Part IV: Applications

1 ADAPTIVE ECHO CANCELLER (AEC)

The performance of AEC heavily depends on implementation details, and requires a very large set of test vectors.

An AEC for a smartphone, with rattling overdriven loudspeaker mechanically coupled to microphone, is very different from a high-end conferencing room AEC with physically separated premium loudspeakers and microphone arrays. An AEC for an ADI Shark DSP, using internal memory only, is very different from an AEC for Intel Core CPU supporting AVX-512, or ARM's NEON. Typically, the core of AEC is implemented in assembly, etc.

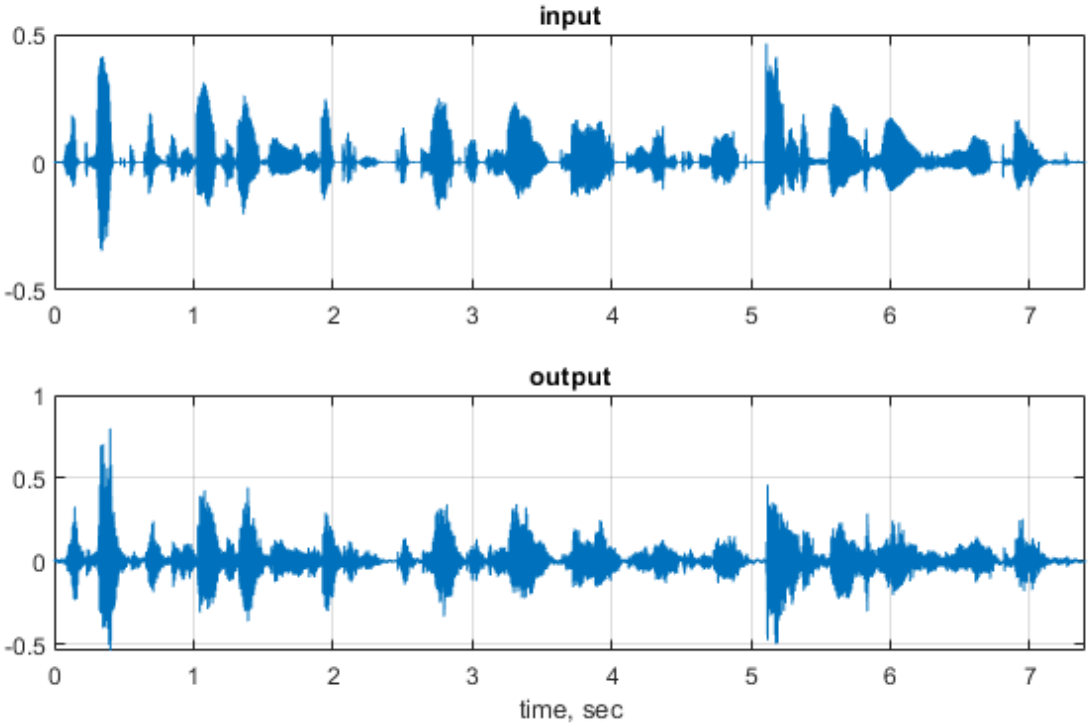
It may look as if an FSAF AEC shall be use-case built and developed entirely in low-level languages. However, there are quite a few common checkpoints which could be adequately addressed with MATLAB.

Disclaimer: the comparisons with SAF1988 can NOT be used in any way, nor considered as negative, in any sense or meaning, remarks towards Dr. W. Kellerman and his works.

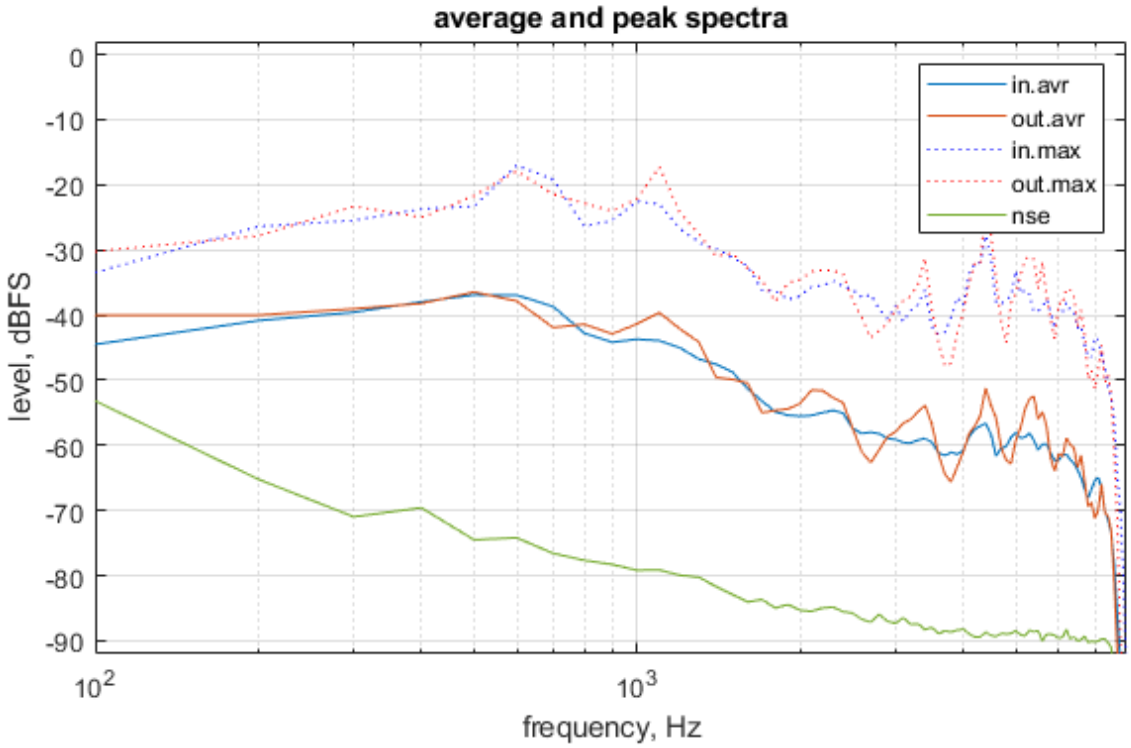
1.1 BASICS

Let's demonstrate what works (and what does not) on the following reference example:

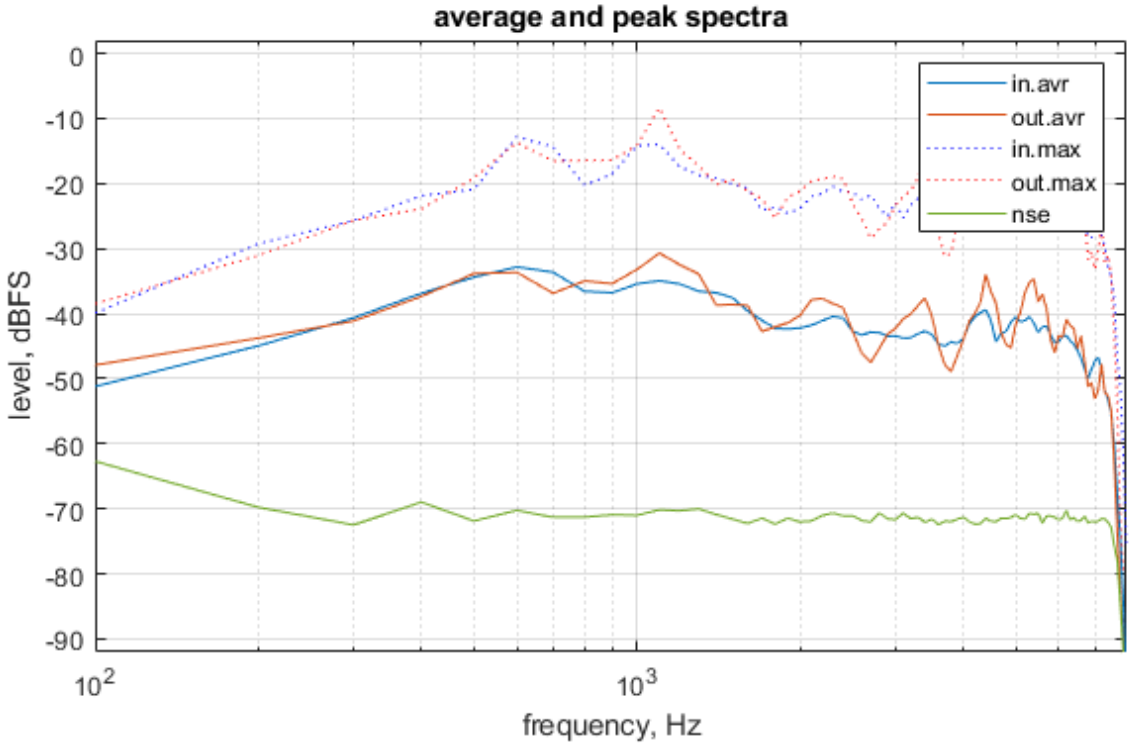
- pre recorded male speech for ~7sec, 3 of so-called Harvard sentences.
- Scarlett 2i2 Gen 3 as audio interface, ASIO 96kHz, buffer of 1024 samples
- A decent loudspeaker with low non-linear distortions, calibrated to provide 60dBA at listener's position on -20dBFS (re sine wave) input.
- Not too noisy microphone: AKG P420 in cardioid mode
- In an acoustically treated living room - which turned out to have uniformly distributed RT_{60} of less than 250ms.
- With usual 40dBA city noise behind the windows.



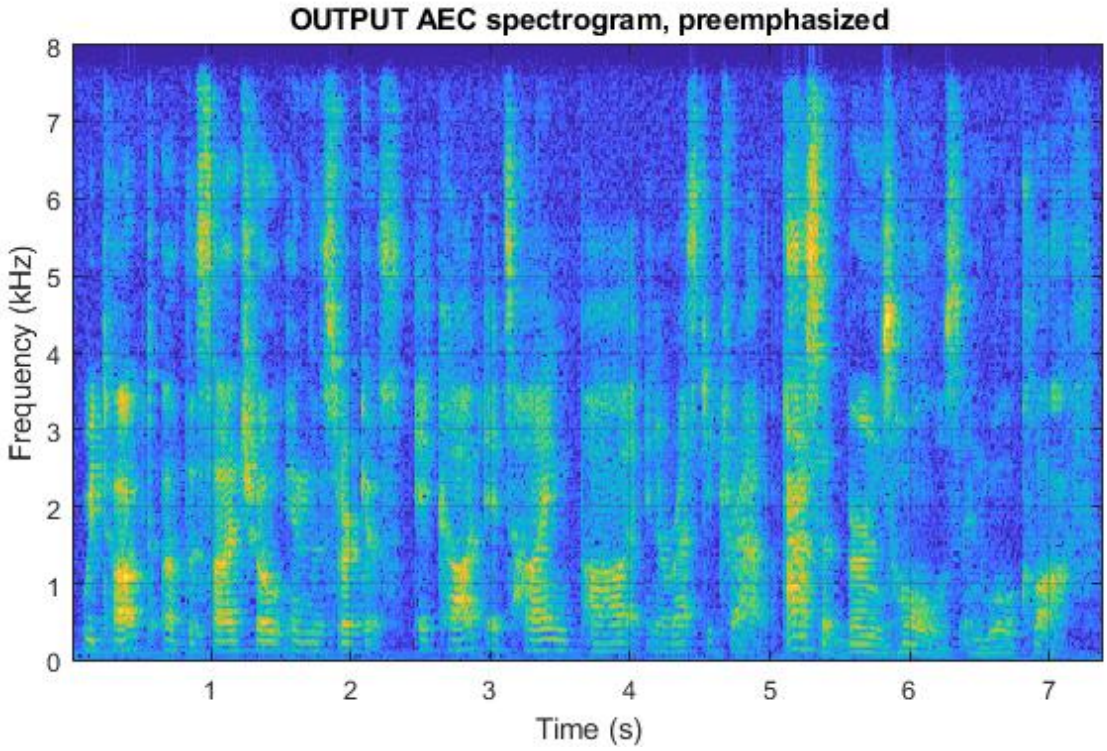
The spectra show that the room noise is dominated by microphone preamp *1/f* noise:



The voice is also 1/f after 500Hz, so the first signal processing operation we do is a spectral flattening with a 1st order IIR filter, the same for both IN and OUT, to be reversed by inverted IIR on RES.



Now we can apply subband decomposition with less fear that aliased powerful low frequency noise may spill over weak high-frequency components. We use 80 bands for 16kHz sampled signals, with long (L=16) asymmetric LD (latency=5) filterbank.

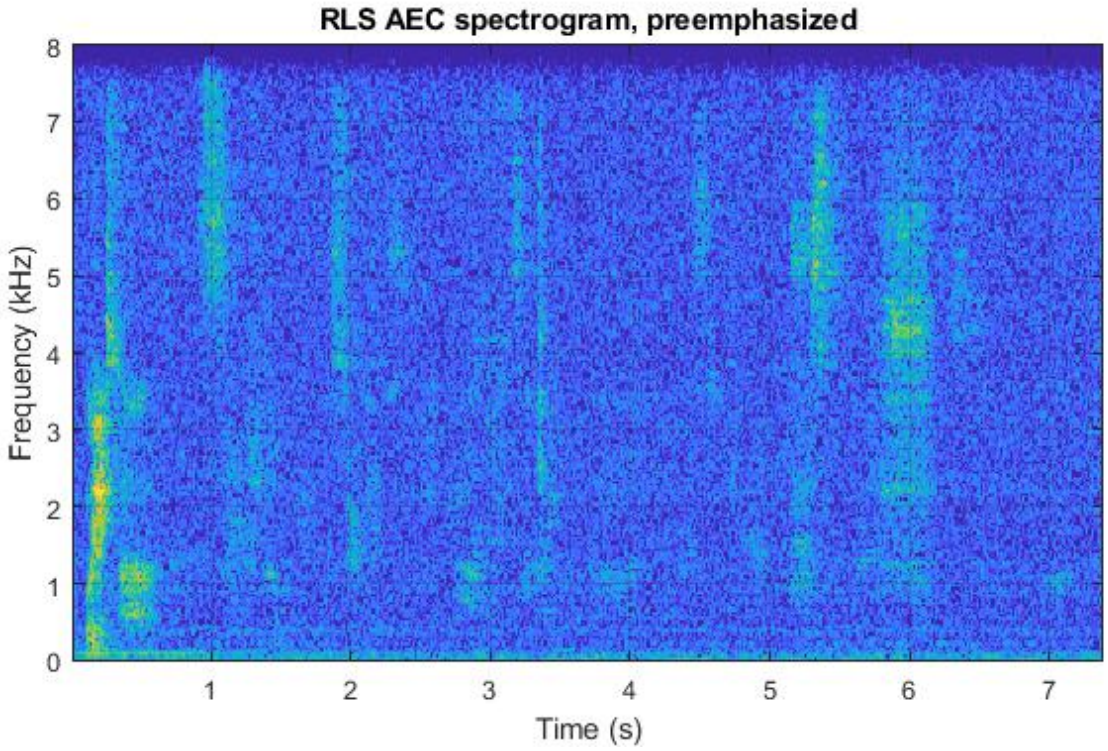


Here we see formant structure, voiced and unvoiced sounds, pauses before stop consonants (plosives), on a background of white noise, with average SNR of 30…35dB, peaking to ~60dB.

