤7 kwords
wideband	≤34 kwords	≤8 kwords

Table 3: memory requirements for the SES candidate (16-bit words)

4.2 Definition

The RAM memory used by the software is the sum of all the non-const arrays or variables defined with a global visibility, all the static arrays or variables (known as the static memory or permanent allocation) and the maximum amount of RAM required by the stack (known as the scratch memory).

The ROM memory used by the software is the sum of all the const arrays or variables (defined in a global or in local visibility). The ROM memory does not include the program ROM (cf. [9]).

The following sample source code explains how the RAM and the ROM memory are evaluated.

```
Word16        buff[16];
const Word32  tab[32];

Word16
func(void *state, Word16 a, const Word16 v[])
{
  Word16 ret;
  Word16 local_buff[8];
  static Word16 state=START;

  [...]

  return ret;
}
```

Code 1: Example of instrumented C-code

In this small example, the memory complexity would be evaluated as follow:

C instruction	Type of memory	Accounted for
`Word16 buff[16]`	static RAM	16
`const Word32 tab[32]`	ROM	64
`void *state`	stack	push 1
`Word16 a`	stack	push 1
`const Word16 v[]`	stack	push 1
`Word16 ret`	stack	push 1
`Word16 local_buff[8]`	stack	push 8
`static Word16 state`	static RAM	1
`Return`	stack	pop (-12)

Table 4: Example of memory assessment

4.3 Additional definitions

4.3.1 Static RAM array initialization

Arrays that are allocated and initialised in the static RAM are accounted simultaneously in static RAM and in ROM.

4.3.2 Stack array initialization

Arrays that are allocated and initialised in the stack are accounted only in static RAM. Furthermore, the code shall be instrumented with as many `move16()` (resp. `move32()`) basic operations than necessary in order to take into account the actual initialisation process. Here follows a small example:

```
Word16
func_proc(Word16 a, Word32 b)
{
  [...]
  Word16 autoBuff[4]={0x4000, 0x1400, 0xFC00, 0xAFF0};
  move16();move16();move16();move16();

  [...]

  return 0;
}
```

Code 2: Instrumented C-code initializing an array in the stack

Said differently, the process of initialising an array allocated in the stack is formally equivalent to the following C-code fragment:

```
Word16
func_proc(Word16 a, Word32 b)
{
  [...]
  Word16 autoBuff[4];

  autoBuff[0] = 0x4000; move16();
  autoBuff[1] = 0x1400; move16();
  autoBuff[2] = 0xFC00; move16();
  autoBuff[3] = 0xAFF0; move16();
  [...]

  return 0;
}
```

Code 3: Unambiguous equivalent C-code for initializing an array in the stack

4.3.3 Constant value usage

Most C compilers for DSP will inline Word16 and Word32 constant values directly in the assembly language code. Therefore, constant values (such as 0x00400000L and 25798L) will not be included in the data ROM; instead they are included in the program source code.

4.3.4 Summary

The following table sums up the different configurations considered for assessing the complexity and the memory usage regarding the usage of constant values in the reference C-code.

C instruction	Type of memory	Accounted for
Word16 swRand[4]={…};	ROM + static RAM	4 each
Word16 autoBuff[4]={…};	stack	push 4
((Word16)0x(vvvv))	program	transparent
0x(hhhhllll)L	program	transparent

Table 4: Memory assessment for initialization of arrays and constant value usage

4.3.5 Example C-code

This following imaginary sample code (which does nothing in particular) illustrates different cases that shall be taken into account for the memory assessment of the SES codec :

```
/* initialization counting for 4 words in the ROM */
Word16        swRand[4] = {8, 12, -4, -7};

Word16
func_proc(Word16 a, Word32 b)
{
  Word16 idx, idx2;

  /* constant value counting for 0 words ROM */
  Word32 enerLog = 0x00400000L;

  /* initialization counting for 0 word ROM */
  Word16 autoBuff[4] = {0x4000, 0x1400, 0xFC00, 0xAFF0};

  /* enerLog initialization */
  move32();

  /* autoBuff initialization */
  move16();move16();move16();move16();

  [...]
  /* loop preparation */
  idx2 = 0;   move16();
  for (idx=0;idx<4;idx++) {
    [...]
    autoBuff [idx] = swRand[idx2]; move16();
    swRand[idx2] = /* small constant 25798L counting 0 word ROM */
       extract_h(L_shr(L_add(25798L,
                              L_mult(swRand[idx2], 10037)),2));
    move16();
    [...]
  }

  [...]

  return 0;
```

Code 4: Sample instrumented C-code

4.4 ROM verification

The source code is used to evaluate the ROM complexity. The amount of ROM memory used by the candidate, as evaluated by the verification laboratories, is compared to the design constraints (cf. Table 3).

