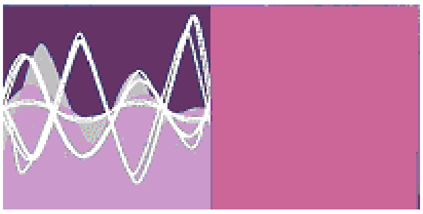

LifeVibes™ Hands-Free Sysol 6000

User Manual

User Manual

Document ID: SER-04121

Version: 1.4

1 INTRODUCTION 3

2 LIFEVIBES HANDS FREE PARAMETER DESCRIPTION 4
2.1 Product overview 4
2.2 Pre-processing 4
2.3 Delay 5
2.4 NLMS 6
2.5 Post Processing Echo Reduction 7
2.6 Post Processing Noise Reduction 8
2.7 Mode 8
2.8 Parameter range 9

1 INTRODUCTION

This manual describes how to tune LifeVibes™ Hands Free. LifeVibes™ Hands Free performs a full-duplex acoustic echo cancellation (AEC) for hands-free operation, e.g, the mobile is put on a table and the loudspeaker volume is turned up. LifeVibes™ Hands Free can be used even in a normal operation, where the handset is held near the ears, in headset and in car kit operation. LifeVibes™ Hands Free includes a stationary background noise reduction.

2 LIFEVIBES HANDS FREE PARAMETER DESCRIPTION

2.1 Product overview

The building blocks are shown in Figure 1.

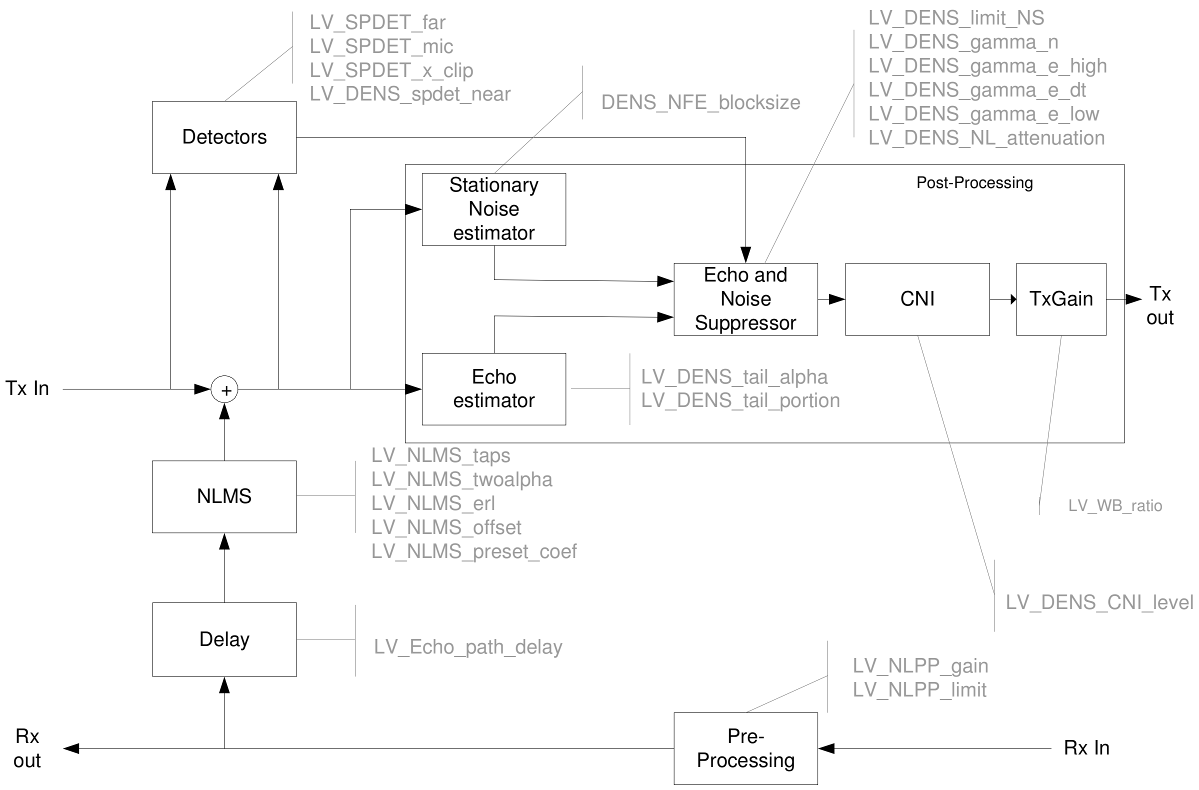

Figure 1: Building blocks and their parameters

2.2 Pre-processing

Figure 2 shows the integration of LifeVibes™ Hands Free in the audio path. To achieve full duplex capability, the echo path must be linear as much as possible. The audio path is defined as the data path from the output of the LVHF_rx to the LVHF_tx including the digital amplifier, AD/DA, loudspeaker, microphone, microphone amplifier and acoustic echo path.