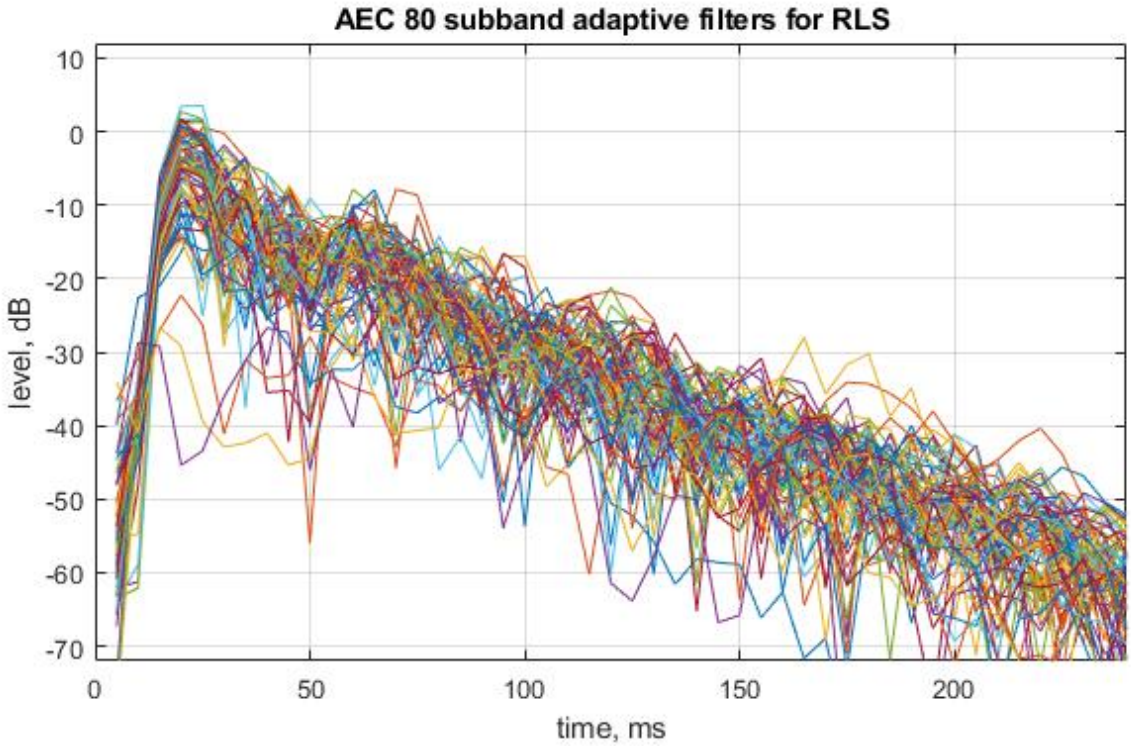
1.2 BIRD'S-EYE VIEW [401]

1.2.1 ReRLS

First, we check what can be achieved with sub-band JTF-RLS, the fastest of available adaptive algorithms:

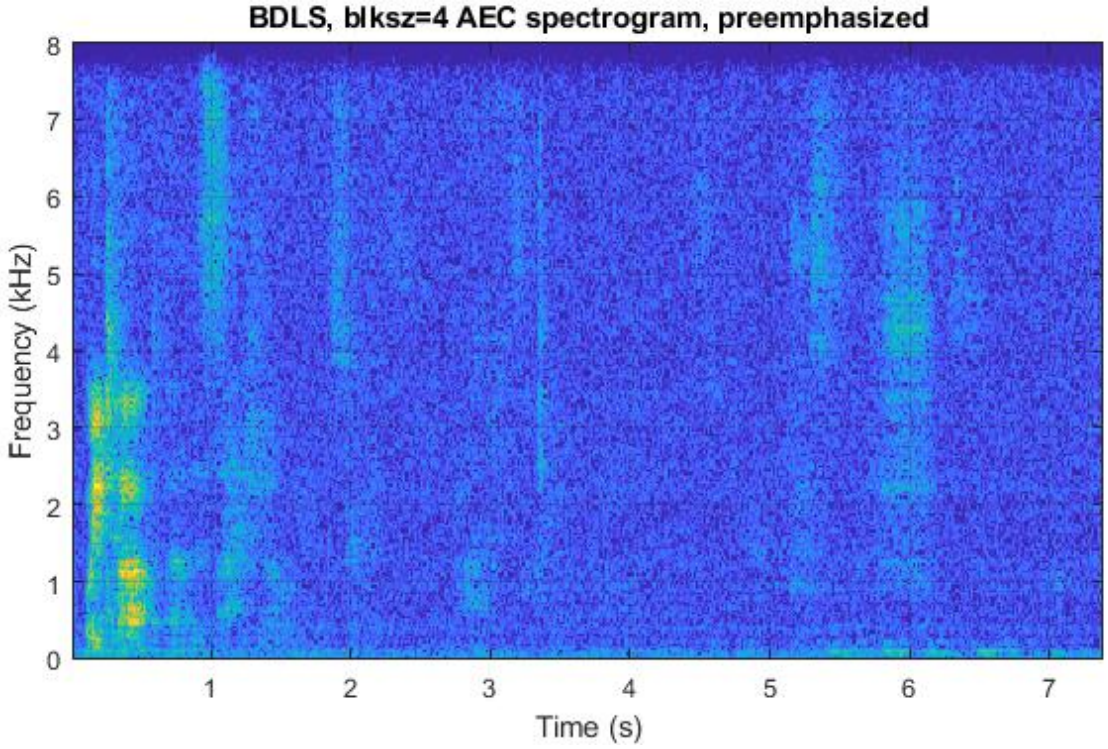


We see that RLS is able to converge perfectly within the first world, by about 0.6 seconds (for all subbands where the signal with sufficient SNR is present). The adaptive filters, with 5ms of extra delay, and 240ms of echo tail capacity, are clean:

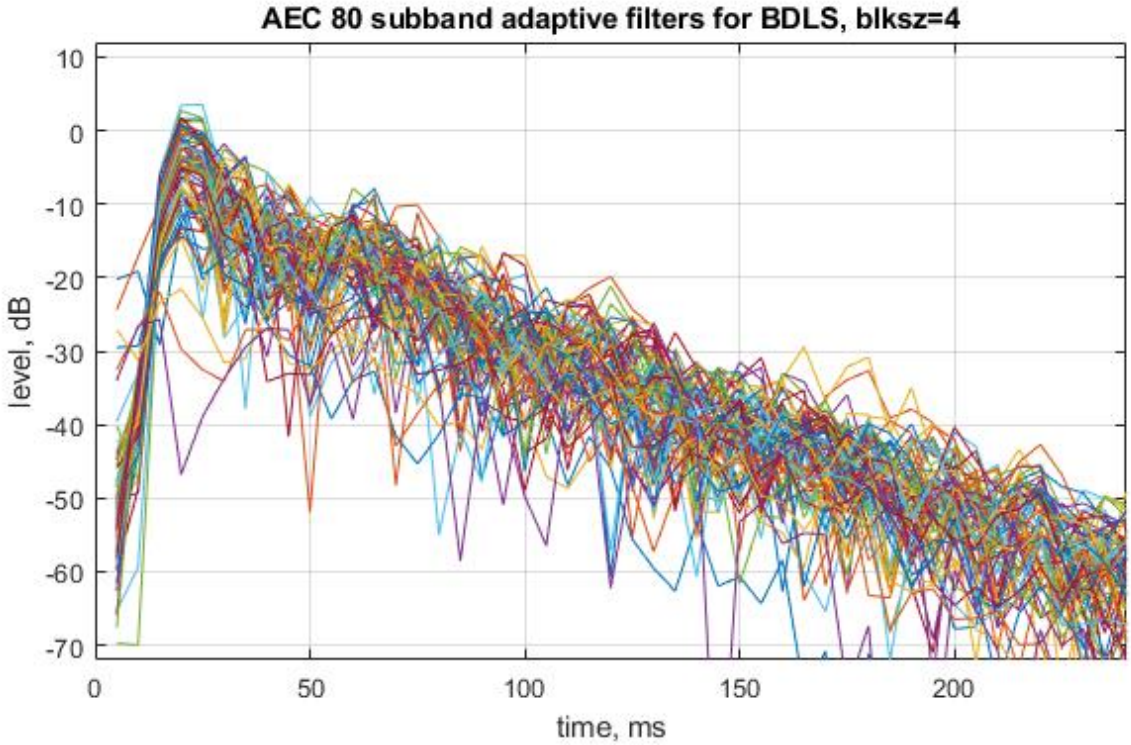


1.2.2 BDLS, block size of 4 frames, relaxation 0.7

The next algorithm is block DLS, with short blocks (~25% performance/memory overhead above LMS):

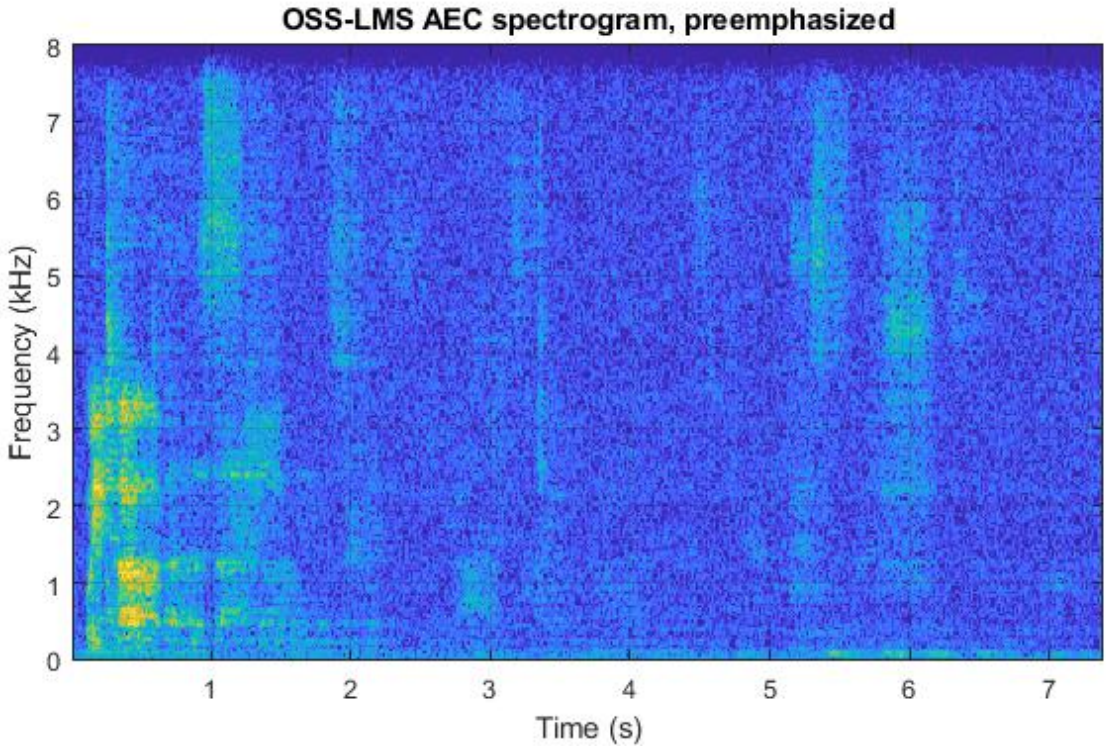


The convergence is slower but not considerably. The adaptive filters are also mostly clean, except for the lowest frequencies, which is noticeable at the first tap:

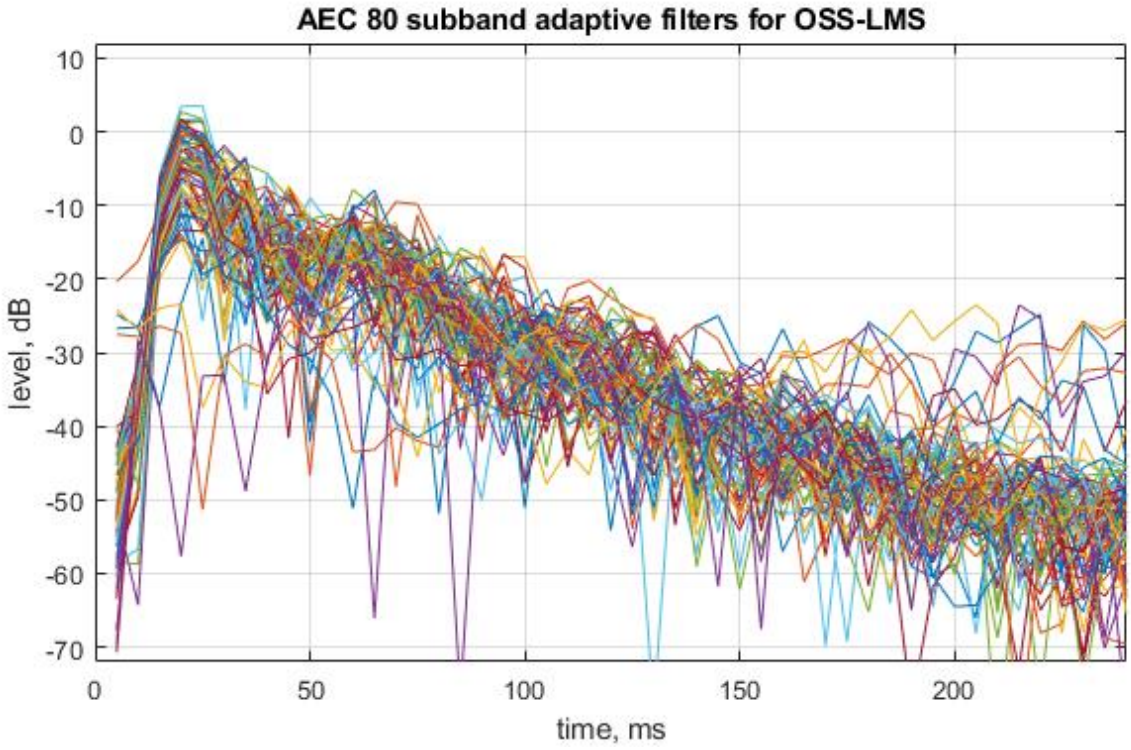


1.2.3 OSS-LMS by Kulczycky (1981)

This algorithm is THE reference for all scalar step size LMS algorithms, the fastest and cleanest of anything in the class. The potentially achievable performance is limited by:

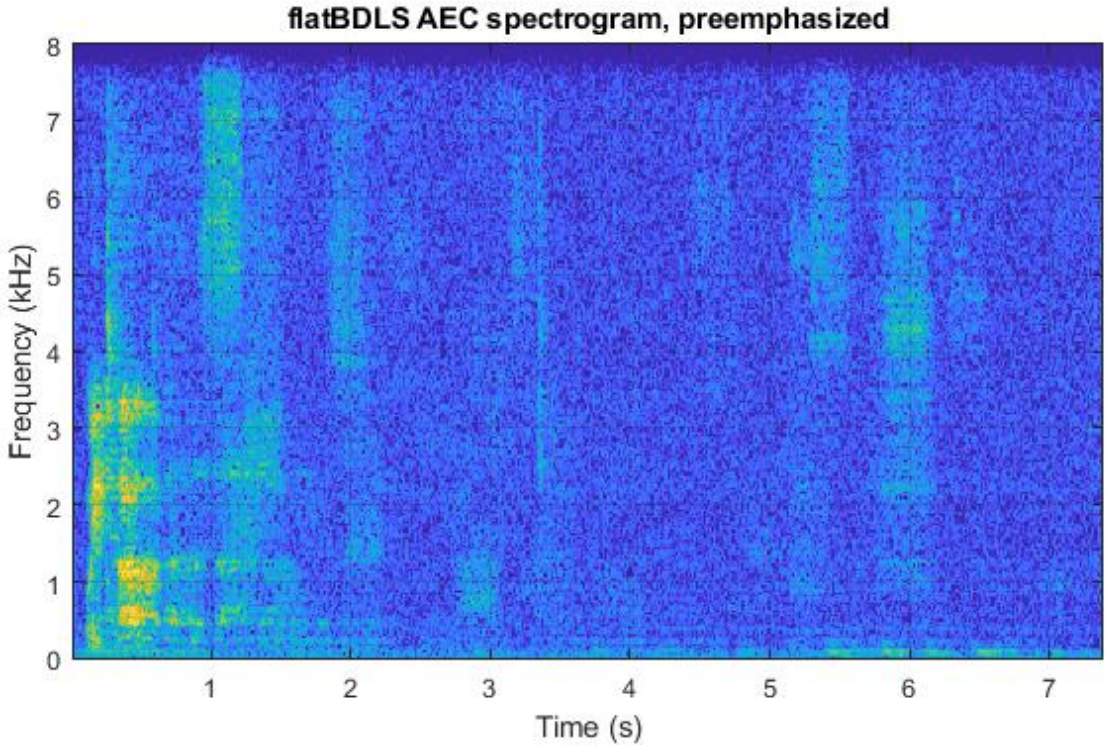


It's somewhat worse than the previous exponential BDLS but still well above disagreeable. The adaptive tails start to deviate, uncapable to converge down:

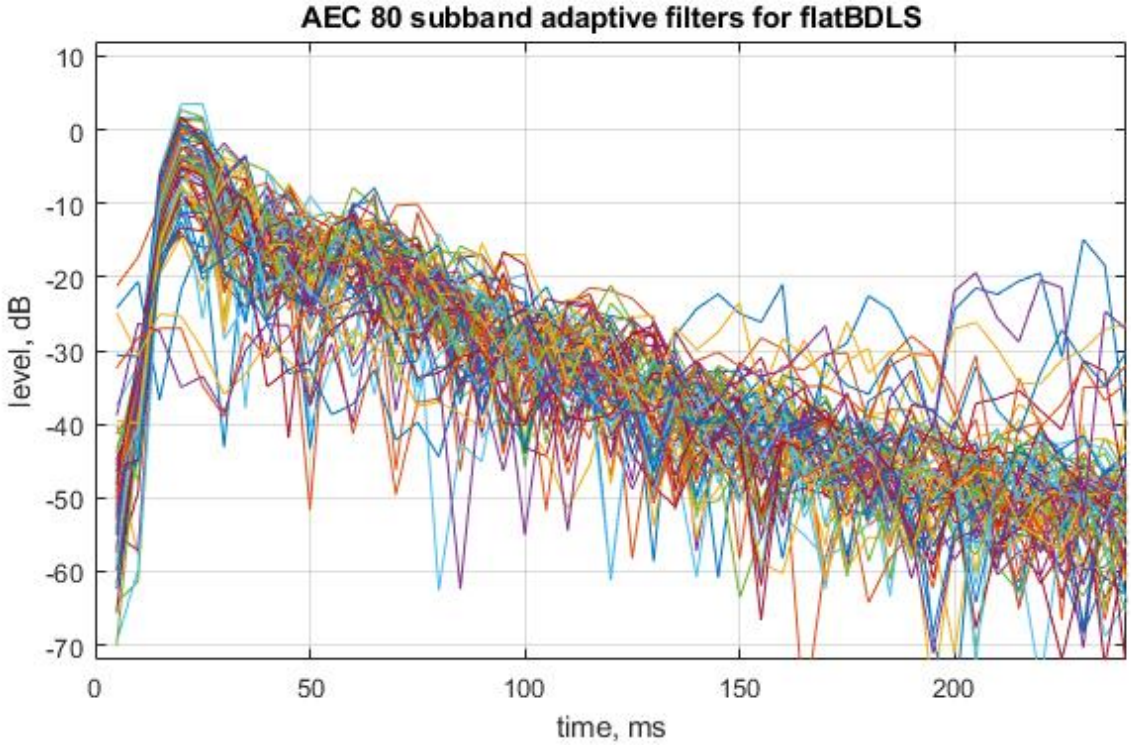


1.2.4 Sub-OSS LMS by flat BDLS, relaxation 0.7

The OSS-LMS can be approximated by flat BDLS. The quality of approximation shall be judged by the step size closely following the OSS-LMS's step size "from below". Otherwise, some unpredictable degree of divergence may occur. The correlated input is accounted for by 0.7 relaxation parameter.

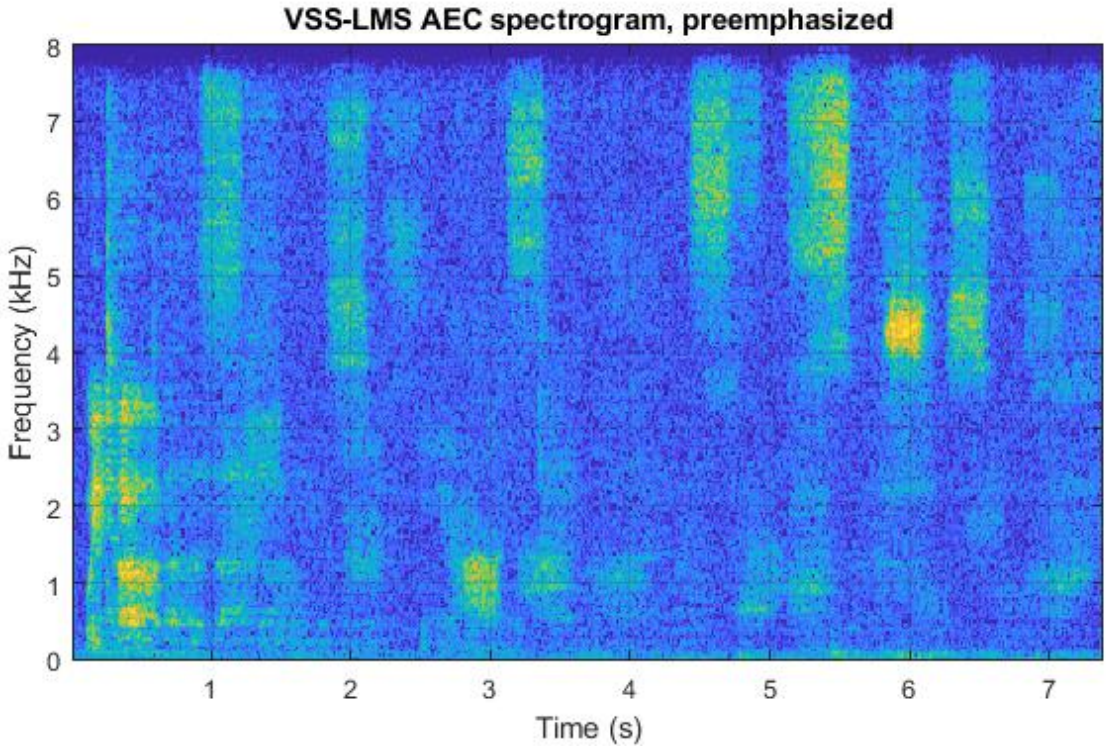


The decaying performance and echo tails are worse but still similar to the previous example:

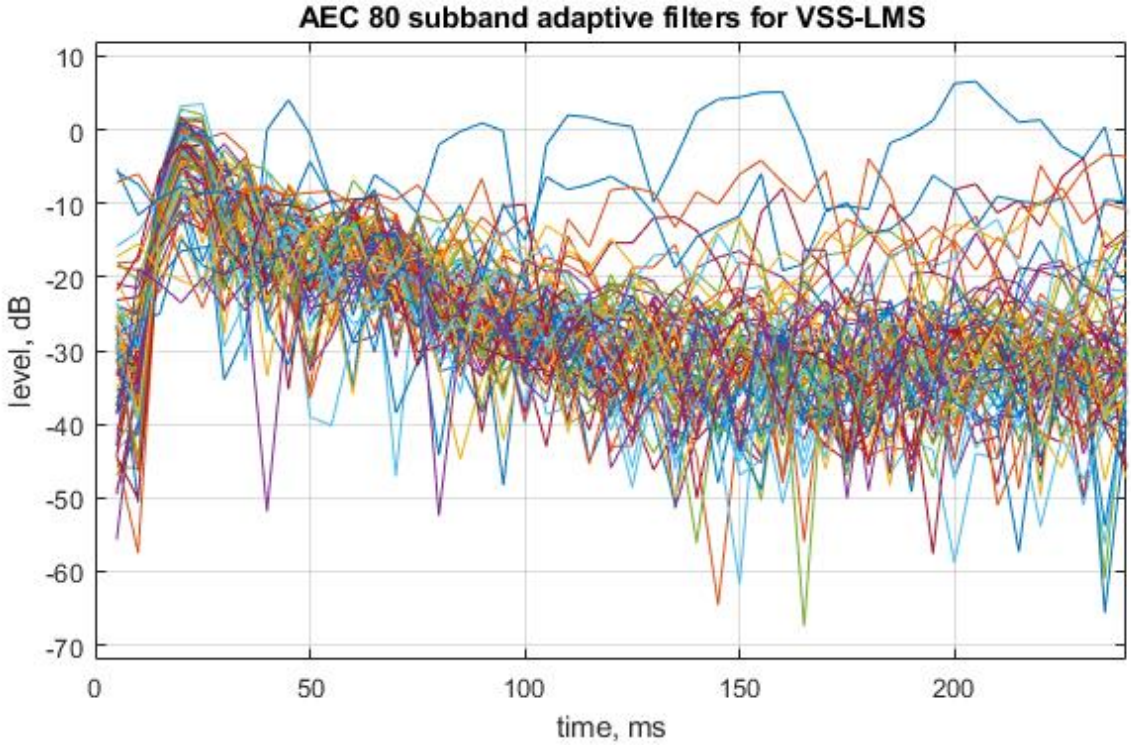


1.2.5 VSS-LMS by Benesty et al (2006)

Now we move to other, empiric step size control algorithms, based on the intuition of respective authors. One of the best (imho) was proposed by Benesty et al in 2006 (the idea is in the right direction):



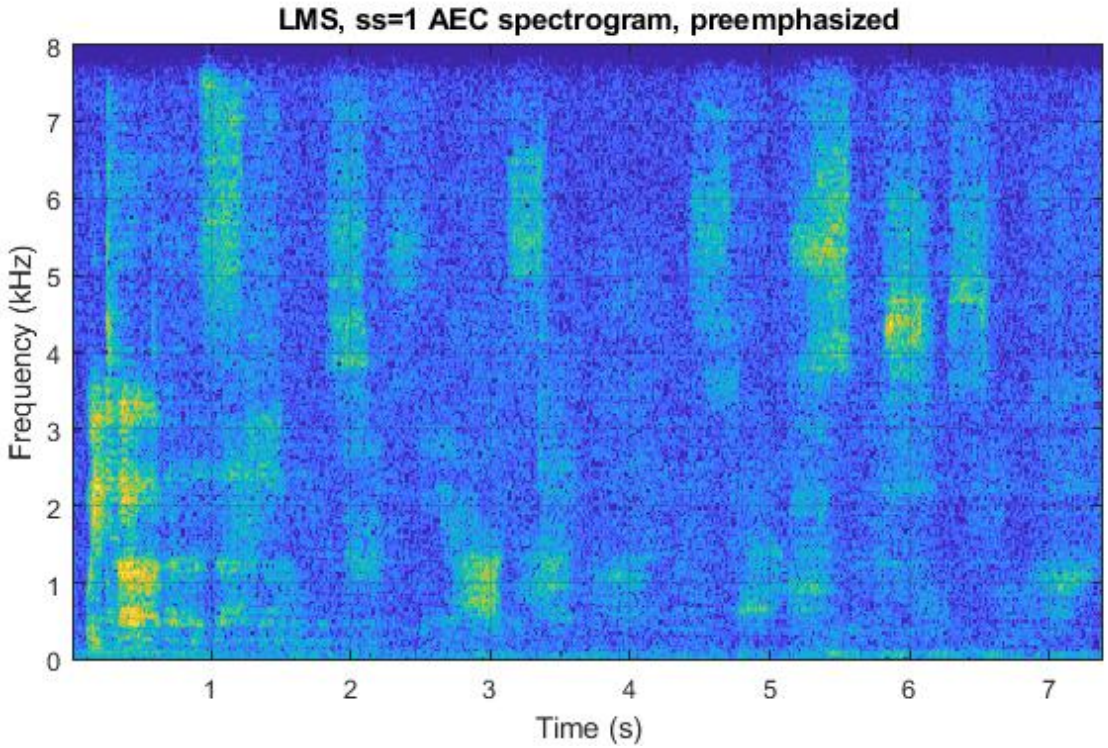
It's a different visual presentation with many signs of degradation. The messed-up adaptive tail shows that "as-is" VSS-LMS is not capable of converging above 25…30dB of ERLE (but it could be improved):



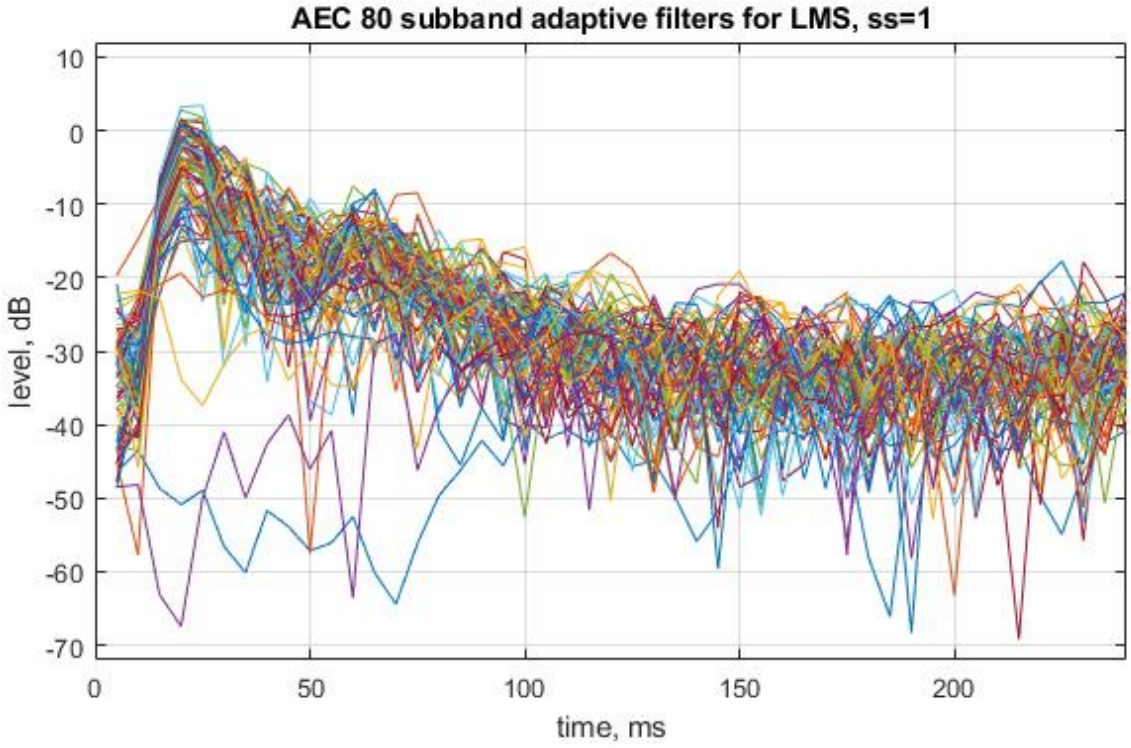
…and thus, it can be cut down to 100ms, leaving most of RIR unidentifiable and therefore uncancellable.