4.5 RAM verification

4.5.1 Permanent RAM verification

The source code is used to evaluate the RAM usage that is not related to the use of the stack. The verification laboratory enumerates all the array and variable definitions corresponding to a permanent allocation.

4.5.2 Stack verification

The source code is used to evaluate the stack usage. The verification laboratory builds the calling tree of the source code and evaluates the worst case for the stack usage.

4.5.3 Conclusion

The verification laboratory sums the amount of static RAM and the maximum amount of RAM required by the stack. The amount of RAM memory is compared to the design constraints (cf. Table 3).

5. Workplan

5.1 Verification laboratories

The verification will be performed by STMicroelectronics (contact is stephan.tassart@st.com) and IBM (contact is sorin@il.ibm.com).

Task	Company
bit-exactness verification, narrowband,linux (cf. 2.3.1)	ST
bit-exactness verification, wideband linux (cf. 2.3.2)	ST
bit-exactness verification, narrowband AIX (cf. 2.3.1)	IBM
source code verification (cf. 3.3.2)	ST
WMOPS verification, narrowband (cf. 3.3.3)	ST
WMOPS verification, wideband (cf. 3.3.4)	ST
RAM verification, narrowband (cf. 4.4)	ST
RAM verification, wideband (cf. 4.4)	ST
ROM verification, narrowband (cf. 4.4)	ST
ROM verification, wideband (cf. 4.4)	ST

5.2 Schedule

The workplan is organized as follow:

Date	Actions
19th Dec. 2003	Agree the verification plan by correspondence

16th Feb. 2004	Complete legal agreements with Alcatel for the A3I8 and A3I16 speech databases. Verification laboratories to obtain A3I8 and A3I16.
19th Feb. 2004	Complete legal agreements (NDA) with Motorola for the X-AFE source code.
5th Mar. 2004	Complete legal agreements with Nokia for MND8 speech database.
16th Feb. 2004	The I/O interface and the format of the log files of the X-AFE candidate are provided to the verification laboratories.
1st Mar. 2004	The testing laboratories to provide the executables (i.e. T8_linux and T16_linux) to the verification laboratories.
23rd –27th Feb.	Meeting SA4#30 – Malaga
1st Mar. 2004	DSR supporting companies to provide the source code to the verification laboratories. The verification laboratories compile the source code and obtain a binary (i.e. B8 and B16_linux).
1st - 3rd Mar.	Bit-exactness verification: B8 versus T8_linux on A3I8.
1st - 3rd Mar.	Verification of the source code instrumentation.
4th - 5th Mar.	Complexity wMOPs verification: A8 on A3I8.
1st - 10th Mar.	Verification of the RAM and ROM figures.
8th Mar.-10th Mar.	Complexity wMOPs verification: A8 on A3I16.
10th Mar. 2004	Conference call: discussion of partial verification results.
10th Mar. 2004	Verification laboratories to obtain MND8.
11th - 12th Mar.	Bit-exactness verification : B16_linux versus T16_linux on A3I16.
11th - 12th Mar.	Bit-exactness verification : T8_AIX versus T8_linux on A3I8.
15th Mar. 2004	Partial verification report completed: memory assessment completed, wMOPs assessment partially completed (A3I8, A3I16), bit-exactness verification partially completed (A3I8, A3I16)
15th - 17th Mar.	Meeting TSG SA4#23
15th - 17th Mar.	Bit-exactness verification : B8_linux versus T8_linux on MND8.
18th - 19th Mar.	Complexity wMOPs verification: A8 on MND8.
26th Mar.	Verification report completed.

6. References

[1] S4-030745 “SES codec verification”

[2] S4-030852 “SES Workplan version 8.0”

[3] ETSI standard ES 202 050 “Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithms”, Oct 2002, http://pda.etsi.org/PDA/home.asp?wki_id=yeZ1Qi@QwpOPXVVTO7wZ2

[4] ETSI standard ES 202 212 “Distributed Speech Recognition; Extended Advanced Front-end Feature Extraction Algorithm; Compression Algorithm”, Nov 2003, http://pda.etsi.org/PDA/copy_file.asp?Action_type=&Action_Nb=&Profile_id=IugJxMadBBxgVRiTVU7weOO&Wki_id=yPyx-MSKzNpqwrsvVBZ_Z

[5] S4-030853 “Draft Report SQ and AUC ad-hoc sessions during SA4#29 plenary meeting”

[6] S4-030866 “Consideration of DSR executable code update to ASR vendors”

[7] IEEE 754-1985 Standard for Binary Floating-Point Arithmetic

[8] IEEE 854-1987 Standard for Radix-Independent Floating-Point Arithmetic

[9] S4-030248 “Design Constraints for default codec for speech enabled services (SES)”

[10] S4-030543 “Test and processing plan for default codec evaluation for speech enabled services (SES)”

[11] S4-030540 “Recommendation Criteria for default codec for speech enabled services (SES)”