# LifeVibes™ Handsfree

## One-person tuning: setup and methodology

PROPOSED $Revision:: 1.0 $ - 2005-May-13 CONFIDENTIAL

Document information

| Info | Content |
|---|---|
| Title | LifeVibes™ Handsfree<br>One-person tuning: setup and methodology |
| Status | PROPOSED |
| Version | $Revision:: 1.0 $ |
| Security | CONFIDENTIAL |
| Author | Rob Goyens |
| Reviewers | Wouter Tirry, Vincent Demanet |
| Document ID | SER-05xxx |

**Philips Software**
**Product Line: Sound**
**1person-tuning**

PHILIPS

# CHANGE HISTORY

| Revision | Description | |
|---|---|---|
| 1.0.0 | Initial Revision | |
| | Rob Goyens | 2005-05-11 |

# 1 INTRODUCTION

## 1.1 Scope

This document describes the setup and methodology for tuning the LifeVibes™ Handsfree algorithm without accessing internal signals of the mobile terminal.

LifeVibes™ Handsfree is suited for acoustic echo cancellation in the following situations:

- Handset or normal mode;
- Headset mode (acoustic echoes are becoming more and more of a problem for Bluetooth headsets of which sizes are shrinking continuously, increasing the coupling between the earpiece and the microphone);
- Speakerphone or handsfree mode (because of the high loudness of the speaker signal this is the most challenging mode for an AEC);
- Car kit.

This tuning-document focuses on tuning the Speakerphone mode. But with some adaptation it can be used for Handset mode too.

## 1.2 Definition of tuning

Tuning can be defined as 'checking and/or adjusting the parameters of the algorithm to achieve the desired algorithm performance'. This can be done by sending test signals to the mobile terminal and recording the transmitted signal from the mobile sent back to the network. This way a tuning-engineer can visually and auditory analyse the transmitted signal. Based on this analysis he/she can adjust parameter-settings of the algorithm to achieve a better performance. The test signals used for tuning are delivered together with this document.

For a detailed description of the parameters of LifeVibes™ Handsfree we refer to the Handsfree user manual [1].

## 1.3 Relation between terminal and tuning

Each mobile terminal has its own specific characteristics that influence the AEC-performance: acoustical coupling factor between loudspeaker and microphone, non-linear effects in amplifier and loudspeaker, internal acoustic reverberation, frequency response of loudspeaker, microphone sensitivity, etc. Therefore, LifeVibes™ Handsfree has to be tuned for each phone model to guarantee optimal performance: **one phone = one tuning**.

# 2 PERFORMANCE

Before proceeding to the setup and the methodology, it is important to define the 'performance' of the LifeVibes™ Handsfree algorithm. Beside an Acoustic Echo Canceller (AEC) LifeVibes™ also provides a stationary Noise Suppressor (NS).

## 2.1 AEC - performance

The AEC cancels the acoustic echo that is caused by the coupling between the mobile loudspeaker and the mobile microphone. For the AEC we can distinguish two important performance metrics: (i) the suppression of the farend echo during single talk and (ii) the quality and intelligibility of the nearend speech during double talk (i.e. both farend and nearend are talking at the same time).

We mainly distinguish three kinds of echoes depending on the occurrence in time or depending on the origin of the echo:

- **Initial echoes**: these echoes can occur during single talk immediately after setting up a call (e.g. first 4 seconds). The reason for this kind of echoes is that the convergence time of the algorithm is too high;
- **Tail echoes**: these echoes can occur anytime during the call. Because of a too short echo path-model the late part, or tail, of the echo is not modelled and cannot be suppressed by the algorithm. Tail echoes have almost the same frequency content as the farend-speech;
- **Non-linear echoes**: these echoes can occur anytime during the call. Non-linear components (amplifier, loudspeaker, etc) in the system generate harmonic frequencies, which cannot be modelled by a linear system. Therefore non-linear echoes have a high-frequent frequency content.

The quality of the transmitted signal during 'farend only' can be assessed by:

- Calculating the echo suppression or ERLE;
- Calculating the spectrogram of the processed signal to detect disturbances in the frequency-domain (e.g. non-linear echoes);
- Calculating the level of the processed signal to detect glitches/disturbances in the time-domain;
- Listening!

Suppressing farend echoes during double talk can attack the desired nearend speech. The algorithm has to be tuned to preserve the quality of the nearend speech as much as possible. Tuning is often making a trade-off between echo suppression and nearend quality during double talk.

We distinguish following classes in duplex capability:

- Good full-duplex: no distortion on nearend speech during doubletalk, high intelligibility;
- Full-duplex: some distortion (e.g. low pass effect) and small attenuation on nearend speech during doubletalk;

- Interruption possibility: nearend speech is not understandable, but the farend speaker can hear that the nearend speaker is trying to interrupt;
- Half duplex: nearend is completely muted during doubletalk, no interruption possibility;

Calculating the suppression of the nearend speech during doubletalk gives a first indication of the quality.

## 2.2 NS - performance

The NS suppresses the nearend stationary noise. Stationary noise is defined as noise of which the characteristics are static in time. Types of noise that are typically being suppressed are noise from fans, car engine noise and the stationary component of any type of noise. For the NS, we can also distinguish two different performance metrics: (i) the amount of stationary noise suppression and (ii) the quality of the transmitted background noise.

To avoid any speech distortion we recommend 10dB of stationary noise suppression in handsfree-mode (15 dB in handset-mode).

With quality of transmitted background noise we mean the absence of disturbing effects like noise-gating: the noise is modulated by the farend-speech and therefore the level is not constant in time.

# 3 TUNING SETUP DESCRIPTION

Figure 1 shows the one-person tuning setup for tuning LifeVibes™ Handsfree. The setup consists of:

- A PC or laptop enabled with a multitrack soundcard and with a multitrack 'playback and recording'-application (like CoolEdit Pro™ or Adobe® Audition™, etc);
- An interface to the mobile network;
- A loudspeaker representing the nearend-speaker;
- The mobile to be tuned.

The tuning should be performed in a quiet room (but not an anechoic room!) to prevent that echoes are masked by ambient noise.

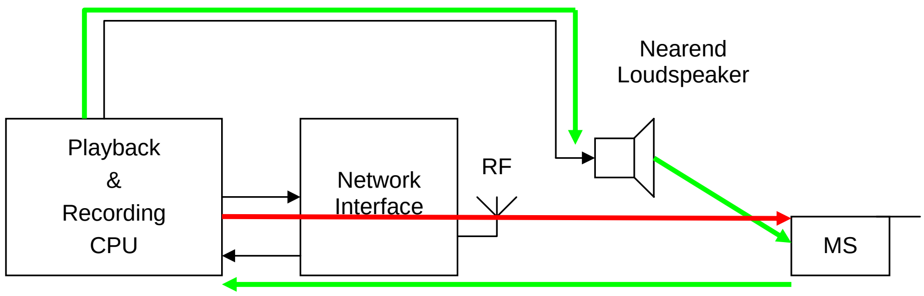



**Figure 1: One-person tuning setup.**

## 3.1 Positioning and volume setting

The position of the nearend loudspeaker should be at a normal 'handsfree'-distance from the mobile microphone, i.e. 50 cm (for tuning Handset-mode the nearend speaker must be positioned closer to the mobile microphone). If the mobile terminal is equipped with a user volume control, the tuning must be performed on the maximum volume setting because this is the worst case (highest echo signals and most distortion) for the AEC. The setup (i.e. positions, levels, etc) should remain fixed during the tuning procedure to be able to compare the recordings made in the different steps.

## 3.2 Signal directions

Above setup is used to play back test signals in two different directions: receiving and sending direction:

- Receiving direction, i.e. the green colored signal path in Figure 1: the 'nearend'-loudspeaker plays back the test signals in sending direction. This signal is picked up by the mobile microphone and transmitted through the mobile into the network.
- Sending direction, i.e. the red colored signal path in Figure 1: The mobile loudspeaker plays back the test signals in receiving direction. These signals were sent to the mobile via the network interface.

Because of the two different signal directions in the system, the test-signals are delivered as stereo audio files: channel 1 is the signal for the receiving direction (played back by mobile loudspeaker), channel 2 is the signal for the sending direction (played back by nearend loudspeaker).

## 3.3 Recordings

First of all a phone connection has to be established between the terminal to tune and the network interface. The terminal has to be set in the desired operation mode (Speakerphone or perhaps handset). Before each recording the terminal has to be reset because of two reasons: (i) to assure the new parameter-set is active and (ii) to initialize all internal variables of the algorithm, e.g. the NLMS-coefficients, the noise estimation, etc.