1.2.6 Plain LMS

Here is a plain LMS, where step size if toggled to 1 if the IN signal has enough energy to create OUT of at least 6dB above noise level:



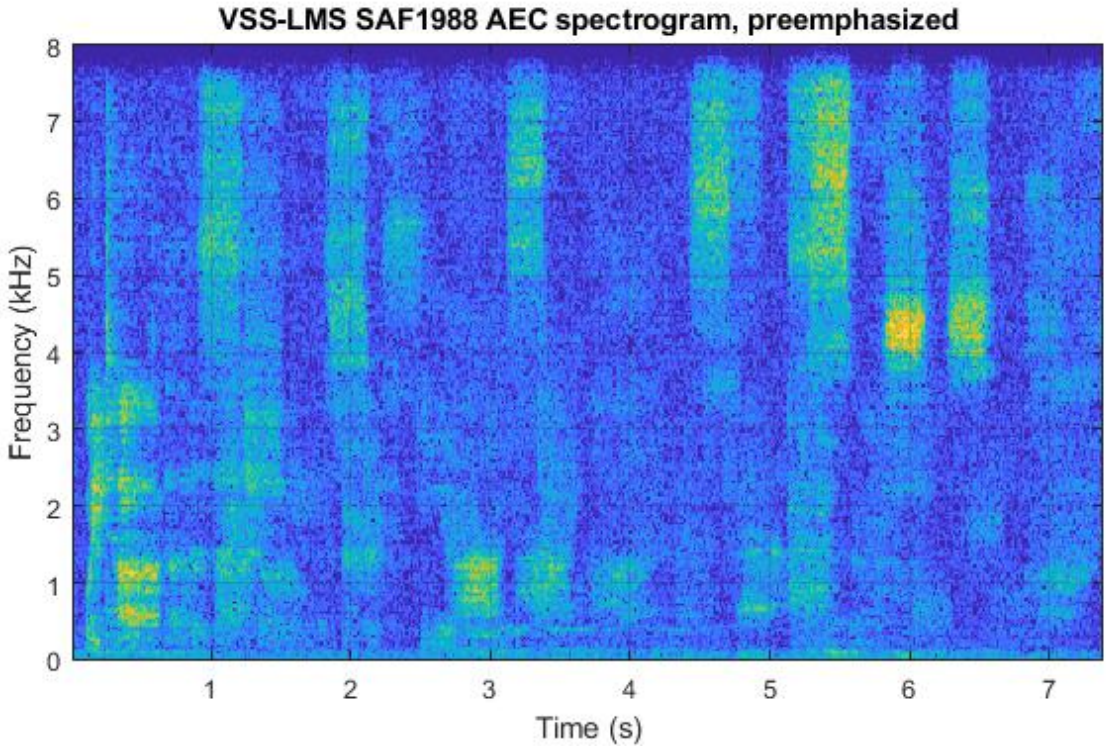
Surprisingly (or not) enough, it's cleaner than "as-is" VSS-LMS:



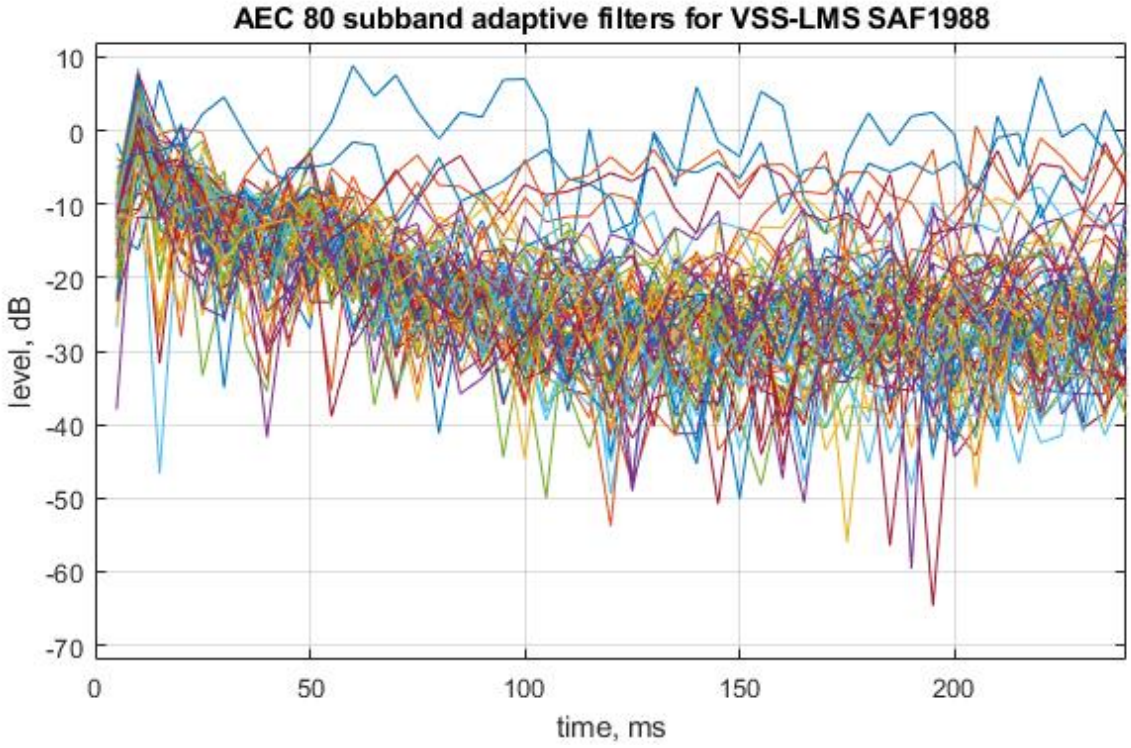
…the same true of the echo tails.

1.2.7 SAF1988 with VSS-LMS

Here we replace FSAF filterbank with old SAF-1988 filterbank, remove extra OUT delay, and use VSS-LMS by Benesty et al (2006):



…and the echo tails:

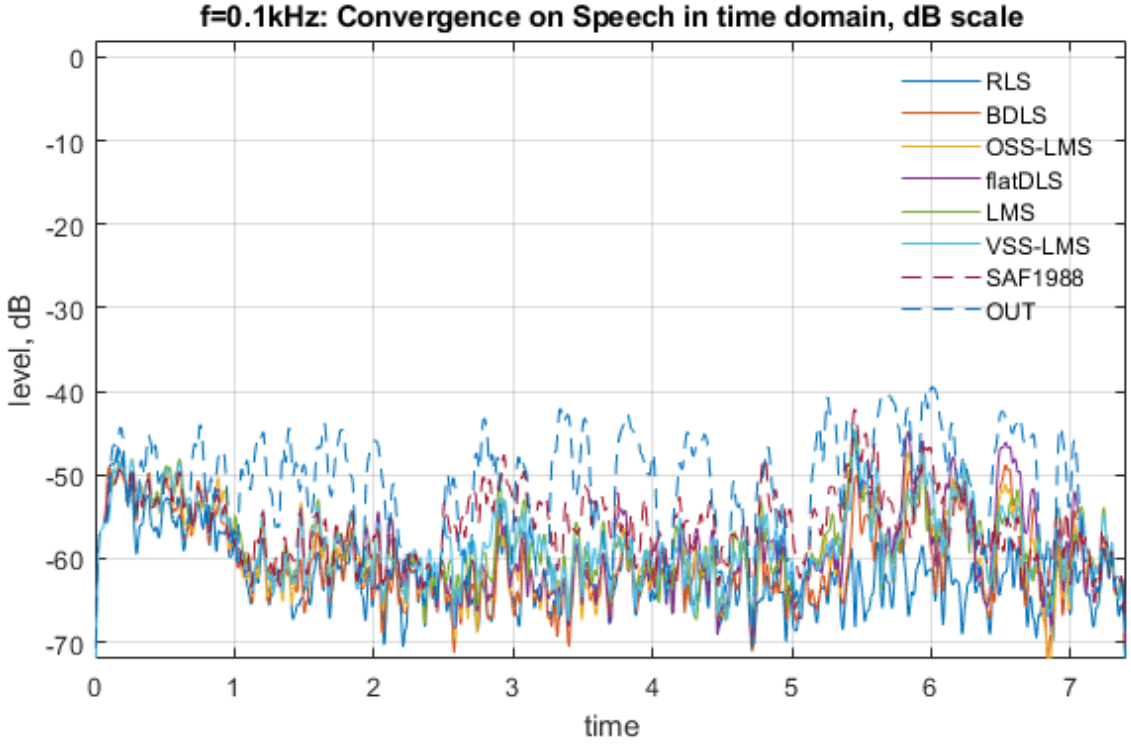


Unfortunately, this is what some companies sell as “the state of art” in modern commercial AECs.

1.3 CLOSE-UP VIEW IN SELECTED SUB-BANDS [402]

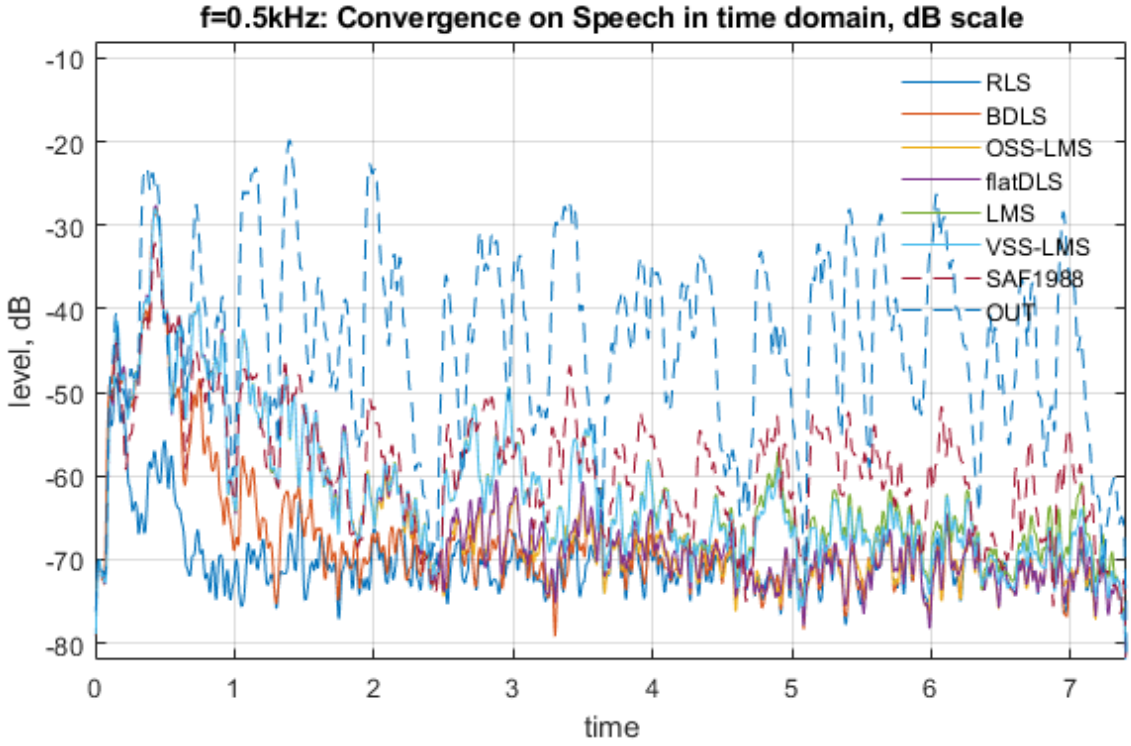
1.3.1 100Hz

On low frequencies, where pitch variations are not yet amplified by bandpass sampling, ReRLS reigns supreme:

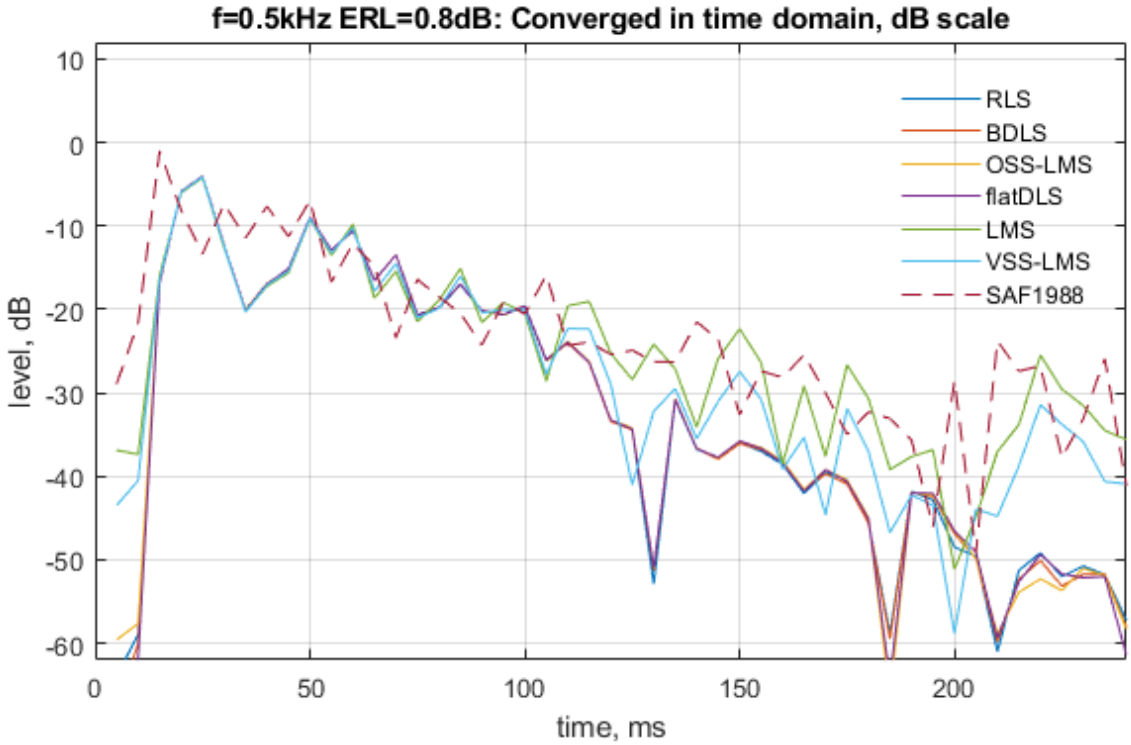


1.3.2 500Hz

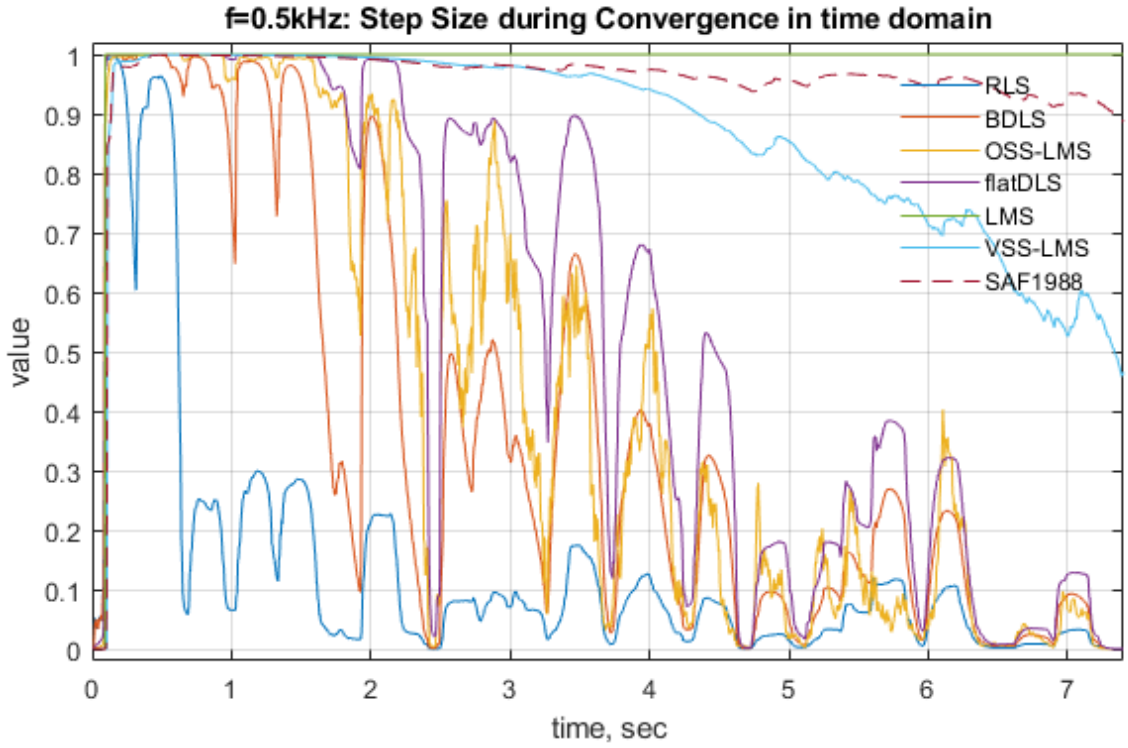
As frequency increases, BDLS starts to catch up:



We see that SAF1988 never converges well, that either VSS-LMS or LMS converge to a degree, and the first 4 algorithms converge to the same RIR (with different the speed of convergence, of course):