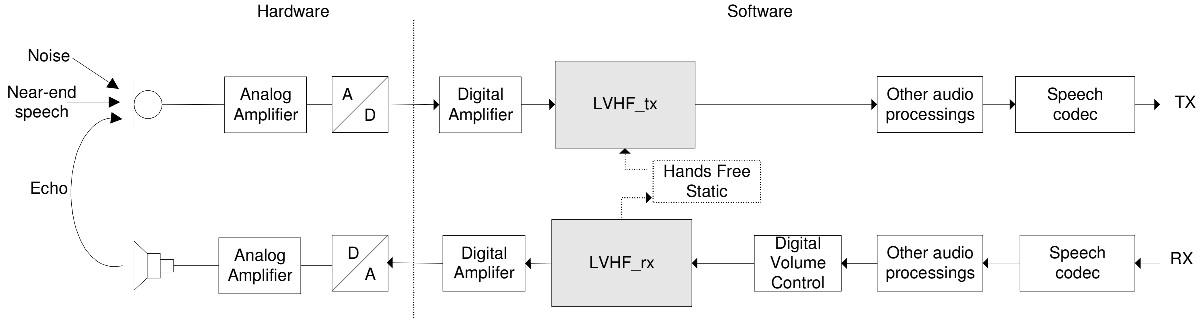

Figure 2: LifeVibes™ Hands Free in the audio path.

The pre-processing module takes care that digital and analog amplifiers are working in their linear range.

The loudspeaker digital amplifier should be set to a value where the DA works in its linear range and as close as possible to its full scale. The analog gain should works in its linear gain. Therefore, the digital and analog gain might not set to their maximum values. To compensate this loss of gain and achieve the requested SPL, the LifeVibes™ Hands Free pre-processing implements a gain and a clipper.

The pre-processing amplifies the input signal by a factor of *LV_NLPP_gain* and clips it at the *LV_NLPP_limit* value. The clipping value is driven by a trade-off between:

- Speech distortion. The clipping value should be near to 1 to avoid clipping of the Rx signal.
- Clipping in the analog and digital amplifiers.

The *LV_NLPP_gain* value is determined by the desired requested SPL considering the setting of the digital and analog amplifiers.

$$LV\_NLPP\_gain = 32768 * 10^{\frac{Gain[dB]-24[dB]}{20}}$$

In handset and headset mode, the digital and analog amplifiers are generally set too low to clip the Rx signal. So, for those 2 modes the pre-processing is generally not activated. For loudspeaker mode, to produce high sound pressure level out of the loudspeaker, high amplification is required and the pre-processing generally must be activated.

To have the proper effect for the pre-processing, the input of the pre-processing should be filtered by a high pass filter so that the path from the input pre-processing to the loudspeaker is not differentiating.

It is recommended to insert the volume control in the Rx path before the LVHF_rx module. So that LVHF_rx takes as input the proper scaled signal and avoid some signal clipping and improve the voice quality for low signal.

2.3 Delay

This delay compensates the fixed delay in the audio path (Figure 2) and can be measured in advance. This delay is caused by the audio I/O buffering, AD/DA converter and propagation time between the loudspeaker and microphone.

LV_Echo_Path_Delay must be set on the safe side to be sure that the direct path can be measured by the NLMS module.

The unit of *LV_Echo_Path_Delay* is sample at 8kHz.

2.4 NLMS

The NLMS module models part of the acoustic path between Rx_{out} and Tx_{in} of LifeVibes™ Hands Free. A FIR filter is used for the modelization. Figure 3 shows a typical example of impulse response. As the first 64 coefficients contains most of the energy of the impulse response. Increasing the filter length will:

- Increase MIPS budget without improving performance,
- Increase the convergence time of the adaptive filter.

For speakerphone and handset modes, it is recommended to set the filter length (*LV_NLMS_taps*) to 64 taps.