Some of the tests require a reference recording to be carried out. With this recording it is possible to calculate the suppression of the echo or the suppression of the nearend speech during double talk, etc. For each reference recording the AEC (both NLMS and DENS) of the mobile terminal needs to be turned off.

## 3.4 Network interface

Figure 2 shows three possible ways to interface to the GSM-network. The top and the middle possibilities connect to the network by the use of an ordinary mobile acting as GSM-modem. Audio-signals can be sent and received through either a car-kit or a headset connection. The third possibility is using the audio-interface of a networksimulator.

For the one-person tuning we recommend the use of a networksimulator because the settings of this simulator can be chosen so that the audio-signal is not influenced by any signal-processing (e.g. DTX[1]). In the other two possibilities, depending on the used networkoperator, the audio-signal can be processed. In this case it is more difficult to reproduce the same recordings.

[1] DTX or Discontinuous Transmission is a method of momentarily powering-down, or muting, a mobile or portable wireless telephone set when there is no voice input to the set. This optimizes the overall efficiency of a wireless voice communications system.

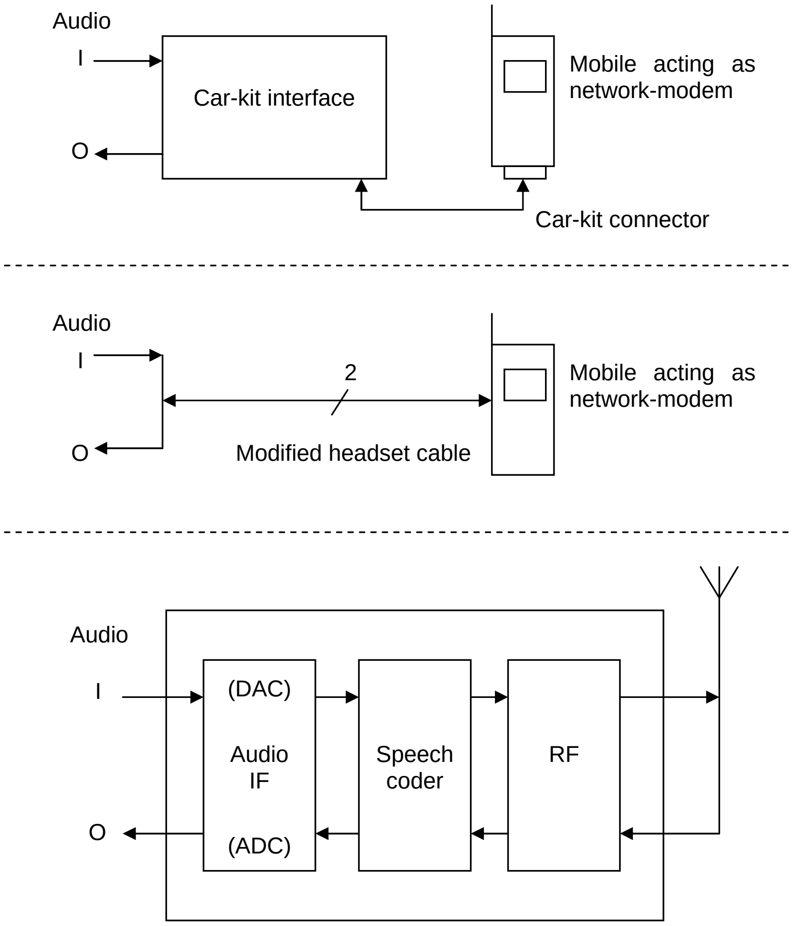



**Figure 2: Three possible ways to interface to the GSM-network. Top: car-kit connection to mobile acting as modem. Middle: headset connection to mobile acting as modem. Bottom: audio-interface of networksimulator.**

# 4 CALIBRATION

To assure proper operation of the tuning setup the following preparation has to be completed:

- Calibration of the sound levels of the nearend and mobile loudspeaker;
- Settings of the soundcard/recording application.

The sound levels produced by the mobile loudspeaker and the nearend loudspeaker have to be calibrated to be comparable to normal handsfree operation.

For the calibration we measure the sound pressure level or SPL at a distance of 10cm from the loudspeaker. Figure 3 and Figure 4 show the setup for calibrating in sending direction and receiving direction respectively. The SPL is measured with a fast integration time and with spectral weighting according to curve A.

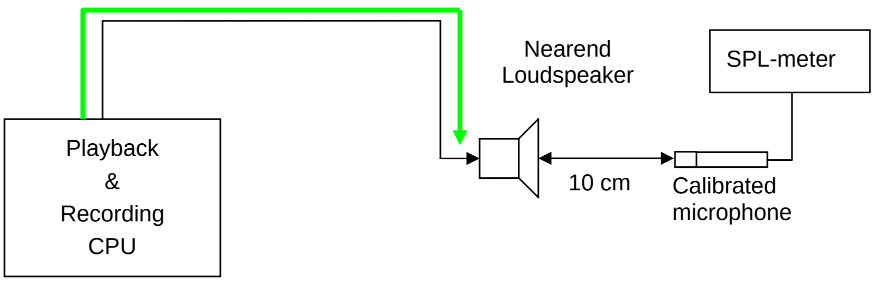



**Figure 3: Calibration in sending direction**

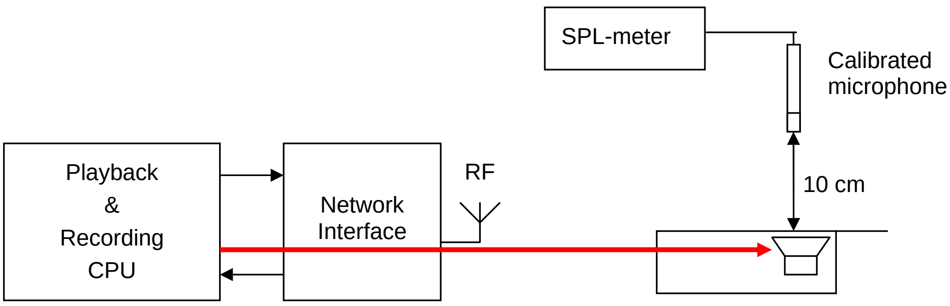



**Figure 4: Calibration in receiving direction**

As calibration signal we use a series of composite source signal-bursts or CSS-bursts (*css_snd.wav* and *css_rcv.wav*). The sound levels of the soundcard should be adapted such that the maximum SPL of the calibration-signal is (around) 82 dB SPL for both nearend- as mobile-loudspeaker.

The settings of the soundcard should be such that there is no level/gain difference between the output and input of the network interface. This can be checked by short-circuiting the interface to the mobile network (connect soundcard-out to soundcard-in, see also Figure 5) and play back the calibration signal in receiving direction (*css_rcv.wav*) and record the signal linked back to the input of the soundcard. The level of the recorded signal should be the same as the played signal.

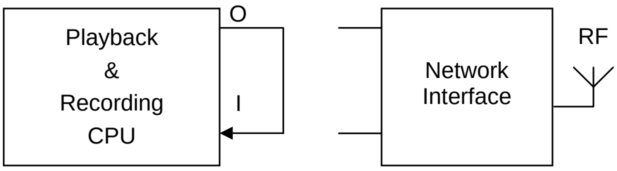



**Figure 5: Checking soundcard-settings**

# 5 TUNING METHODOLOGY

## 5.1 Overview

Figure 6 describes the general steps to be followed to tune the LifeVibes™ Handsfree module.

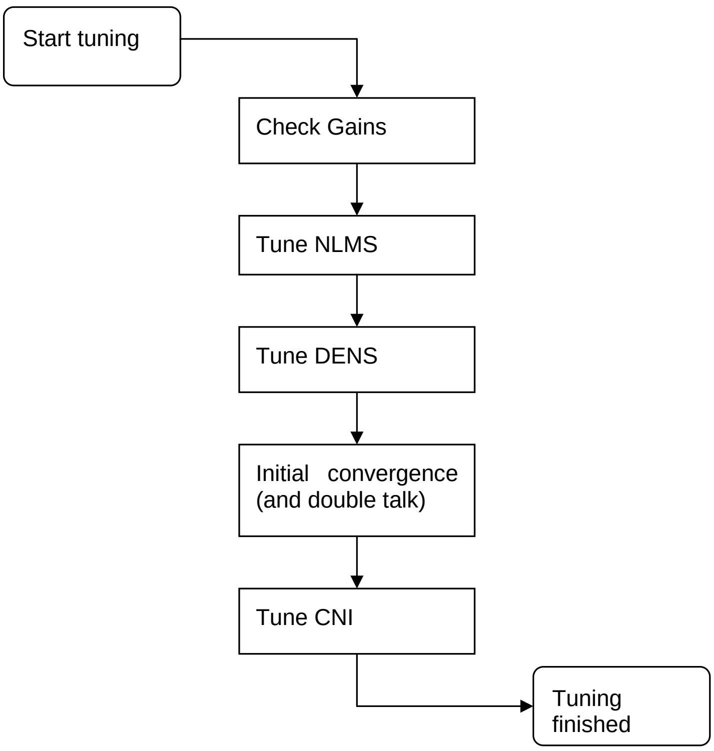



**Figure 6: General tuning steps**

In the next paragraphs each of the tuning steps is explained more in detail.