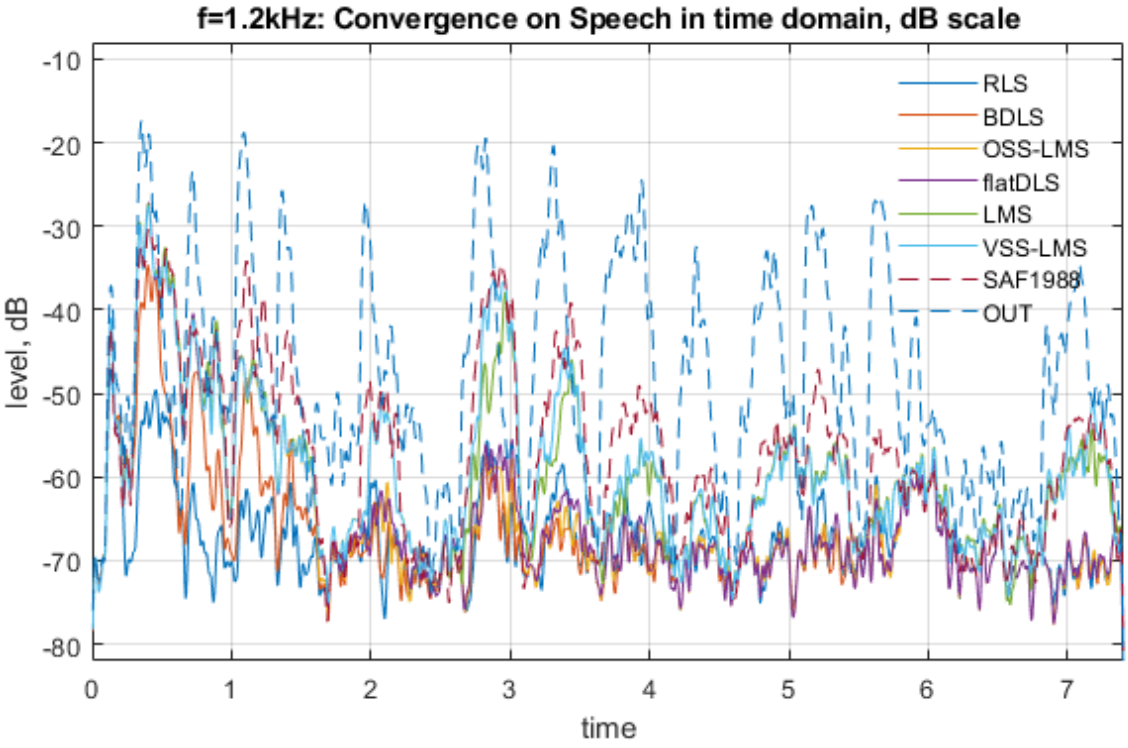


Ideally, we shall see that sub-optimal flat DLS shall produce the same step-size as the OSS-LMS. In real life, the degree of agreement varies:



1.3.3 1200Hz

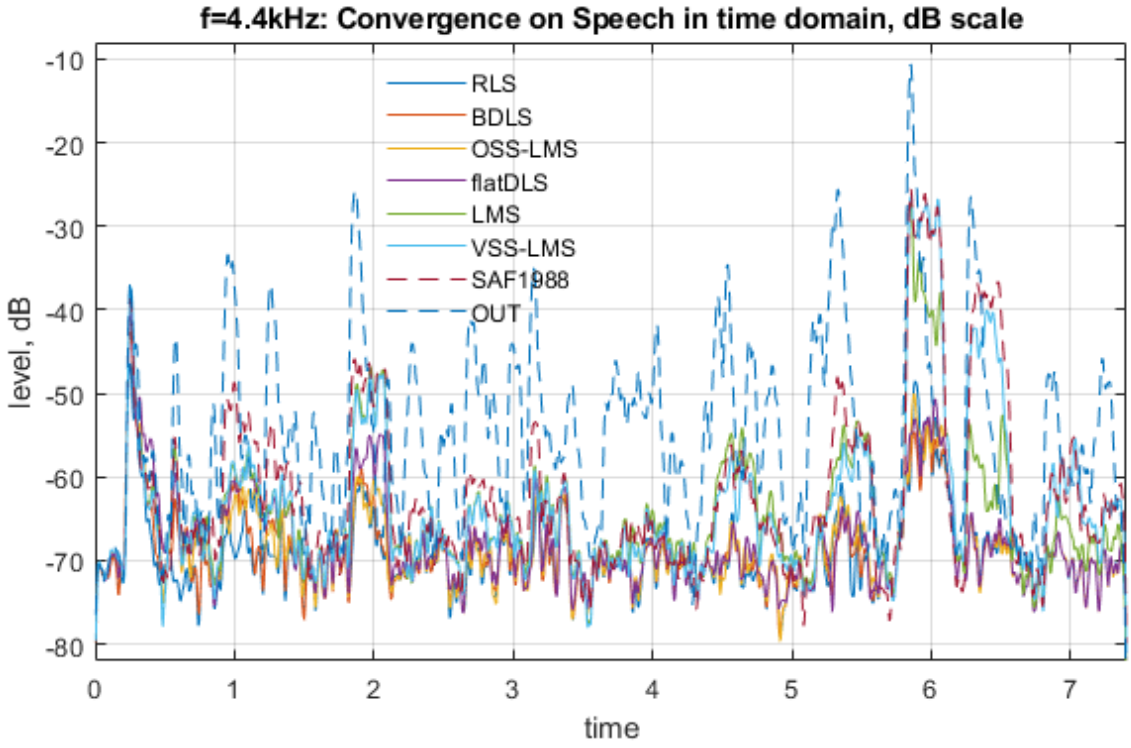
The higher frequencies demonstrate the advantages of theory-based approaches to step control quite clearly:



… and we start to see that long echo tail is detrimental to VSS-LMS and plain LMS: the residual error signal exceeds the OUT signal due to misbehaving echo tail (around 3 sec).

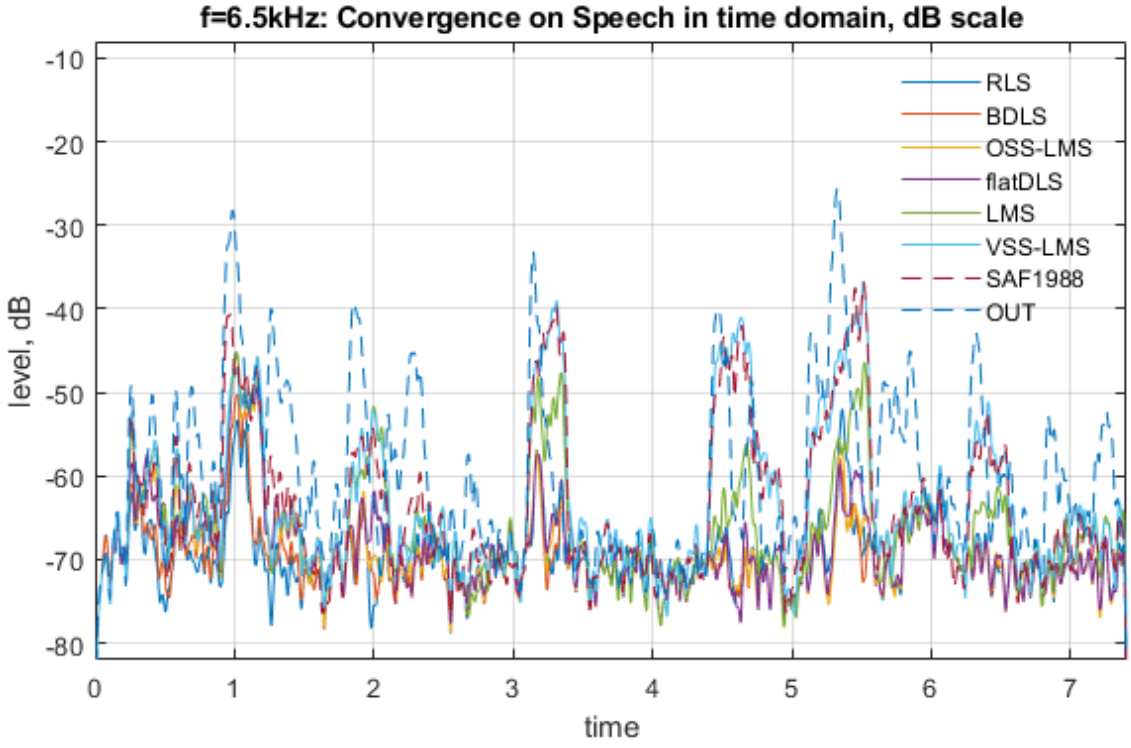
1.3.4 4400Hz

This effect grows with frequency because the fronts of the signal in higher subbands become sharper:



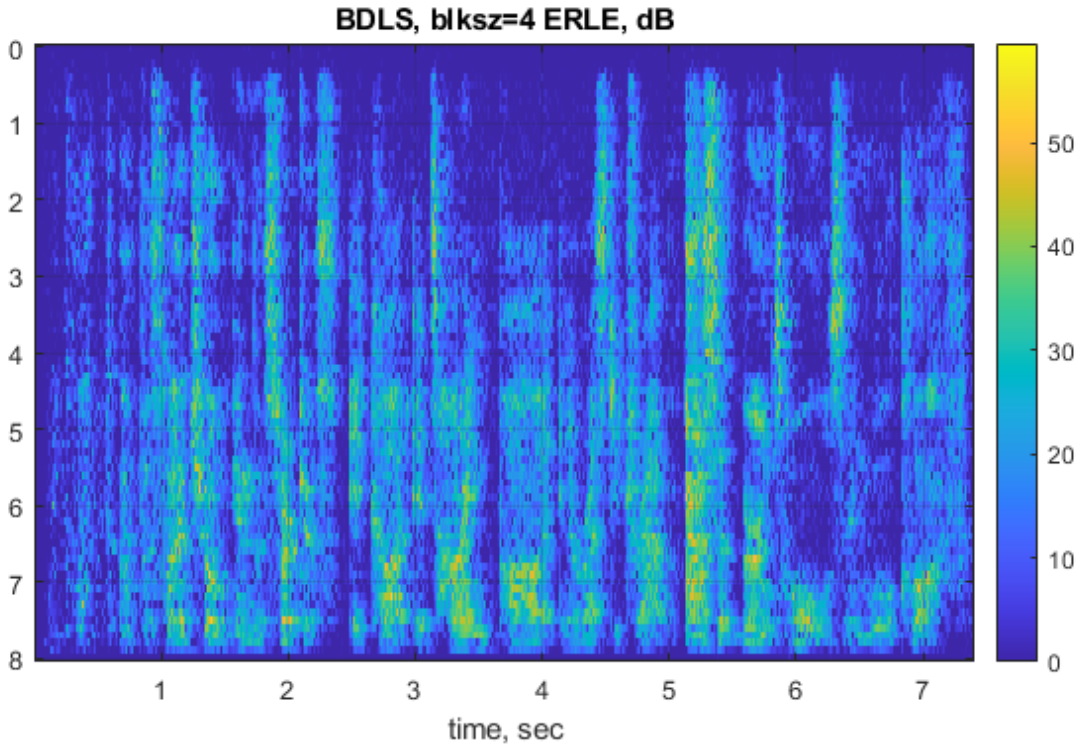
1.3.5 6500Hz

The higher frequency is, the more ridiculous it looks:



1.4 SUMMARY [401]

We conclude that neither advanced adaptive algorithms nor better subband decomposition work on their own. They shall be combined together. The echo cancellation of 45…55dB (in real life, on real voice) is readily achievable using either JTF-RLS or BDLS, or a combination of them: JTF-RLS < 2kHz, and BDLS otherwise. You don't need good double-talk performance above 4…5 kHz thus you can shorten echo tail for higher frequencies.



We did not touch on non-FSAF-specific topics as

- multiple models,
- DT and RIR change detection and analysis,
- speaker non-linearity,
  - how it affects the achievable echo cancellation
  - how it affects step size tuning,
  - how IMD envelope can be "predicted" and masked (see SpkId chapter),
  - of course, if you can predict the non-linear distortions, you'd better linearize the loudspeaker (which is no so easy to do)
- same with aliasing,
- how the expected echo estimation shall be used for suppressing the remnants of echo (NLP),
- how ARC – dereverberation and noise reduction shall be integrated and interacted with,
- Stereo AEC. Please…no one needs it.

We did not touch on AEC benefiting another side(s) in conversation/conference, nor on the vendors' need to cooperate in research and applications of AEC/ARC. Although I am convinced that there will be no end to the number of violators believing themselves entitled to everything, I, personally, am curious to learn if there also would be a vendor or two (with adequate legal department) out there, willing to support the freeware open-source FSAF AEC initiative to benefit the commonwealth.

We did not touch (yet) on implementation issues although the best way to understand an algorithm is to implement it on a fixed-point DSP in assembly:

```
; C54xx ---------------------------------------------------------
        mvmm ar1, ar2
        mar     *+ar1(#2)
B20S
                stm     #AEC_SECTION_SZ-1, brc
                rptb    B20E-1
                        ldr     *ar3+, a                ;[4]
                        ldr     *ar3-, b
                        ld      a, -6, a                ;[6]
                        ld      b, -6, b
                        mac     *ar4, *ar2, a
                        mac     *ar5, *ar2+, b
                        mac     *ar5-, *ar2, a
                        mas     *ar4-, *ar2-, b
                        sth     a, 6, *ar3+             ; [10]
                        sth     b, 6, *ar3+
B20E
                ld      *ar1, t
                mpy     *ar2, a
                add     #1, 14, a
                sth     a, 1, *ar2+
                mpy     *ar2, a
                add     #1, 14, a
                sth     a, 1, *ar2-
        banz    B20S,   *ar6-
;
; C55xx ---------------------------------------------------------
        localrepeat {
                localrepeat {
                        ac0 = *ar3+ << t2;
                        ac1 = *ar3- << t2;
                        ac0 = ac0 + (*ar4  * coef(*cdp+)),
                        ac1 = ac1 + (*ar5  * coef(*cdp+));
                        ac1 = ac1 - (*ar4- * coef(*cdp-)),
                        ac0 = ac0 + (*ar5- * coef(*cdp-));
                        *ar3+ = HI(saturate(rnd(ac0 << t3)));
                        *ar3+ = HI(saturate(rnd(ac1 << t3)));
                }
                ac0 = *ar6(#(AEC_tSc.sDexp)) * coef(*cdp);
                *cdp+ = HI(rnd(ac0 << #1));
                ac0 = *ar6(#(AEC_tSc.sDexp)) * coef(*cdp);
                *cdp- = HI(rnd(ac0 << #1));
        }
;
```

Unfortunately, ensuring that an AEC runs under an OS like Windows properly is much easier said than done:

- Equipment is not under your control. We are miles away from “plug-n-play” audio, when a “central point” knows what has been connected and is capable of control and troubleshooting.
- The gain plan is out of your control.
- Users don’t understand what they should and should not do, and it’s not easy to direct them towards a solution. Good luck explaining normal people that they can not use a separate USB microphone or compressing speakers.

- OS and driver software developers are a solid brick wall. You can design and implement a fast pre-emptive low-latency OS with proper double buffering, but people who are capable of writing such OS would never agree to spend the rest of their lives merely supporting it, and the results will be predictably disastrous.[1]
- While an AEC implementation is easy, under 150MHz on modern cores with AVX2/512, the Windows and OSs alike have not been designed for real-time media handling. Ensuring that the audio thread wakes up and run on a 10ms schedule used to be non-trivial even in the normal conditions, but there always could be a weird device driver or a BIOS version which ruins AEC's timing. Generally, an AEC fixing a failed OS is a weird idea[2].
- Intel provides an audio DSP in their recent designs – may be, that's the way forward?

I have more hopes about omnipresent smart TVs:

- TVs are routinely set at the proper teleconferencing distance, ~3m, so the camera won't show disfigured faces.
- TV vendors can put multiple mics at the top shelf of TV, use fixed beamforming and ARC.
- TV vendors control the gain plan fully.
- TV vendors control which speakers are used for videoconferencing; if the internal speakers with known IMD are chosen, an AEC may be tuned to these[3]. Generally, AECs are loudspeaker-specific rather than microphone-specific.
- TV vendors can enforce properly prioritized low-latency thread scheduling.
- TVs typically have wired connection to the router.

We need to touch on AEC testing. We need to acknowledge that the verbal communications are non-literal and "processed" subconsciously. An AEC shall provide for smooth natural conversation flow. There are no objective measurements for it. There are few recommendations based on observations of AECs ranging from outright horrible to merely poor. Their validity, impact and relation to the communication integrity are far from unquestionable. Extrapolation of these superficial indirect measures beyond the area of observations is simply unscientific.

It was noted that the better an AEC is, the more people stress it during testing. If AEC is poor, they perform the tests and hang up. If AEC is good, they move to talking about weekend plans after (or instead) of testing, they start rocking in their chairs, smile and laugh, interrupting each other all the time, continuing each other sentences, etc and it may take some effort to end the test. Everything happens subconsciously. The tester's conscious ratings may be worse for a good AEC because testers stress it more.