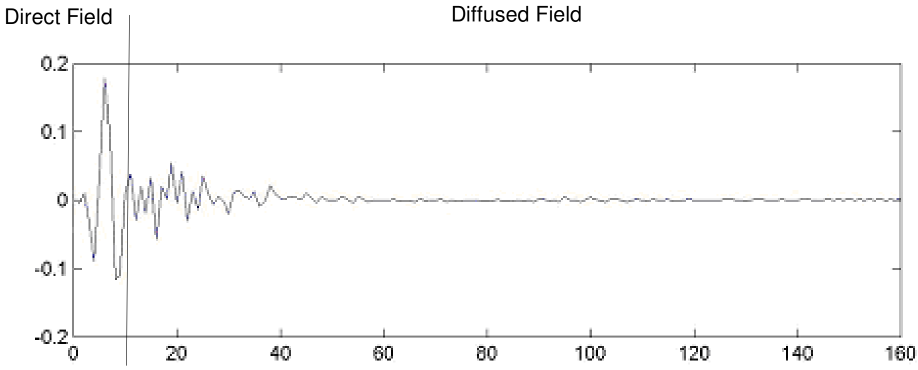

Figure 3: Impulse response of the echo path

The adaptation speed of the NLMS is controlled by the following parameters:

- *LV_NLMS_twoalpha* is the step size to update the coefficients of the adaptive filter. It is the trade-off between convergence speed/tracking of acoustic changes and stability of the adaptive filter. High value leads to a fast speed of the adaptation. A low value leads to a slow speed adaptation but a stable adaptive filter.
- *LV_NLMS_erl*: To avoid divergence during double talk, the NLMS module uses a simple step size control which slow down the adaptation speed when P_{xx}<ERL P_{rr}. Here P_{rr} is the estimated power in the residue signal, P_{xx} is the estimated power in the incoming Rx signal after far-end pre-processing and ERL is the inverse of the coupling factor. The coupling factor is the ratio between the average power of the Rx signal and the average of the echo in the microphone signal. The physical ERL value can be measured knowing the Rx and microphone signal. Real physical value of the ERL could leads to a non-stable adaptive filter during double talk. This is especially the case in speakerphone mode. Choosing a higher ERL value has positive impact on the stability of the solution. Such choice will improve the quality of the near–end speech during double talk. Only the initial convergence could suffer from the increase of the ERL.

$$LV\_NLMS\_erl = 64 \cdot 10^{\frac{ERL[dB]}{20}}$$

- *LV_NLMS_offset*: It is an offset added to the computation of the power P_{xx} of incoming Rx signal to avoid that this P_{xx} converges toward zero when Rx is low. The main reason is that the step size *LV_NLMS_alpha* is normalized by the P_{xx} power.

- *LV_NLMS_preset_coefs*: Indicates whether a preset of NLMS coefficients should be done. Setting to 0 means no preset, 1 means preset with certain non-zero values, 2 means preset with zeros.

2.5 Post Processing Echo Reduction

The adaptive filter estimated by the NLMS module cannot cancel 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.

The post processing echo reduction removes such remaining echoes.

The post-processing is controlled by the following parameters:

- *LV_DENS_Tail_alpha* is related to the reverberation time RT60 of the acoustic environment. This RT60 can't really be related to the RT60 of a room because there is not enough SPL produced by mobile phone to excite the acoustic of a room. Moreover most of the coupling is coming from the internal mechanics of the mobile phone itself. The RT60 can be measured from the Energy Decay Curve: $EDC(k) = 10.\log\left(\sum_{i=k}^{\infty} h^2(i)\right)$) of the impulse response h [a]. The slope of the linear part of the EDC is estimated and provides the $RT_{60} = \frac{60}{|Slope|F_s}$ (Fs is the sampling frequency)

$$LV\_DENS\_Tail\_alpha = 32768.10^{\frac{-0.03}{RT_{60}}}$$

- *LV_DENS_Tail_portion* corresponds to the ratio of the echo power estimated with the NLMS adaptive filter to the power of the tail echo produced by the next 80 samples of the impulse response. With N=*LV_NLMS_taps*:

$$LV\_DENS\_Tail\_portion = 32768\frac{\sqrt{\sum_{n=N}^{N+79} h^2[n]}}{LV\_DENS\_tail\_alpha\sqrt{\sum_{n=0}^{N-1} h^2[n]}}$$

  To check is *LV_DENS_Tail_portion* is set properly, switch off all processing except NLMS and DES (mode=3), use a low loudspeaker volume to avoid non-linearity, speak and listen at the far-end: if only late echoes and no early echoes are audible, increase *LV_DENS_Tail_portion*. This parameter should no be set to high in order to keep undistorted the voice in double talk.

[a] The impulse response from the loudspeaker to the microphone can be measured by playing a signal (preferable a pink noise) through the loudspeaker and recorded by the microphone. The impulse response is the ratio of the FFT of the microphne signal over the FFT of the loudspeaker signal.