## 5.2 Step 1: Check Gains

### 5.2.1 Background

The position of the AEC in the audiopath is shown in Figure 7. The analog and digital gain-settings in the audiopath can influence the performance of the AEC in several ways:

- If the level of the NLMS-reference signal is lower than the level of the NLMS-microphone signal the coefficients of the NLMS-filter can saturate (coefficients are limited to the range [-1, 1 [). As result, the direct path cannot be modeled accurately by the NLMS-filter and the echo suppression-performance will be non-optimal;
- If the acoustic echo signal is clipping in the analog or digital gain of the Tx-path before the AEC, extra non-linear components will be added which also will result in a non-optimal echosuppression.

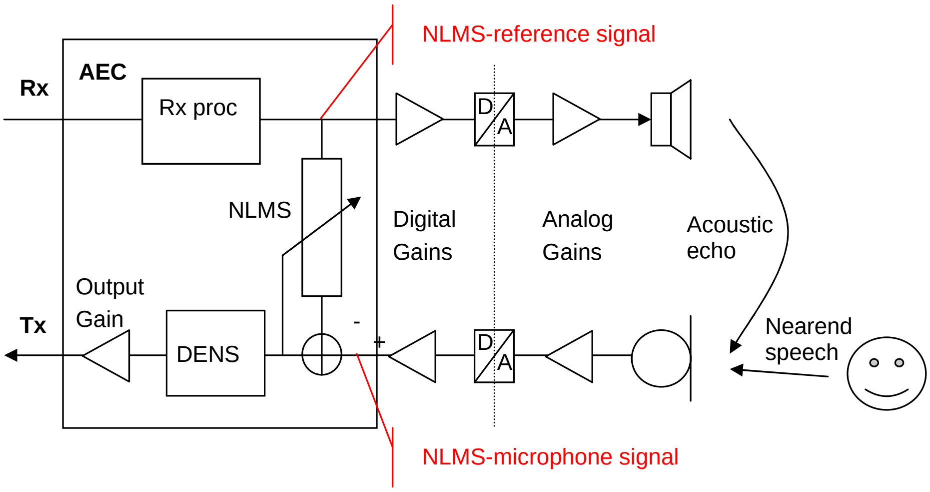


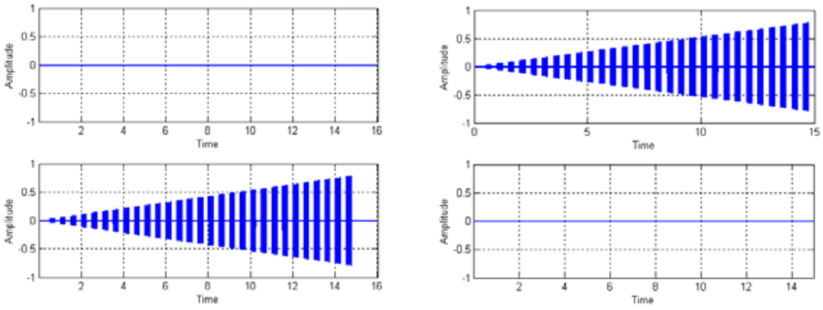

**Figure 7: AEC in the Audio path**

This test will only check whether the echo-signal clips before the AEC during only farend. To detect clipping NLMS-coefficients we refer to step 2 of the tuning methodology.

Because we don't use internal signals we can only analyze the transmitted Tx-signal. Therefore it is difficult to determine whether the NLMS-microphone is clipping due to clipping in the amplifiers in the Tx-path or due to limitations in the Rx-path (loudspeaker or loudspeaker amplifier). We bypass this problem by first determining the clipping level of the NLMS-microphone signal by playing a test signal via the nearend loudspeaker (fully under control, not limited).

### 5.2.2 Test signal

Figure 8(a) shows the test signal (*check_micgain_ref.wav*) played by the nearend loudspeaker to detect the clipping level of the Tx-gains. Figure 8(b) shows the test signal (*check_micgain.wav*) played by the mobile loudspeaker, for detecting a clipping echo signal. The test signals consist of a series of multitone-signal-bursts (sine waves at different frequencies), increasing in level.




(a) (b)

**Figure 8: Test signals used for checking the Tx-gains: (a) signal to check clipping level and (b) signal to check echo level.**

### 5.2.3 Procedure

Following procedure has to be followed (also depicted in Figure 9):

1. Preparation: turn off both the DENS and the NLMS, set the outputgain after the DENS to 0 dB (to assure the signal doesn't clip there!). To assure that the signal played by the nearend loudspeaker clips in the gains before the AEC, we position this loudspeaker close to the mobile microphone (e.g. 5 cm) and, if needed, the digital Tx-gain can also be increased.
2. Play back the test signal by the nearend loudspeaker, record the transmitted signal and determine the clipping level.
3. Play back the test signal by the mobile loudspeaker and record the transmitted signal. Compare the level of the recorded echo with the level found in step 2. If the level of the echo reaches the clipping level of the nearend the gains in the Tx-path should be reduced until there is headroom of 3 to 6dB. If the level of the echo is already much lower than the clipping level, keep the gains.
4. Compensate the reduced gain with the outputgain of the DENS to keep the overall gain in the Tx-path.

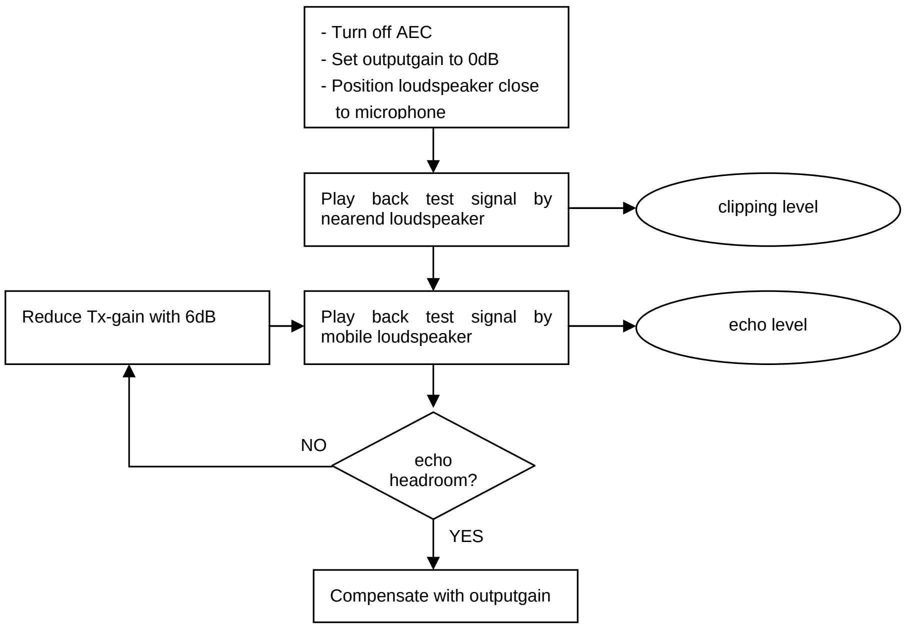
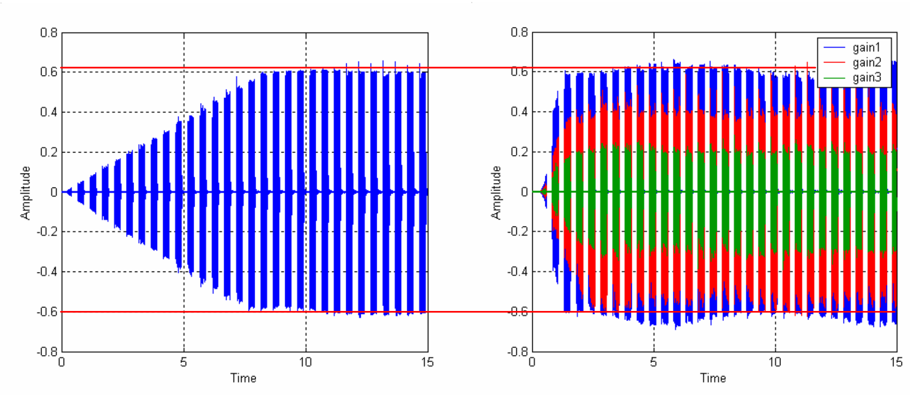



**Figure 9: Procedure for checking the Tx-gains.**

## 5.2.4 Result

Figure 10 shows the result of this test; in Figure (a) the signal played by the nearend loudspeaker is depicted. The clipping level can easily be determined from this graph. Figure (b) shows the transmitted echo signal for three different settings of the digital Tx-gain. For gain1 and gain2 the echo signal can reach the level where the Tx-gains clip. Therefore these gain-settings are too high. For gain3 the echo signal never reaches the clipping level, so this setting is all right.