The best testers, by far, are teenagers. Even better tests could be performed in gaming scenarios, when roughly equal-strength teams of players, communicating 'hands-free', each team using the same AEC but different from other teams. There should be games where team cohesion is more critical for winning than in others.

---

[1] Once upon a time, I wrote a real-time multithreading preemptive OS and a network OS for a DSP network of 512 chips on 32 boards. The non-so-trivial context switch code for the thread scheduler was taken from the DSP vendor's manual. A couple of years after I left that company, some DSPs in the network started to fail tests. The timing and location of such DSPs was unpredictable. Of course, you could not attach 512 JTAGs. It took quite a few months of a large group of very clever people to find out that somebody "optimised" the original vendor-supplied code: changed pairs of stl a,*-; sth a,*- into st32 a,*- which took 1 cycle instead of 2. However, execution of st32 affect the saturation of the guard bits depending on a bit in a mode register - while execution of stl…;sth… doesn't. People commonly think that their predecessors were sillier than them, and there is nothing you can do about it.

[2] Shouldn't AEC also fix "global warming"?

[3] As far as TV vendors allow users to install an AEC of users' choice, they can use a free FSAF-based AEC.

After you've made meaningful tests and got the recordings, you can analyse and evolve AEC. The main problem is that a real-world AEC is rather complicated, with multiple nested feedback loops inside and outside subbands, and you need well-structured feedback on everything what's going on inside AEC; so, you end up with multiple screens, covered with dense graphics, changing in a slowed-down real-time.

AEC performance is limited by loudspeakers' non-linearity and non-stationarity (non-LTI-ness). This topic is discussed in a separate submission "Loudspeakers for AEC: Measurements and Linearization" due to its sheer volume.

In a following revision, when I succeed to comment the code out of being completely cryptic, I'll add a reference C code with testing frameworks.

2 ADAPTIVE REVERBERATION CANCELLER (ARC)

2.1 BASICS

2.1.1 Outline

The idea of de-reverberation has been borrowed from M. Delcroix et al (2014) presented at DEREVERB Challenge.

- The speech can be de-reverberated thanks to the timing gap between quick transient processes in the vocal tract and the much-delayed room reflections.
- Convergence shall start immediately after the falling edge of syllables (the vocal tract itself calms quickly) while the room settles down to background noise level, as per RT_{60} and direct-to-reverberant ratio (DRR).
    - The RT_{60} *(subband)* is the same as for adjacent AEC, and RT_{60} stays about the same between different talkers or while a person moves
    - DRR changes unpredictably.
    - Background noise level is assumed to be known.
- The convergence opportunity window is short and shall not be wasted.
- Therefore, the algorithm choice is JTF-RLS.
- We need to use short IN/OUT analysis filters and ignore 2…4 first taps of sub-band IR because they are mostly determined by those analysis filters, not by the RIR.
- Proper step size control is of utmost importance. In AEC, the noise is a sum of background noise, under-modelling noise, and IMD. In ARC, the noise is a sum of background noise, under-modelling noise and the de-reverberated speech (decision directed approach ++). If you want to modify an AEC into ARC, that's the only major change you have to make.

2.1.2 Clean and Reverberated Speech

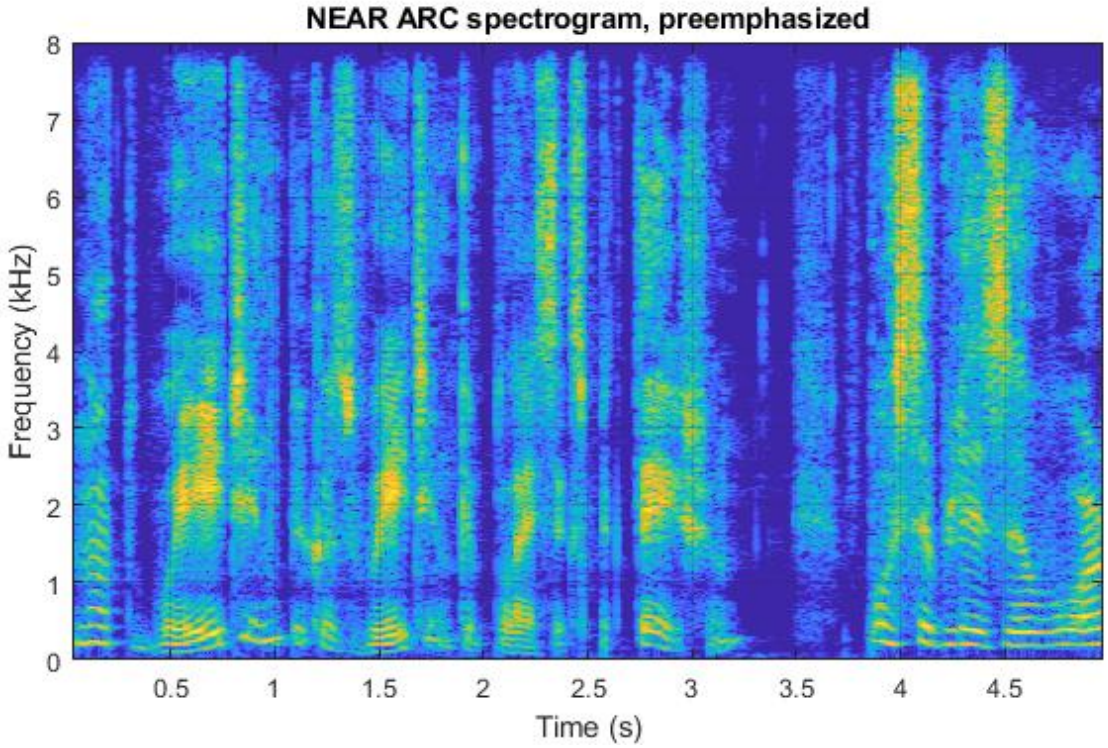
The problems with reverberated speech intelligibility are due to some fascinating facts of language(s) structure:

- The most significant "parts" are short low energy unvoiced sounds.
- The least significant "parts" are longer +15…25dB powerful voiced sounds.

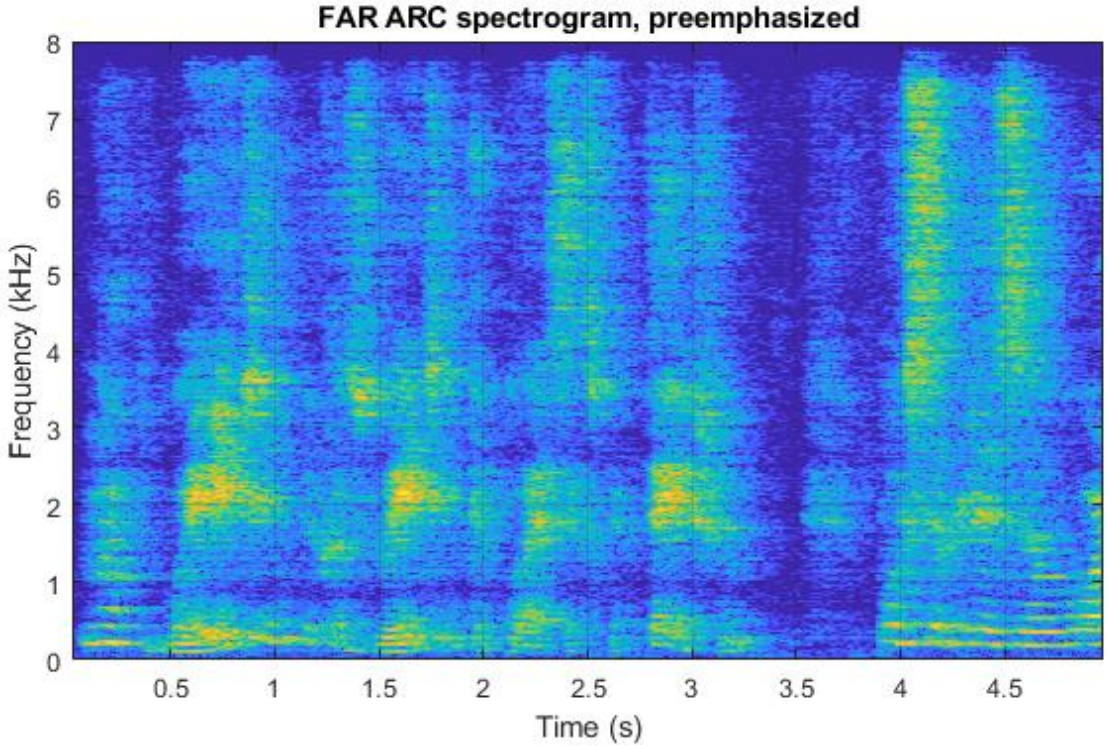
Whenever reverberation happens, the "shade" of the long and strong voiced sounds overlaps the short and weak unvoiced sounds, and affects their formant structure. The overall speech intelligibility suffers.

The transition from a voiced sound to a plosive like 'p' or 'b' is particularly tricky. We somehow rely on 20-30ms of silence preceding the plosive, which itself is very short, 2…3ms. The reverberation of a voiced sound may overlap that critical silence period completely, crushing the formant structure of both the plosive and of the front of the following sound, turning the speech practically unintelligible.

Some people are better than others in understanding reverberated speech but nobody likes it. All people report listening fatigue (of various severity) upon listening to reverberated speech. It appears that it takes quite a bit of efforts to "un-reverberate" speech in our heads. Any human person is vastly superior to all existing Automatic Speech Recognition algorithms.



A typical (far from extreme) picture of what reverberation does to speech:

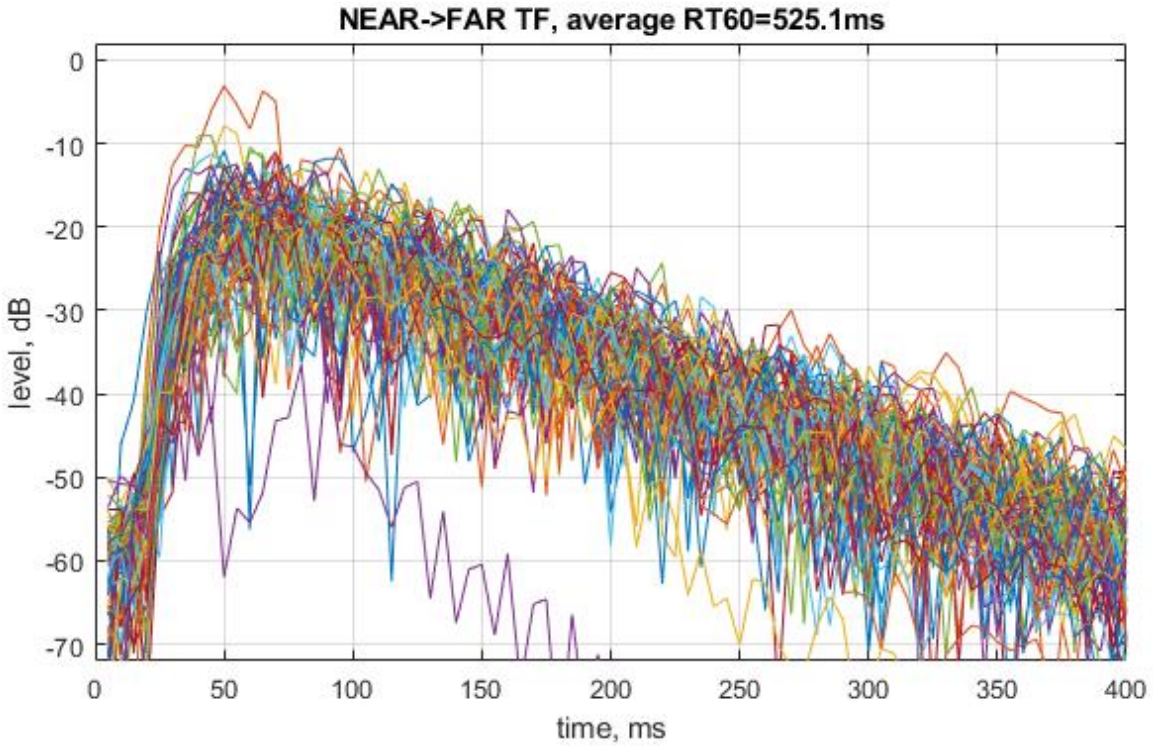


Surely enough, while the Indo-European and Afro-Asiatic languages have been developing, our ancestors[4] did not live inside caves; the popular 19th century armchair philosophers' "caveman" hypothesis contradicts to Zeitgeist's lasting imprints.

[4] at least, not all of them

2.1.3 RIR

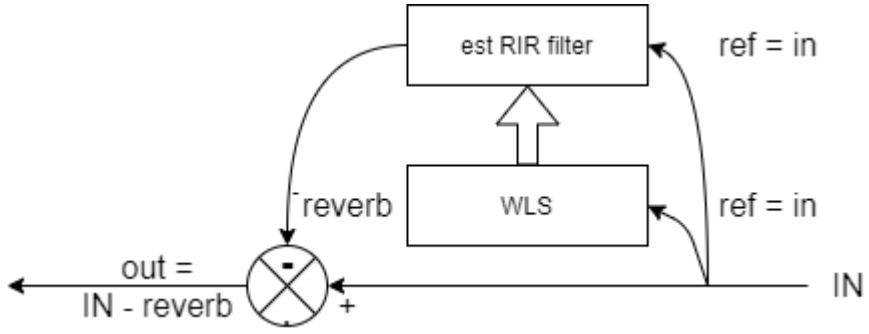
Because we have saved "near" recordings, we can easily compute the transfer function between NEAR and FAR microphones, together with RT_{60} estimation:



When we have only FAR recording, estimating TF is not so easy. In a general case, blind deconvolution is impossible.

2.2 WEIGHTED LS [404]

The usual solution is to identify the auto-regressive function by an adaptive method, focusing on late reverberation [Yoshika 2012], with time delay τ_{late}.

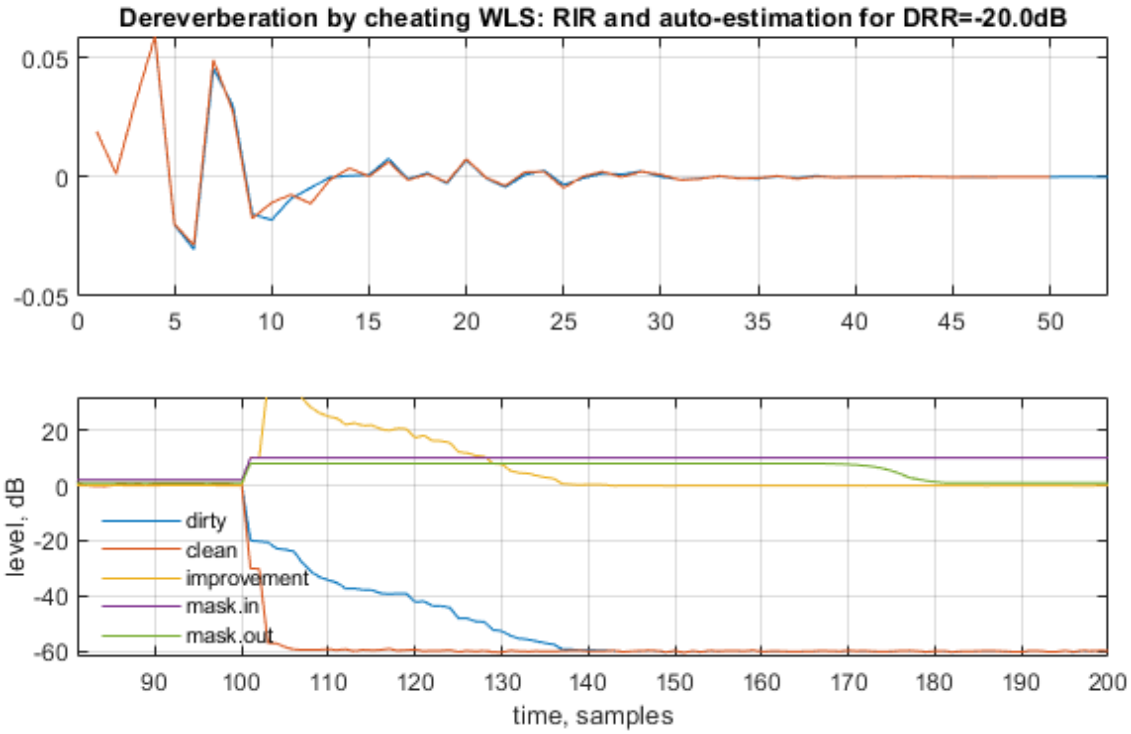


Let's investigate how it works using a simple model, without any additional complications like subband decomposition, etc.