If there is still echo with a high value of *LV_DENS_Tail_portion*, then increase *LV_DENS_Tail_alpha.*

- To customize the amount of echo in the different scenarios (far-end only, double talk, near-end only), there are 3 gains, which are applied on the estimated echoes in the post processing:
  - *LV_DENS_gamma_e_high*: gain applied during far-end only
  - *LV_DENS_gamma_e_dt*: gain applied during double talk
  - *LV_DENS_gamma_e_low*: gain applied during near-end only. Avoid unnecessary echo suppression caused by far-end background noise.

The switch between these different parameters relies on the detectors (*LV_SPDET_far*, *LV_DENS_spdet_near*, *LV_SPDET_mic*), which are rather difficult to tune for all kind of potential scenarios. It is recommended not to play too much with those parameters and keep them as close as possible to each other. It will avoid some switch effects.

- After tuning all previous parameters, if there are echoes remaining due to non-linearity in the audio path, then the non-linear echo mechanism should be tuned. It is controlled by the 2 following parameters:
  - LV_SPDET_x_clip: It is set to the digital level of the loudspeaker output signal, where the echo path has some distortion.
  - LV_DENS_NL_atten: Sets the amount of extra non-linear suppression.

2.6 Post Processing Noise Reduction

The post-processing module includes stationary noise suppression. The stationary component of the noise is estimated over a window of *LV_DENS_NFE_Blocksize* frame. With *LV_DENS_limit_ns* the maximum amount of noise can be set:

$$LV\_DENS\_limit\_ns = 32768.10^{-\frac{\max noise\ \sup pression[dB]}{20}}$$

With *LV_DENS_gamma_n*, the sensitivity of the noise suppressor can be chosen.

10dB of noise suppression is recommended to avoid speech distortion.

To avoid noise-gating effect, comfort noise is implemented. The amount of comfort noise is controlled by the *LV_CNI_Level* parameter.

2.7 Mode

The mode word allows switching off functionality of Hands Free.

Mode bit	Functionality on or off
0	NLMS adaptive filter
1	Post processing echo suppression
2	Post processing noise suppression
3	Comfort Noise insertion
4	Post processing non linear echo suppression
5	High-band mute/active
6	High-band variable attenuation off/active
7	Pre-processing of Rx

Table 1: mode word

2.8 Parameter range

Parameter name	Min.	Max	Default handset	Default hands free	Default headset
LV_mode	0	0xFF	0x0F	0x8F	0x0F
LV_echo_path	0	0x10E	0xFE	0xFE	0xFE
LV_NLMS_taps[a]	0x000C	0xC8	0x40	0x40	0x01
LV_twoalpha[a]	0	0x7FFF	0x1999	0x1999	0x1999
LV_NLMS_erl[a]	0	0x7FFF	0x3F4	0x3F4	0x3F4
LV_NLMS_preset_coef[a]	0	2	1	1	1
LV_NLMS_offset[a]	0	0x7FFF	0x148	0x148	0x148
LV_SPDET_far[a]	0	0x7FFF	0x4000	0x4000	0x4000
LV_SPDET_mic[a]	0	0x7FFF	0x7333	0x7333	0x7333
LV_SPEDET_x_clip[a]	0	0x7FFF	0x7FFF	0x7FFF	0x7FFF
LV_DENS_spdet_near[a]	0	0x7FFF	0x200	0x200	0x200
LV_DENS_tail_alpha[a]	0	0x7FFF	0x4CCC	0x4CCC	0x4CCC
LV_DENS_tail_portion[a]	0	0x7FFF	0x147	0x147	0x147
LV_DENS_gamma_e_high	0	0x7FFF	0x200	0x200	0x200
LV_DENS_gamma_e_low	0	0x7FFF	0x100	0x100	0x100
LV_DENS_gamma_e_dt	0	0x7FFF	0x100	0x100	0x100
LV_DENS_gamma_n	0	0x7FFF	0x1E6	0x1E6	0x1E6
LV_DENS_NFE_blocksize[a]	0	0x7FFF	0x100	0x100	0x100
LV_DENS_limit_NS[a]	0	0x7FFF	0x2AAA	0x2AAA	0x2AAA
LV_DENS_NL_atten[a]	0	0x7FFF	0x800	0x800	0x800
LV_DENS_CNI_level[a]	0	0x7FFF	0x2000	0x2000	0x2000
LV_NLPP_gain	0	0x7FFF	0x800	0x800	0x800
LV_NLPP_limit	0	0x7FFF	0x7FFF	0x7FFF	0x7FFF
*LV_*WB_ratio	0	0x7FFF			

Table 2: Allow range and typical value of Hands Free parameters

End of document

[a] It is recommended not to change this value.