(a) (b)

**Figure 10: Transmitted Tx-signal. (a) Signal played by nearend loudspeaker, (b) Echo signal at three different settings for digital Tx-gain.**

## 5.3 Step 2: Determine NLMS-filterlength

### 5.3.1 Background

The NLMS-filter models the acoustical path between the loudspeaker and the microphone. The filterlength **N** should be chosen such that most of the energy of the real (unknown) impulse response is covered by the filter. Therefore, the filter should model the direct path and a part of the diffuse field of the echo.

Increasing the filterlength will:

- Increase MIPS without improving performance;
- Increase convergence time.

### 5.3.2 Test signal

The test signal (*tune_NLMS.wav*) used for determining an appropriate filterlength is shown in Figure 11. It consists of a 4-second white noise training sequence followed by CSS-bursts at four different levels (there is a 6dB difference between the levels). The NLMS-filter should be adapted on the training sequence before the start of the CSS-bursts.

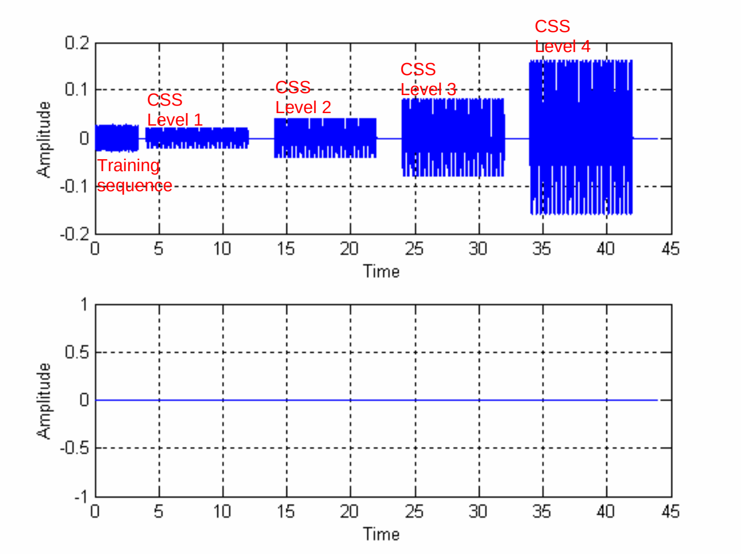



**Figure 11: Test signal used for determining an appropriate adaptive filterlength**

### 5.3.3 Procedure

Figure 12 shows the procedure to tune the filterlength. The first step is making a reference recording with the same signal used for tuning the NLMS (AEC turned off). Next, the NLMS is turned on and the filterlength N is swept from *Nmin* to *Nmax* with steps of *Nstep* (see Appendix).

After sweeping the filterlength, the analysis of the recordings can start. For each recording with filterlength *xx* the echo suppression of the NLMS is calculated as:

$$ERLE = 10\log_{10}\left(\frac{\varepsilon\{Nxx^2\}}{\varepsilon\{ref^2\}}\right)[dB]$$

during the active periods of the CSS-bursts.

An appropriate filterlength can be chosen from the curve representing the echo suppression as function of the filterlength. This filterlength should be used in the further steps of the methodology.

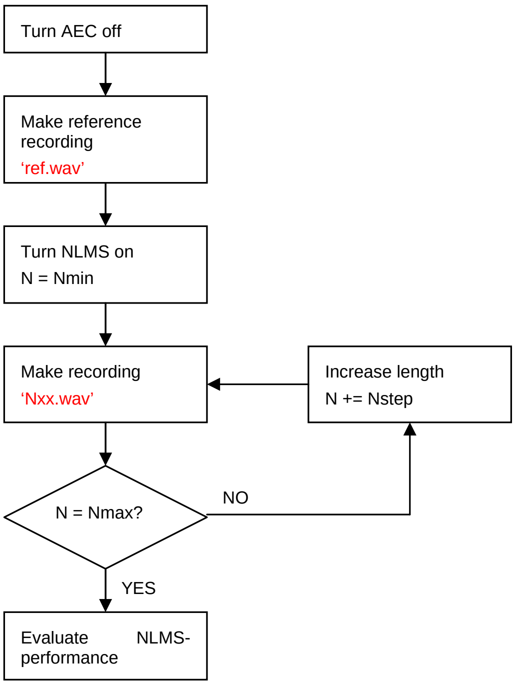



**Figure 12: Procedure for determining the NLMS filterlength**

Two problems can occur while tuning the NLMS: (i) the NLMS doesn't converge well or (ii) the suppression of the NLMS is abnormal low. The first problem can be solved with the convergence-parameters (see also step 4), the second problem can be solved by adjusting the gain-settings.

Normally the NLMS-filter should be converged within the initial training sequence. If it takes more time to converge, set the **ERL**-parameter to a very low value. If this doesn't reduce the convergence time, some other convergence-problems can be present:

- Oscillating convergence: the filter doesn't longer approach to his optimal solution smoothly but shows the tendency to overshoot the optimum and oscillate very slowly to the optimal solution. In this case the stepsize **twoalpha** has to be decreased.
- Marginal convergence and divergence: filter doesn't converge at all, or even 'runs' away from the optimal solution. In this case the stepsize is also too large and has to be decreased.

It is recommended not to increase the stepsize twoalpha from its default value. This could lead to a non-stable filter.

A normal functioning NLMS should yield an ERLE around 15dB or more. If the NLMS converges well but if the suppression of the NLMS is substantially lower for each filterlength, then most probably the coefficients of the NLMS are clipping (see also step 1). In this case the analog (if possible) or digital gain before the AEC should be reduced and compensated with the outputgain after the DENS, such that normal NLMS-behavior is reached.

### 5.3.4 Result

Figure 13 shows the recorded reference signal together with the suppressed echo signal for a certain filterlength. As can be seen from this figure the convergence is well within the initial training sequence. So the convergence-settings are all right for this test.

Figure 14 shows the NLMS echo suppression-curves for two different mobile phones. Each curve has more or less a saturation behavior. An appropriate filterlength is a value somewhat above the knee-point of the curves. Otherwise too much processing power is spent without much improvement. So, for these phones a filterlength of around 80-90 seems to be a good value.

There is also a noticeable difference between the echo suppression for the different levels of the CSS-bursts. Lower levels are more suppressed than higher levels. Higher levels lead to non-linear echo components due to limitations of the amplifier and the loudspeaker. A linear filter cannot cancel these non-linear components; furthermore the non-linear components can result in a jitter on the NLMS-coefficients. The more linear the system, the better the NLMS can model the echo path.