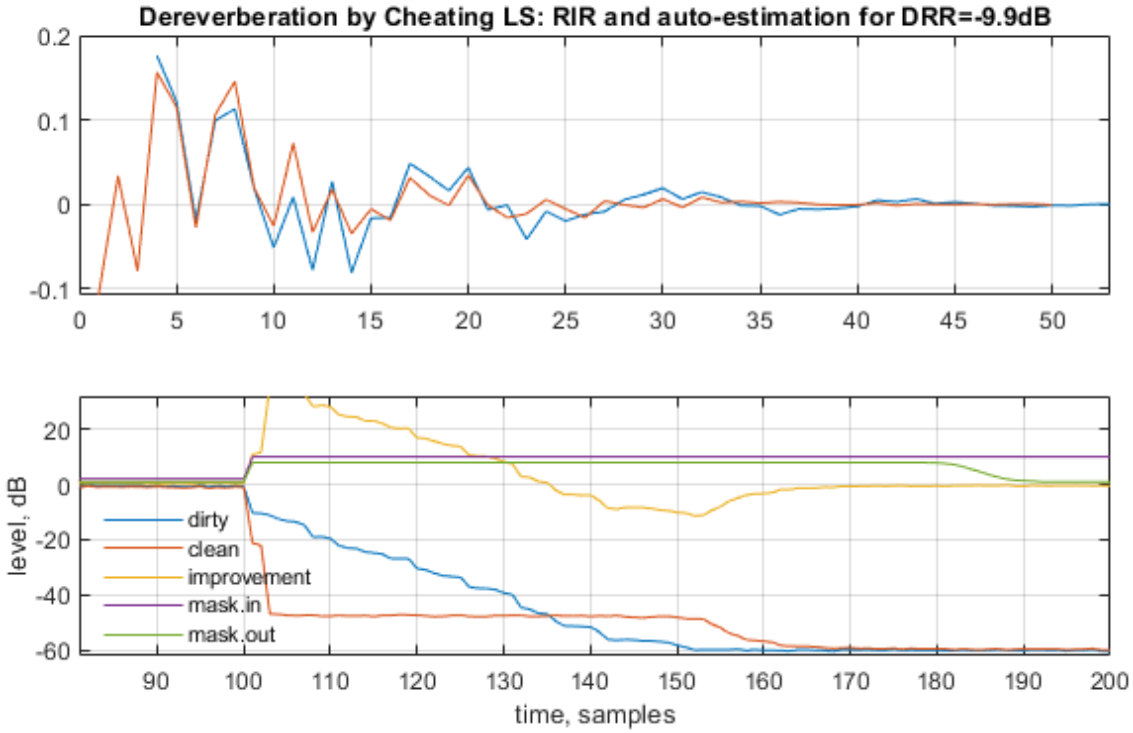
- Artificial sampling rate is very low, say, 100Hz, so that our models are of low dimension;
- Original signal is composed of bursts of white noise of, say, 1 second, separated by 1 second of silence
- The original is distorted by a RIR with a known RT_{60}
- AWGN of low level, say -60dB

- We “know” when to start adapting: immediately after the original burst ends. We do not run multiple models here (as we would do in reality) to find that moment.
- We “know” that after the burst ends, there is no signal, just reverberation and AWGN.
- We “know” AWGN level
- We cheat by using a Weighted Least Squares with binary “On” while only reverberation (above AWGN) happens and “off” otherwise. To make it not-cheating W-ReLS, we need to weight the observations with an inverse square of their respective residual error … which should have almost the same effect.
- We average the results over, say, 100 bursts to see the trends.

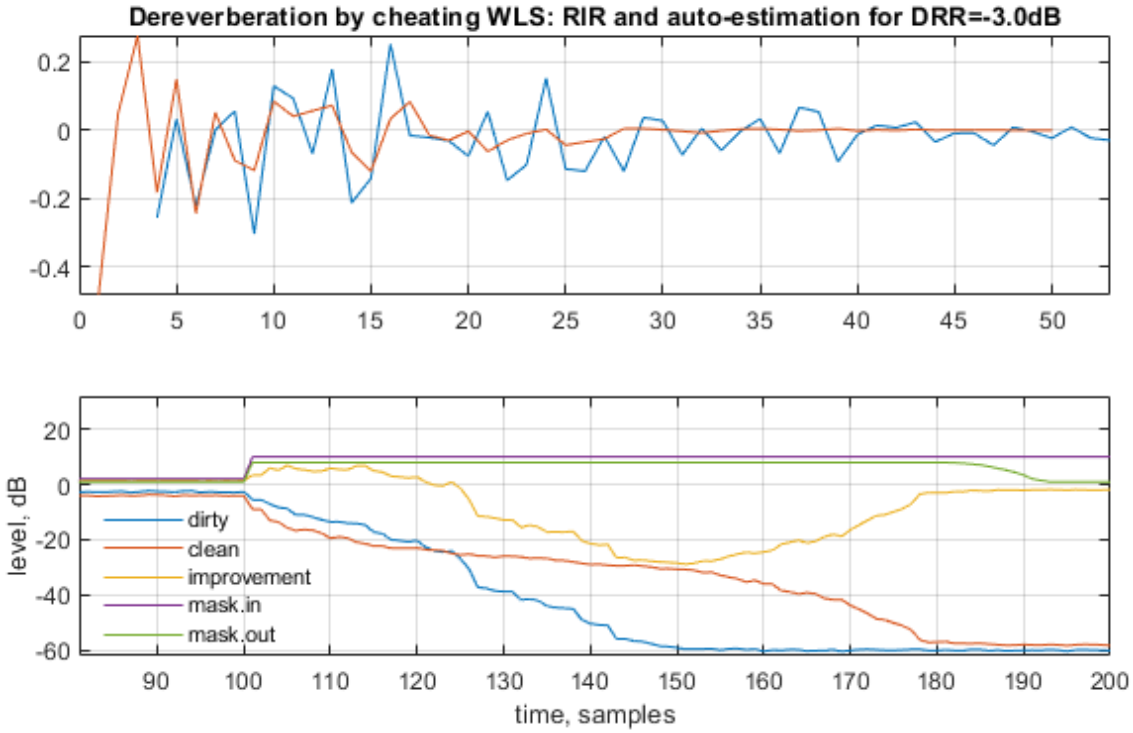
For low DRR (20dB, 0.1), the things look somewhat good. The convergence speed is kind of low, but nothing too alarming.



With lower DRR, the convergence slows down (not seen here, either trust me or check yourself) and it does not look so rosy:



We start seeing the shelf of errors produced by noise-on-input, but it's still not too bad (or not always too bad, depending on the τ_{late} under-modelling severity and other factors). With worse, but still realistic DRR, the things fall apart completely and loose any meaningful shape:

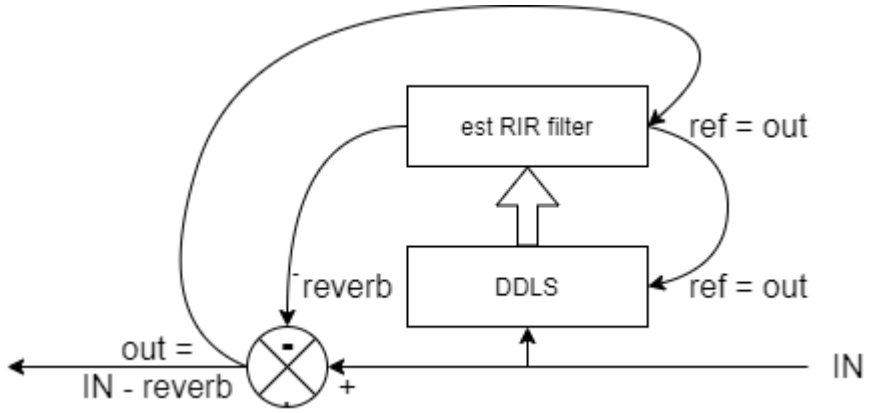


The main source of that high-DRR-related problem is the underlying hypothesis of autocorrelation:

- Inside the slope of reverberant tail, there is no autocorrelation between samples. They do not depend on the previous samples, but only on the last samples of clean original.
- The noise on input, due to the mixing of the original with reverberated, becomes obnoxious, and convergence slows to a crawl
- The time delay τ_{late} introduces an under-modelling error without a way to account for it adequately. This high magnitude non-negligeable error impacts convergence pretty seriously.
- Besides affecting convergence, let's note that the autocorrelation function exists only in statistical sense. Even if it is identified perfectly, there still will be a "shelf" of errors stretching out, after cleaning, due to the noise on input.
- The problems grow up as a square of DRR because both output and input become contaminated, and stretch out with $RT_{60}+L_{ADF}$.

2.3 Overcoming WReLS deficiencies with DD-WReLS

Let's explore using Decision-Directed approach in a loopback, i.e. let's use the output, the cleaned signal as the excitation for each next iteration.



Basically, we reduce ARC to AEC by splitting the signal into:

- IN excitation: "everything" that happened within RT_{60} (or so) before the reverberation slope started. The IN is corrupted with same reverberation, and we "cancel" it by using the same RIR we have just identified (but without adapting, SNR is too bad). After the reverberation slope starts, IN is assumed to be zero.
- OUT afterwards, till the next word starts. There is AWGN on top of it. We adapt on it as usual in AEC. Before the reverberation slope starts, OUT is assumed to be zero (or contaminated by noise of infinite variation σ^2).
- A narrow transient area between them, which we can not really classify it as either IN or OUT, and it's not exactly clear what is the best way of dealing with it.

AEC is not real-time here because deciding when to adapt (and when not to) is not trivial. It's much easier when you have the history of the signal and can observe long-term trends. Thus, we use ReLS instead of ReRLS.

There are quite a few interesting questions: Will such DD-WReLS converge? Sometimes or always? What is the distribution of errors, normal Gaussian or fat tail-ish with huge outliers? What would happen for this or that DRR? How do we test it? What can go wrong?

2.4 REVERBERATED SPEECH SEGMENTATION

Segmentation is the most difficult task. We need to find all falling slopes, suspicious to have mostly reverberation, run an iteration of ARC, DD-WReLS, see if it is a good starting point for iterative optimization:

$$W = arg\left(min_w(\Sigma\|e_{res}\|^2)\right)$$

where W is a covariation matrix (more like a diagonal) of the true original signal before reverberation for WReLS. Here we can simplify the inverting W by exploiting the huge dynamic range and therefore treating the W^{-1} estimation as a sequence of ones and zeros. As usual, we can do adaptation the best if we know the statistical properties of the system and signal, the more, the better.

Instead of iterative optimization, a more realistic Multiple Model approach can be used for finding acceptable segmentation, with optimality criterion as having very low residual error in the marked segments just after the transient area.

Segmentation of real-life speech in sub-band domain is a non-trivial algorithm, guided mostly by speech-specific analysis techniques and best suited for ASR specialists. Alas, there is next to nothing there what can be productively discussed within adaptive filtering approach except that segmentation and clean speech postprocessing could be aided by adaptive cancellation: if we succeed to cancel some reverberation at a few adjacent subbands, there may be more of it, right here, and we can dig dipper with applying attenuation.

Suppose we somehow solved the segmentation… the remaining is AEC-like and less challenging.

2.5 ADAPTIVE PROCESSING NUANCES [406]

As usual, we start with most simple models, pulsed white noise and delta-function RIR, and slowly move up to real-life signals, one step at a time.

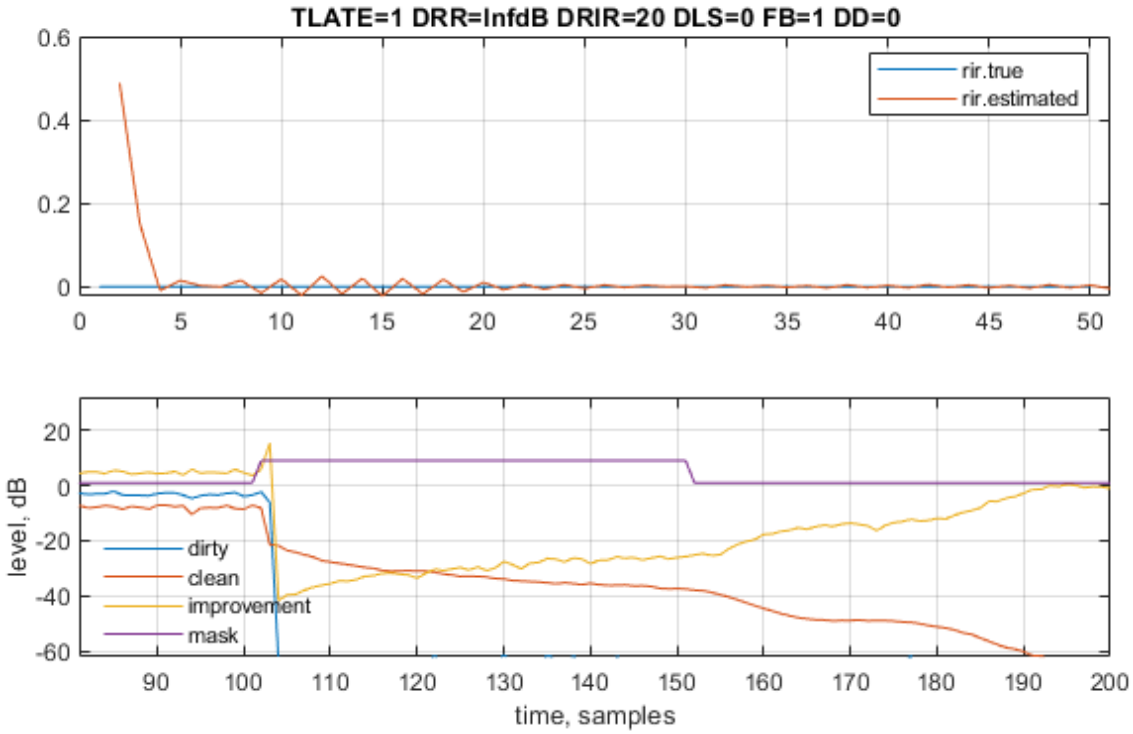
The class *fsaf_arcc.m* (ARC cheating) was designed to provide visibility into internals of ARC's WReLS. On the figures below:

- TLATE = τ_{late}
- DRR – Direct to Reverberant Ratio in dB. 10dB -> *norm(RIR)= 0.316*;
- DRIR:
  - if > 0, an offset to a single pulse
  - if == 0, exponentially (RT60) decaying RIR is generated
  - if < 0, the first (-DRIR) samples of decaying RIR are zeroed.
- DLS: which algorithm is used. 0=WReRLS, 1=NDLS
- FB: do we emulate the effects of FSAF filterbank
- DD: if Decision Directed WReLS is used

2.5.1 DSF considerations

ARC is a version of FSAF AEC, and the same delta-function spreading effect is present in sub-band dereverberation algorithms. An attempt to identify an artificial RIR of all-zeros (following 1.0) results in

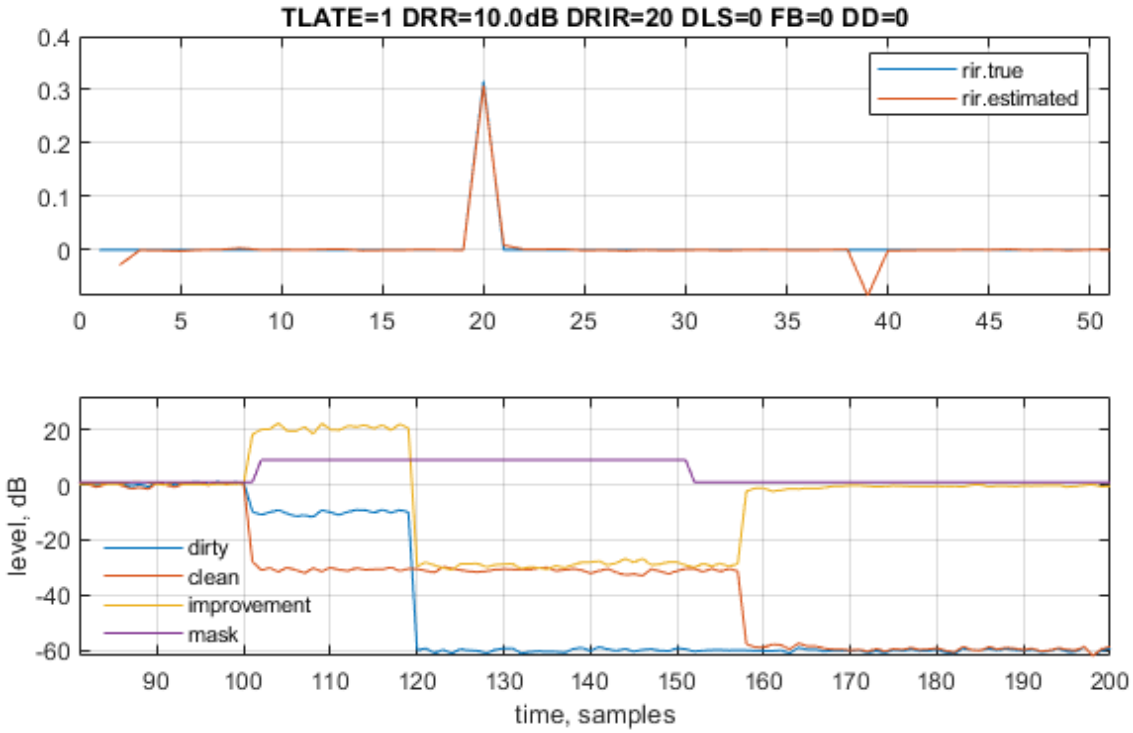
artifacts like a DSF shoulder and appearing reverberation instead of clean “dirty” signal:



Let’s turn sub-band filterbank simulation off to avoid masking of other effects.