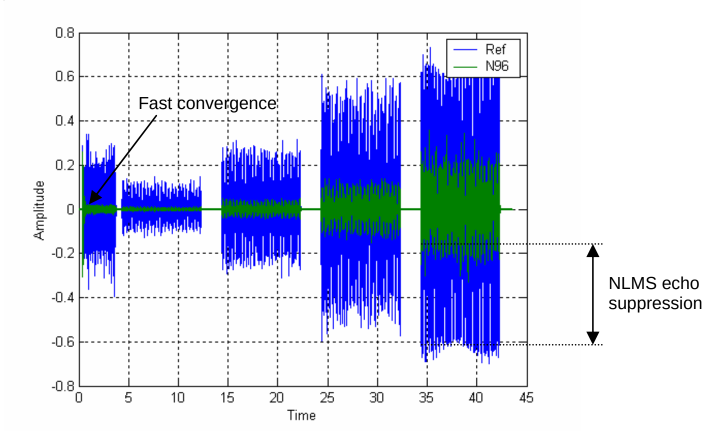



**Figure 13: NLMS reference signal together with suppressed echo signal.**

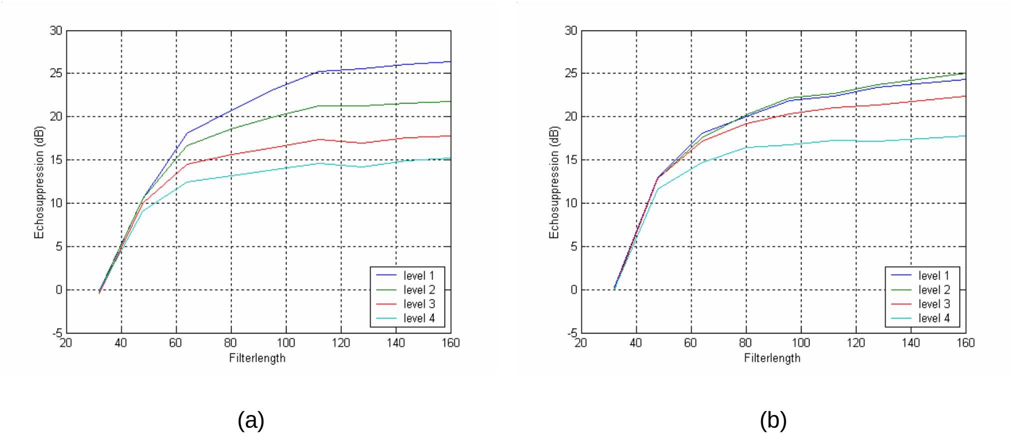



**Figure 14: NLMS echo suppression in function of filterlength. (a) Bartype phone, (b) Slider phone, slider open**

## 5.4 Step 3: Determine DES-parameters

### 5.4.1 Background

The NLMS-filter cannot suppress all the echoes because of the following reasons:

- Under-modeling: the length of the acoustic impulse response is infinite and the adaptive filter is always an insufficient model.
- Tracking: due to movements, the impulse response is always changing, which might not be directly followed by the adaptive filter.
- Non-linear distortion: there are non-linear distortions present in the system (amplifier, loudspeaker, housing, microphone) and they result in non-linear distorted echoes, which cannot be modeled and cancelled by the linear adaptive filter.

带格式的：项目符号和编号

The post processing echo reduction or DES removes such remaining echoes.

The parameters of the post-processing that need to be tuned in this step are:

- **Tail_portion**: corresponds to the ratio of the echo power estimated with the NLMS adaptive filter to the power of the tail echo of the impulse response. Tail_portion determines the amount of energy the DES is going to extrapolate. A too low value will result in remaining tail-echoes while a too high value will result in a degraded double talk performance.
- **Tail_alpha**: is related to the reverberation time of acoustic environment and/or the internal mechanics. Tail_alpha represents the exponential decay of the echo-tail energy. A too low value will result in remaining tail-echoes while a too large value will result in a degraded double talk performance.
- **NL_atten** and **Xclip**: NL_atten sets the amount of non-linear echo suppression. Setting xclip to a low value will activate the non-linear echo suppression mechanism.
- **Gamma_e_high**: is the echo-subtraction factor that is applied during farend-only frames.

### 5.4.2 Test signal

Figure 15 depicts the signal used for tuning the DES-parameters. The signal consists of a short training sequence, followed by an only-farend sequence (male counting) and a double talk sequence in the end (male and female talking with almost complete overlap).

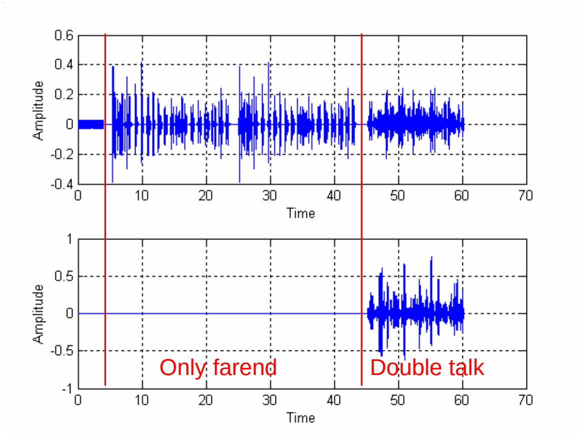



**Figure 15: Signal used for tuning the DES-parameters**

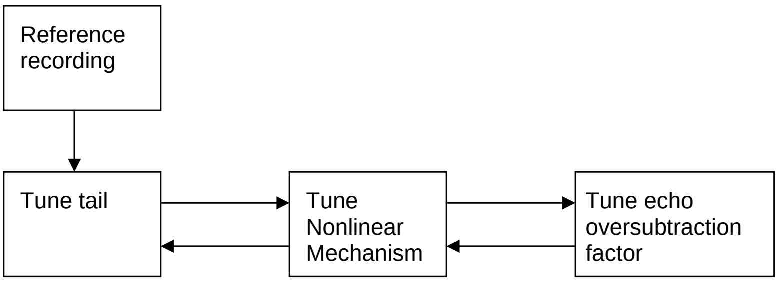



**Figure 16: Tuning procedure for DES-parameters**

### 5.4.3 Procedure

The general procedure is shown in Figure 16. First of all, a reference recording has to be made to be able to calculate the ERLE of the DES and the nearend suppression of the nearend during double talk (AEC turned off!). The signal for this reference recording (*DENS_ref.wav*) is the same as the signal used for tuning the DES, except the double talk sequence is replaced by an only-nearend sequence.

The tuning of the DES starts with tuning the two tail-parameters. For this, the other parameters are set to standard values determined from experience. If tuning the tail is not sufficient or if from the tuning of the tail can be assumed that the standard settings for the other parameters are too conservative, the standard values for the non-linear echo suppression have to be retuned. In very extreme cases, i.e. previous parameters are not sufficient to suppress all echoes, it can be needed to retune the oversubtraction factor.

In the next paragraphs each of the steps is explained more in detail.

#### 5.4.3.1 Tuning the tail

Tuning the tail comes down in finding a combination of tail_portion and tail_alpha that results in excellent echo suppression performance and the best possible double talk performance. Figure 17 shows the complete parameter-space for the tail_alpha and tail_portion (both from 0 to 32767). Each combination of tail-portion and tail-alpha will lead to a certain echo canceling and doubletalk-performance. Because certain combinations of the tail-parameters will certainly lead to a bad AEC-performance (e.g. very low tail_portion and tail_alpha won't suppress enough echoes, very high tail_portion and tail_alpha completely suppress the nearend speech during double talk) it is possible to reduce the search-space. Furthermore, it is not necessary to check each possible value in the reduced search-space, but it suffices to do it with a certain stepsize. So, for the tail-parameters we define maximum and minimum values and the stepsize, which can be found in the Appendix together with some standard values.

To reduce the number of tuning-iterations to come to a good combination, we can make use of following properties:

- For a same tail_portion a decreasing tail_alpha results in better double talk performance but worse echo suppression;
- For a same tail_alpha a decreasing tail_portion results in better double talk performance but worse echo suppression;

This way certain combination can be eliminated for further tuning steps. For example if a combination gives reproducible echoes, it is not needed to try combinations with a lower tail_alpha or lower tail_portion because these will certainly result in echoes.

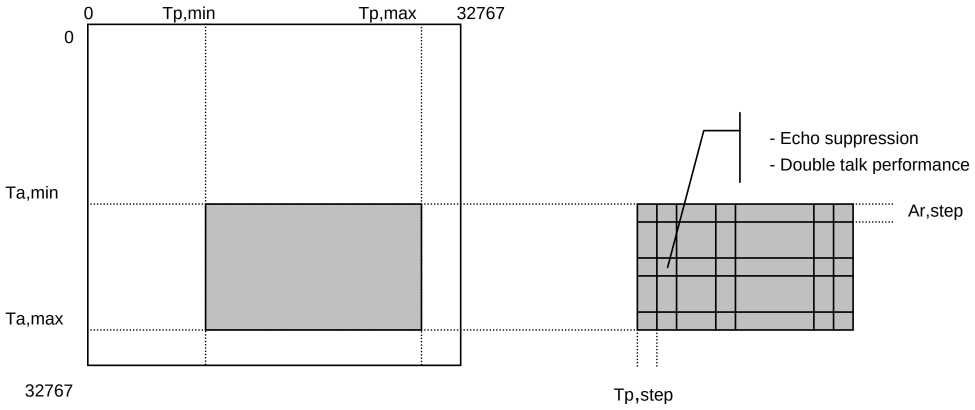



**Figure 17: Parameter-space for the tail**

This results in following tuning-procedure for the tail (see also left graph of Figure 18):

1. Start with tail_portion and tail_alpha at maximum values. Analyze (listen, level-versus-time, spectrogram, etc) the transmitted signal. If this combination results in bad AEC-performance (glitches, echoes) skip the rest of this tuning procedure and go to ‘tuning of non-linear mechanism’.
2. Keep tail_alpha constant to the maximum value and decrease tail_portion to the minimum possible value that doesn’t result in audible echoes. Eliminate all combinations with a lower tail_portion for further investigation.
3. Keep tail_portion to the value found in the previous step and decrease tail_alpha to the minimum possible value that doesn’t result in audible echoes. Eliminate all combinations with same tail_portion but lower tail_alpha.

After these three steps we have a combination wherefore decreasing tail_alpha or tail_portion will result in echoes, which is depicted in the middle graph of Figure 18 as the black rectangle. The red rectangles represent the eliminated combinations. The green rectangles represent combinations with without echoes but worse double talk performance. The blue rectangles represent combinations that are not yet eliminated and could result in the same echo suppression performance but perhaps a better double talk-performance. Therefore, step 2 and step 3 has to be performed also on the remaining blue part, etc (right graph of Figure 18).

All the combinations found this way are combinations wherefore decreasing one of the two tail-parameters results in an echo. The double talk performance of each of these combinations should be compared. Choose the combination with the best double talk performance.

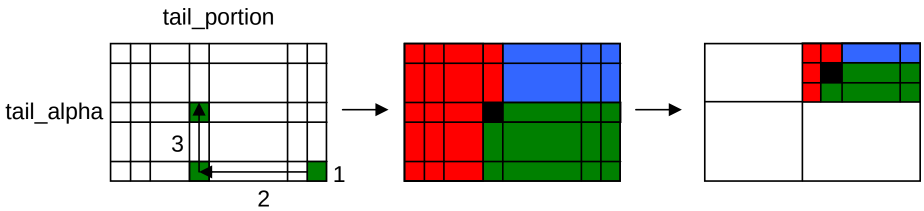



**Figure 18: Tuning procedure for tail-parameters**

### 5.4.3.2 Tuning Non-linear Mechanism

In two scenarios it can be needed to change the parameters of the non-linear mechanism: (i) even with a very low tail_portion and tail_alpha there are no echoes and (ii) even with very high tail_portion and tail_alpha there are echoes.

In the first case, the standard values for the non-linear-mechanism are too high for the mobile. It can be investigated if decreasing the NL_atten with one step and increasing the tail-parameters can improve the double talk performance.

In the second case, the standard values for the non-linear mechanism are too low for this mobile. This can be the case for mobile with a lot of non-linear components. NL_atten should be increased with one step or maximally two steps. If this is not sufficient to suppress all echoes, then it is needed to change the oversubtraction factor.

#### 5.4.3.3 Tuning Echo Oversubtraction Factor

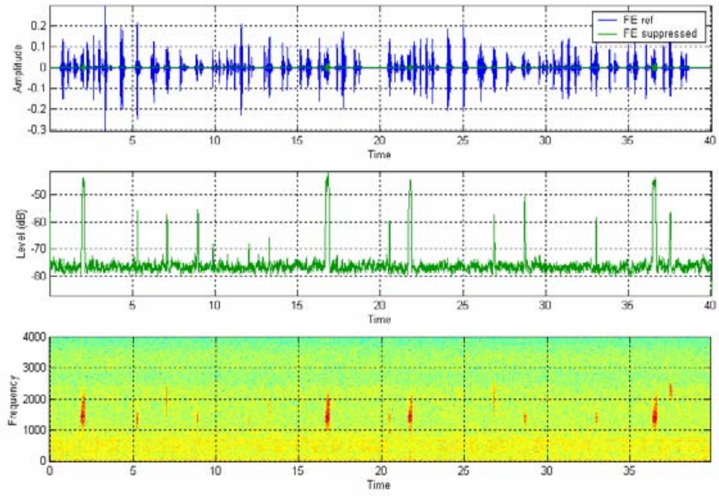
It is recommended only to change the gamma_e_high-parameter in extreme cases when the previous parameters are not sufficient to suppress all echoes.

### 5.4.4 Result

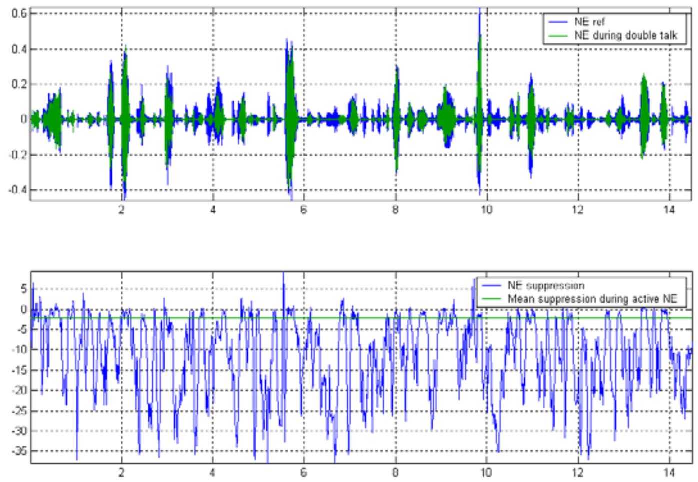
Following figures show the analysis of the echo suppression and double talk performance for two different settings of the tail-parameters. The analysis of the echo suppression consists of: the time-plot of echo-reference signal and suppressed echo signal, a level versus time plot of the transmitted signal, and the spectrogram of the transmitted signal.

Figure 19 shows an example of a bad tuning of the DES. In the level-versus-time plots glitches can easily be detected: they are 20dB higher than the noisefloor, so clearly audible. Also in the spectrogram the glitches can be detected: the glitches occur in a mid-frequency band 1000-2000 Hz. It can be tail echoes or perhaps 2$^{nd}$ or 3$^{rd}$ harmonic non-linear echoes. Figure 20 shows an example of a good tuning of the DES: no echoes can be detected nor in the level-versus-time plot, nor in the spectrogram.

The analysis of the double talk performance consists of a time-plot of the nearend-reference signal and the nearend signal during double talk and a nearend-suppression-versus-time plot. The green line represents the average suppression of the nearend during active periods of the nearend speaker. The nearend is less suppressed in Figure (a) than in figure (b), but this doesn't weigh against the bad echo suppression performance of (a).

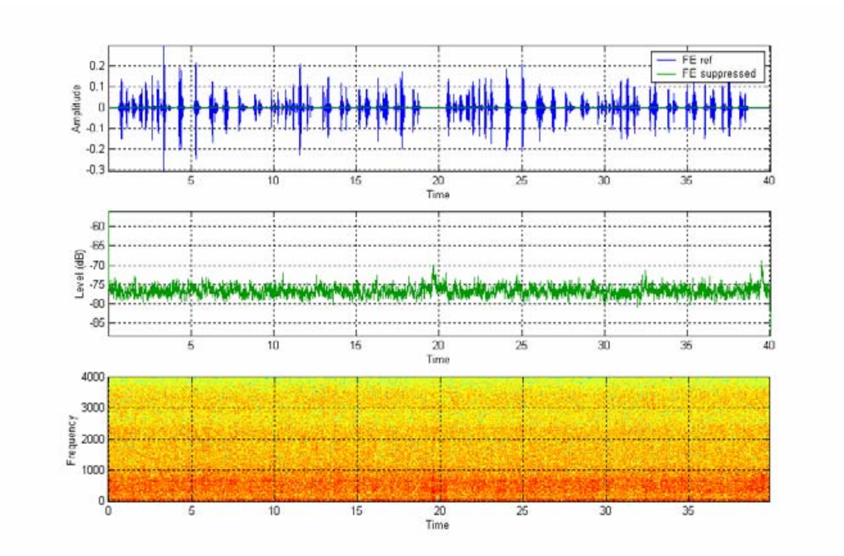



(a)

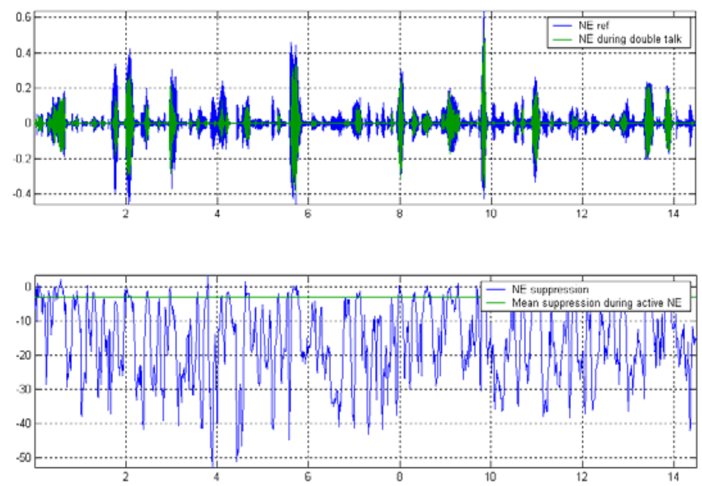



(b)

**Figure 19: Analysis of transmitted signal for tail-combination with unacceptable residual echoes. (a) Echo suppression analysis, (b) Double Talk analysis.**




(a)




(b)

**Figure 20: Analysis of transmitted signal for tail-combination with very good echo suppression. (a) Echo suppression analysis, (b) Double Talk analysis.**

## 5.5 Step 4: Determine Initial convergence-parameters

### 5.5.1 Background

The initial convergence time is determined by the stepsize of the NLMS-adaptive filter. A small stepsize has a positive impact on the stability of the filter but the convergence could suffer and as result initial echoes can occur. A large stepsize results in a very fast adapting filter but the drawback can be divergence during double talk and as result bad double talk performance. Therefore the choice of the stepsize must be based on a compromise between fast convergence (no initial echoes) and stability (good double talk).

Following two parameters influence the stepsize of the NLMS:

- **twoalpha**: normalized stepsize-parameter of the NLMS-algorithm. This parameter controls the overall adaptation speed of the NLMS. A low value leads to a slow but stable filter, a high value leads to a fast filter. A too high value for twoalpha can lead to oscillating convergence or even divergence.
- **ERL:** to prevent coefficient divergence during double talk the adaptation of the NLMS can be slowed down or stopped with an adaptive stepsize control. The ERL controls a threshold from where the adaptive stepsize control reacts. The needed value for the ERL-parameter depends on the acoustic coupling factor of the mobile: a large coupling factor indicates low ERL-value, small coupling factor indicates to a high ERL-value. Choosing a too low value for ERL can lead to a non-stable adaptive filter during double talk, while a too high value for ERL results in slow initial convergence which can lead to an unacceptable amount of initial echoes.

### 5.5.2 Test signal

Figure 21 depicts the test signal used for tuning the initial convergence (*tune_CONVERGENCE.wav*). It starts with three repetitions of an only farend sequence (male counting) interleaved with a 5-second pause, followed by two double talk sequences.

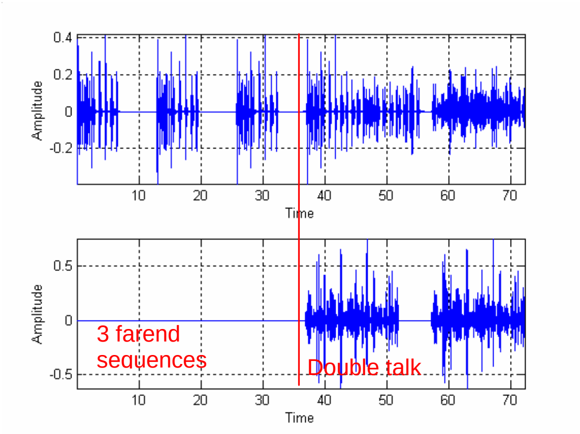



**Figure 21: Signal used for tuning the initial convergence**

### 5.5.3 Procedure

First of all a reference recording has to be made (AEC turned off!). The signal for this reference recording (*tune_CONVERGENCE_ref.wav*) is the same as the test signal depicted in Figure 21 except the double talk sequences are replaced by only-nearend sequences. This way the suppression of the nearend during double talk can be analyzed.

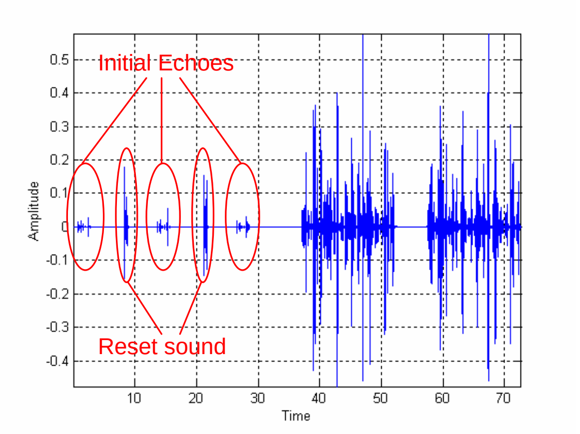
The AEC has to be reset three times: one time just before playing the test signal, and then in the pauses before the second and third farend sequence. With each reset, the NLMS-coefficients are reset to their initial values[2]. Initial echoes can then occur in the beginning of each farend-sequence.

Start with a low ERL-value for which there are no initial echoes. Increase the ERL until the amount of initial echoes is unacceptable or until the enhancement in double talk-performance becomes negligible. To illustrate what kind of initial echo is allowed, we refer to the VDA-specifications [2] where it is stated that the echo return loss should reach his nominal value after 1 second.

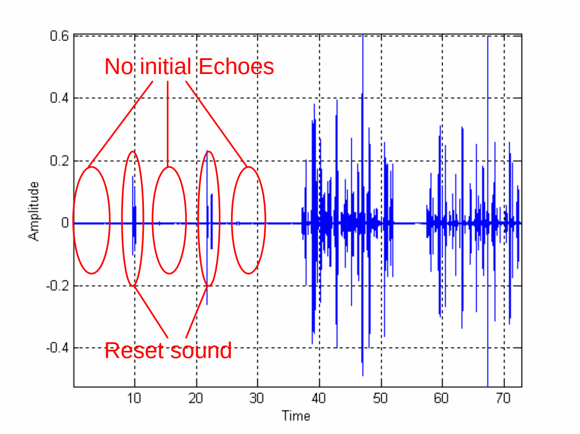
### 5.5.4 Result

Figure 22 shows the transmitted signal for two different ERL-settings. In Figure (a) a too high value was chosen for the ERL, resulting in initial echoes in the 2-3 seconds after each reset. The sound recorded during resetting the device must be ignored. In figure (b) the setting of the ERL was low enough to prevent initial echoes. The trade off can be seen clearly from these figures: the nearend is less attenuated during double talk for the setting with the initial echoes!

[2] Initial values of the NLMS depend on the NLMS_preset_coeff - parameter.




(a)




(b)

**Figure 22: Analysis of transmitted signal. (a) Unacceptable initial echoes, very good double talk; (b) good convergence behavior. worse double talk.**

## 5.6 Step 5: Determine CNI-parameters

### 5.6.1 Background

Oversubtraction of echo results in an extra suppression of the background noise. This effect is called noise-gating: the transmitted background noise is modulated by the farend speech. To prevent this effect, spectrally shaped comfort noise is injected.

The amount of comfort noise is controlled by the **CNI_level**-parameter. Ideally, the level of the inserted comfort noise should be equivalent in power level to the (suppressed) background noise. According to the VDA-specification a deviation of –5dB to +2dB is allowed [1].

### 5.6.2 Test signal

The complete test signal is depicted in Figure 23. The test signal in receiving direction consists of a short training sequence to adapt the NLMS-filter, a pause of 12 seconds and a periodical repetition of CSS-bursts (on-time of 250 ms) to enable comfort noise injection. In the meantime (after the training-sequence), 'car'-noise is played back via the nearend loudspeaker (or perhaps via other loudspeakers in the room to create a more diffuse sound).

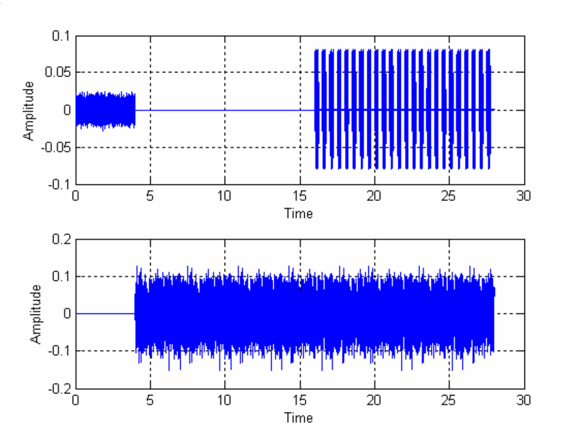



**Figure 23: Signal used for setting the level of comfort noise**

### 5.6.3 Procedure

The test signal described in previous paragraph has to be applied in the setup. The transmitted signal is recorded in sending direction. In the transmitted signal we can distinguish three different parts:

1. The original transmitted background noise: i.e. the part where the DENS didn't suppress the background noise yet;
2. The suppressed transmitted background noise: i.e. the part where the DENS suppressed the background noise (after 4 to 8 seconds);
3. The transmitted background noise during farend: i.e. the part where the CSS-bursts are applied and where perhaps noise-gating can be detected.

In each of these three parts the level of the transmitted signal has to be determined. First, the level of part 2 has to be referred to the level of part 1. This difference (difference 1) should normally be -10 dB (depending on the actual setting of the NS). Second, the level of part 3, during the application of a CSS-burst, has to be referred to the level of part 2. This difference (difference 2) should be within the range of -5 dB to +2 dB.

For tuning the CNI, start with a low CNI_level-value *CNIlow*, and increase it with *CNIstep* until difference 2 is just within the allowed range. While increasing, monitor difference 1 and make certain that this difference stays around the -10 dB (depending on the actual setting of the NS). This way, 'over-injection' of comfort noise is prevented.

### 5.6.4 Result

Figure 24 shows the level and the spectrogram of the transmitted background noise for three different CNI_level-settings. In the level-versus-time figures the three parts described above can be recognized. The level is calculated with a 40ms integration time. The spectrograms (or frequency-versus-time figures) are made only over the part with the farend bursts. The spectrograms are calculated with 1024-bins FFT over a window of 1024 samples with 50% overlap.

In figure (a) the CNI_level is set too low: the modulation of the background noise by the farend bursts is clearly noticeable. The noise level difference during a farend burst from the noise level without farend is about -8dB. The noise-gating effect is also clearly visible in the spectrogram. In figure (b) the CNI_level is increased until the noise level difference is –2dB, which is in the specified range of –5 to +2dB. The noise suppression is around 10dB. In the spectrogram can be seen that the spectrum of the inserted noise matches the spectrum of the original background noise. Figure (c) shows the noise level when the CNI_level is set too high: there is of course no noticeable noise-gating effect, but the CNI-module injects too much noise, resulting in a reduced suppression of the NS.

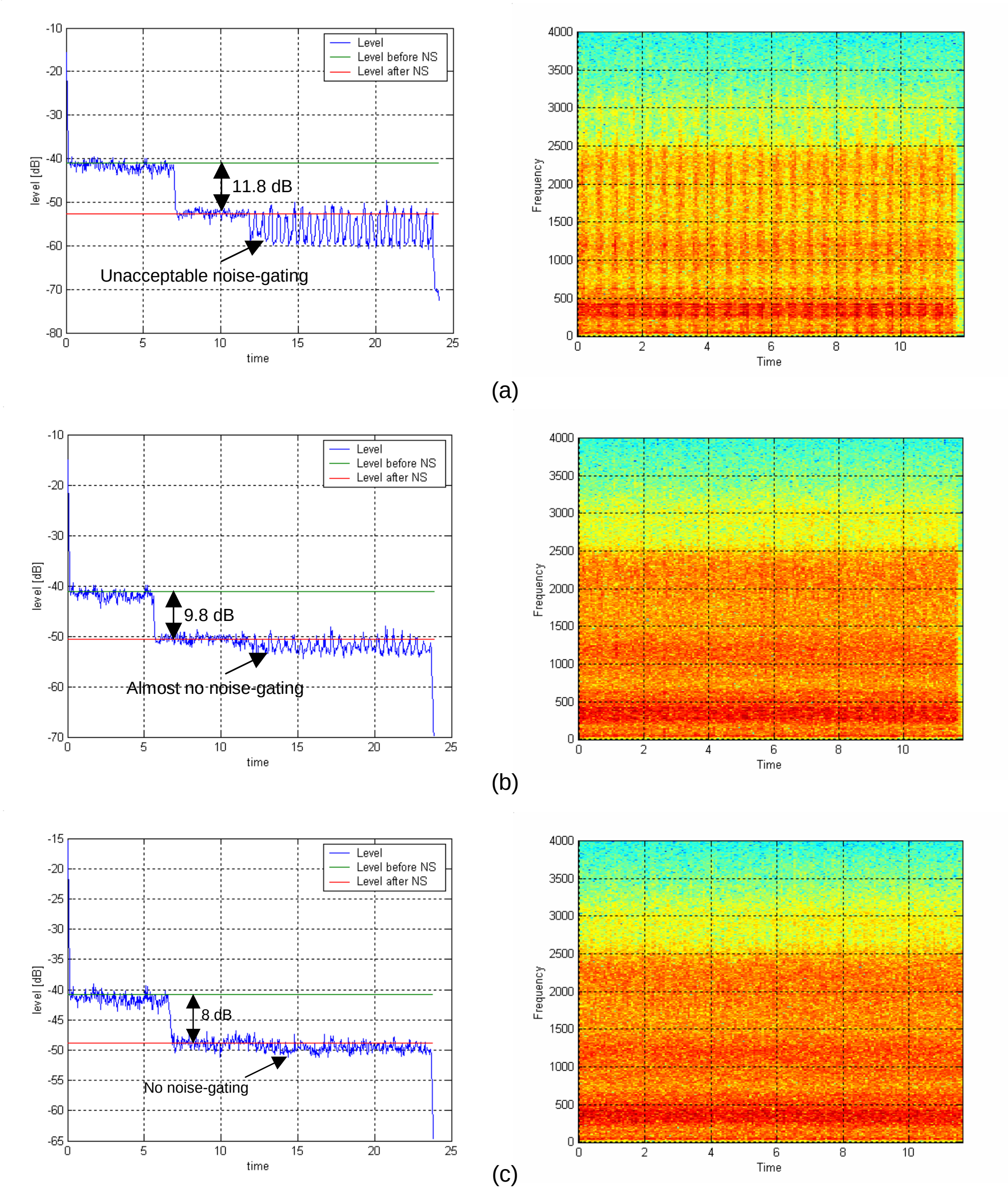



**Figure 24: Level and spectrogram of the transmitted background noise for three different CNI_level-settings: (a) Too low, (b) Good, (c) Too high.**

# 6 CONCLUSION

# 7 APPENDIX

## 7.1 Parameter ranges

Table 1 gives an overview of the parameter ranges that should be used in the tuning methodology together with the tuning step. To have a reference during tuning, Table 2 shows default parameter values (determined from tuning experience) for two different modes of the terminal: handsfree and handset.

**Table 1: Parameter ranges and tuning steps**

| Parameter | Min | Max | Step |
|---|---|---|---|
| *N* | 32 | 160 | 16 |
| *tail_alpha* | 20000 | 26000 | 1000 |
| *tail_portion* | 2000 | 12000 | 1000 |
| *NL_atten* | 0 | 1024 | 128 |
| *xclip* | 1 | 1 | 0 |
| *gammae_high* | 512 | 1024 | 128 |
| *CNI_level* | 0 | 16000 | 1000 |

**Table 2: Default parameters for handsfree and handset**

| Parameter | Default handsfree | Default handset |
|---|---|---|
| *N* | 96 | 40 |
| *ERL* | 300 | 900 |
| *twoalpha* | 8192 | 8192 |
| *tail_alpha* | 23000 | 19600 |
| *tail_portion* | 6000 | 6000 |
| *NL_atten* | 512 | 0 |
| *xclip* | 1 | 1 |
| *gammae_high* | 512 | 512 |
| *CNI_level* | 12000 | 12000 |

## 7.2 Tuning overview

Following table gives an overview of the test signals and the tuning target for each tuning step. The marked tuning steps require a reference recording to be done.

**Table 3: Overview of test signals and tuning target**

| Tuning step | Test signal | Target |
|---|---|---|
| Level calibration rcv | level_rcv.wav | 82 dB SPL |
| Level calibration snd | level_snd.wav | 82 dB SPL |
| Soundcard calibration | level_rcv.wav | Equal I/O-level |
| Check Gains | check_micgain.wav | Farend limited instead of clipping echo |
| Tune NLMS | tune_NLMS.wav | > 15 dB Echosuppression |
| Tune DENS | tune_DENS.wav | No echo, best double talk |
| Initial convergence | tune_DENS_init.wav | Trade off initial echoes with double talk |
| CNI_level | tune_CNI.wav | No Noise gating, correct NS |

# GLOSSARY

| | |
|---|---|
| AEC | Acoustic Echo Canceller |
| NS | Noise Suppression |
| CNI | Comfort Noise Injection |
| RF | Radio Frequency |
| MS | Mobile Station |
| DTX | Discontinuous Transmission |
| Tx | Transmit |
| Rx | Receive |
| CSS | Composite Source Signal |
| NLMS | Normalized Least Mean Squares |
| DENS | Dynamic Echo and Noise Suppressor |
| DES | Dynamic Echo Suppressor |

# REFERENCES

[1] *LifeVibes™ Handsfree User Manual*
[2] *VDA Specification for Car Hands-free Terminals*

## End of document