2.5.2 WReLS estimates zero-delay inverted RIR

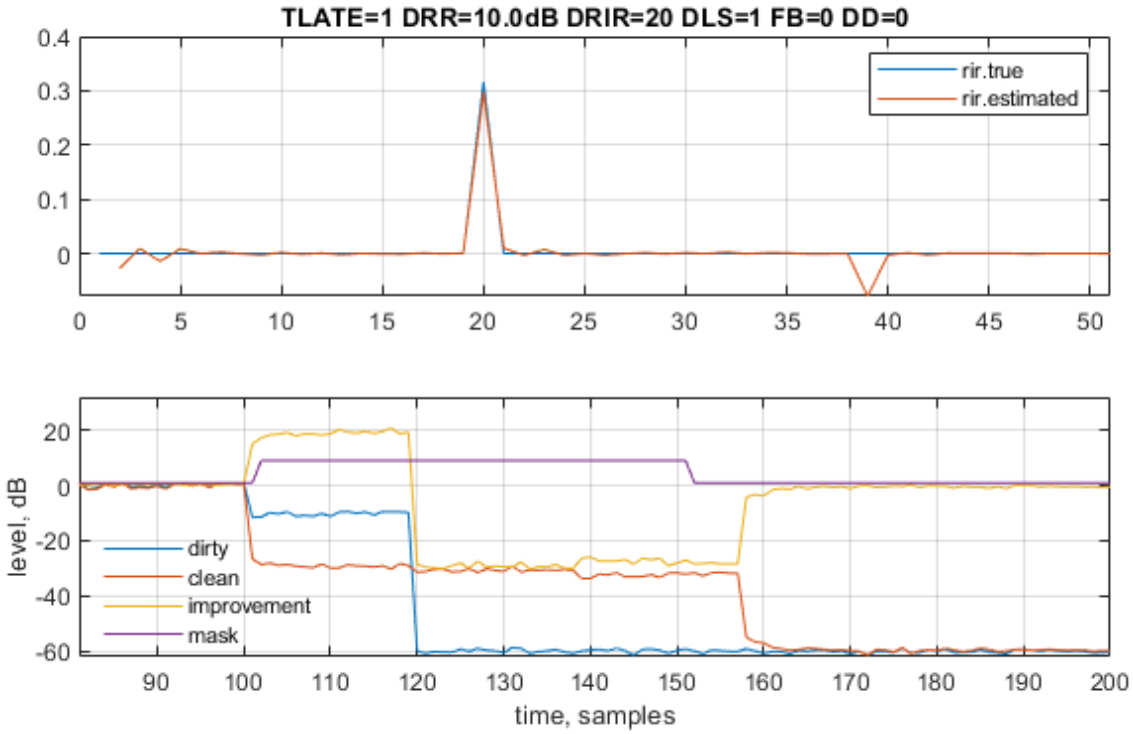
The non-DD WReLS attempts to compute a delayless de-reverberating FIR, i.e. one that *conv(RIR, FIR)=δ(0).* It would be perfect if it were always existing:



…but it isn't.

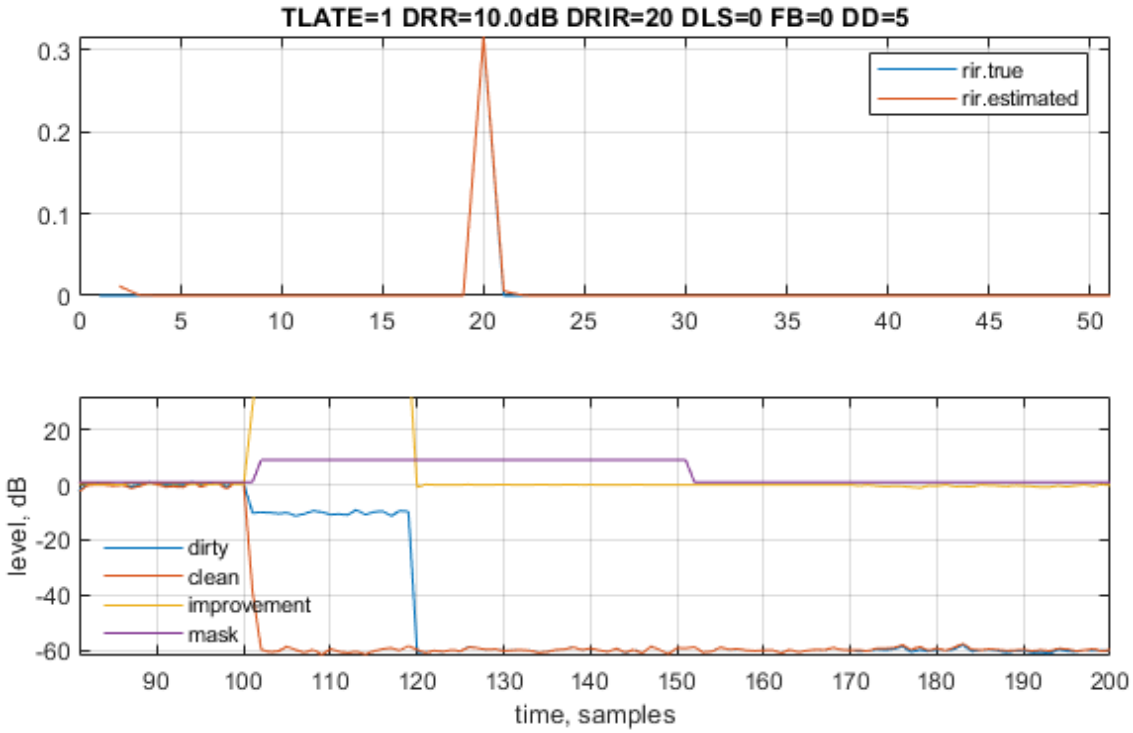
2.5.3 Using other simple adaptive algorithms

… is not a good idea because the time for RIR identification is so short. Here we attempted to use a vector-sized NDLS:



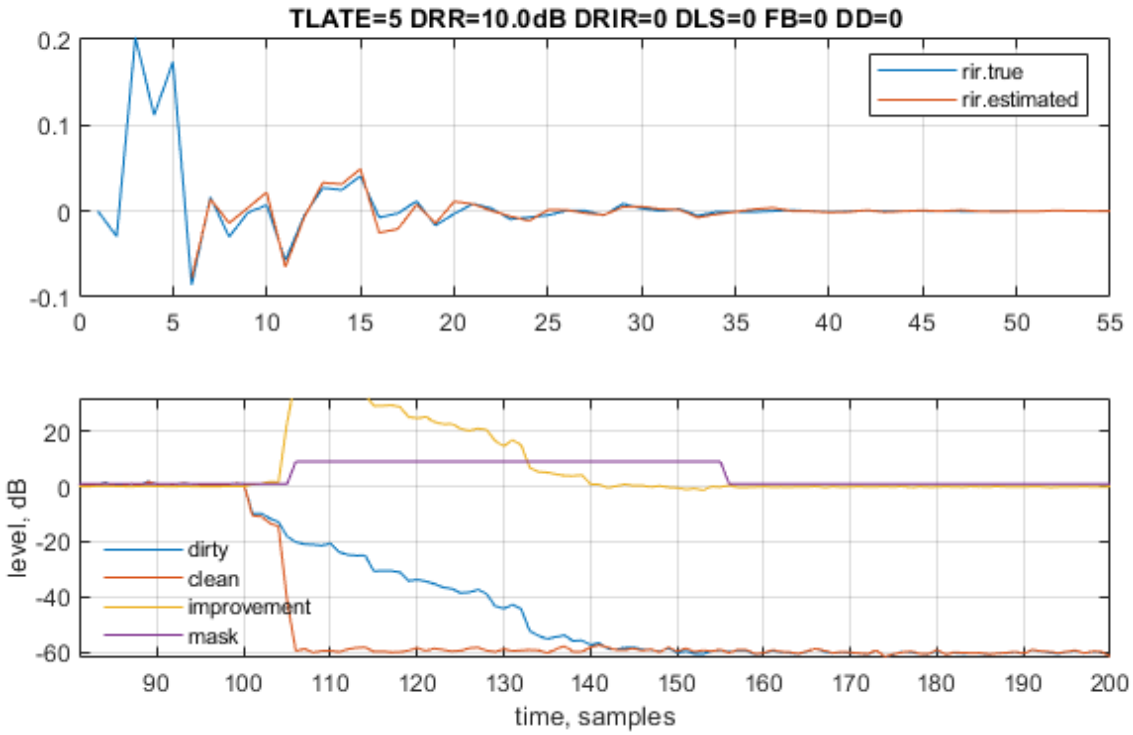
2.5.4 DD-WReLS

Using Decision Directed WReLS leads to computation of something having much closer resemblance to RIR, with fewer artifacts:



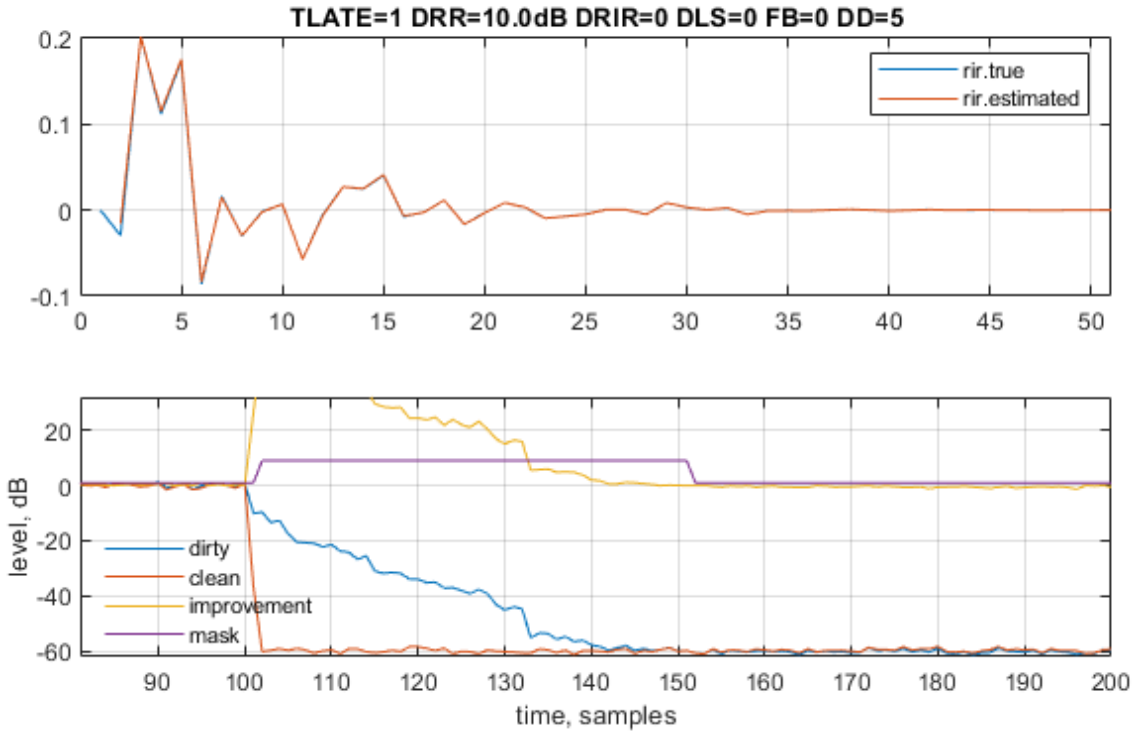
2.5.5 RT_{60} tail, Long τ_{late}

The longer is τ_{late}, the easier is the task computationally because effective DRR improves, etc. But how it sounds and affects WER is a completely different story:



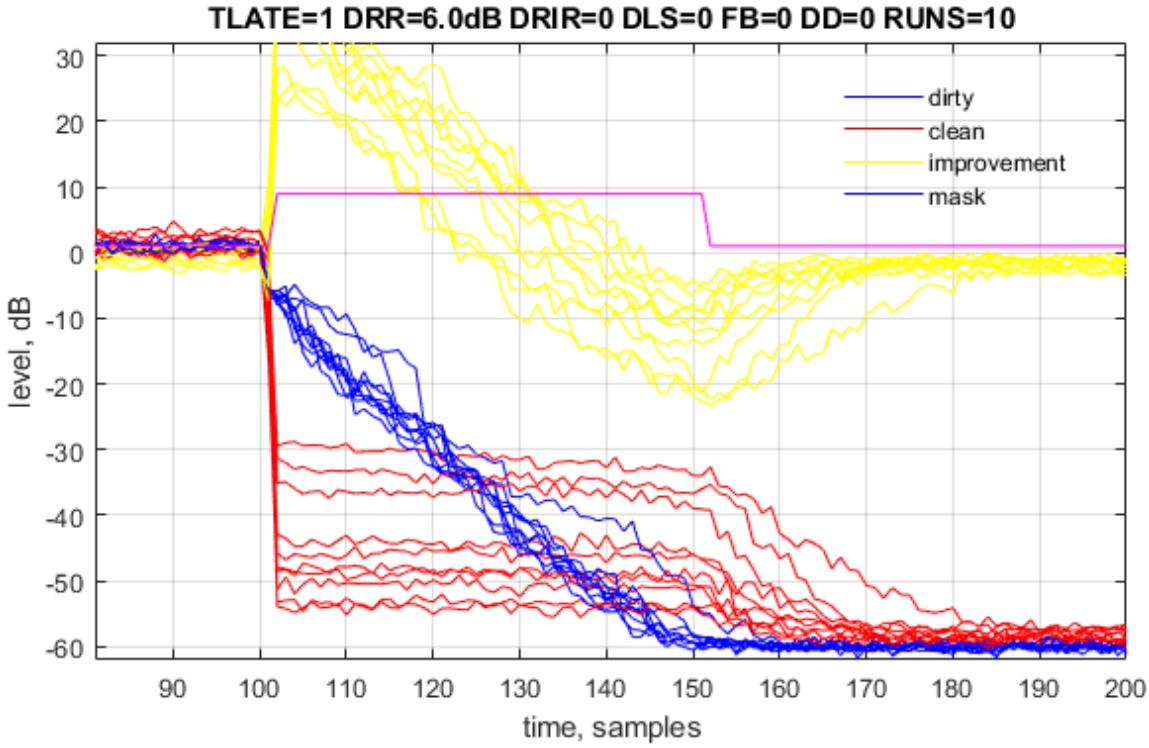
2.5.6 RT_{60} tail, Short τ_{late}, DD-WReLS

That is about working algorithm:



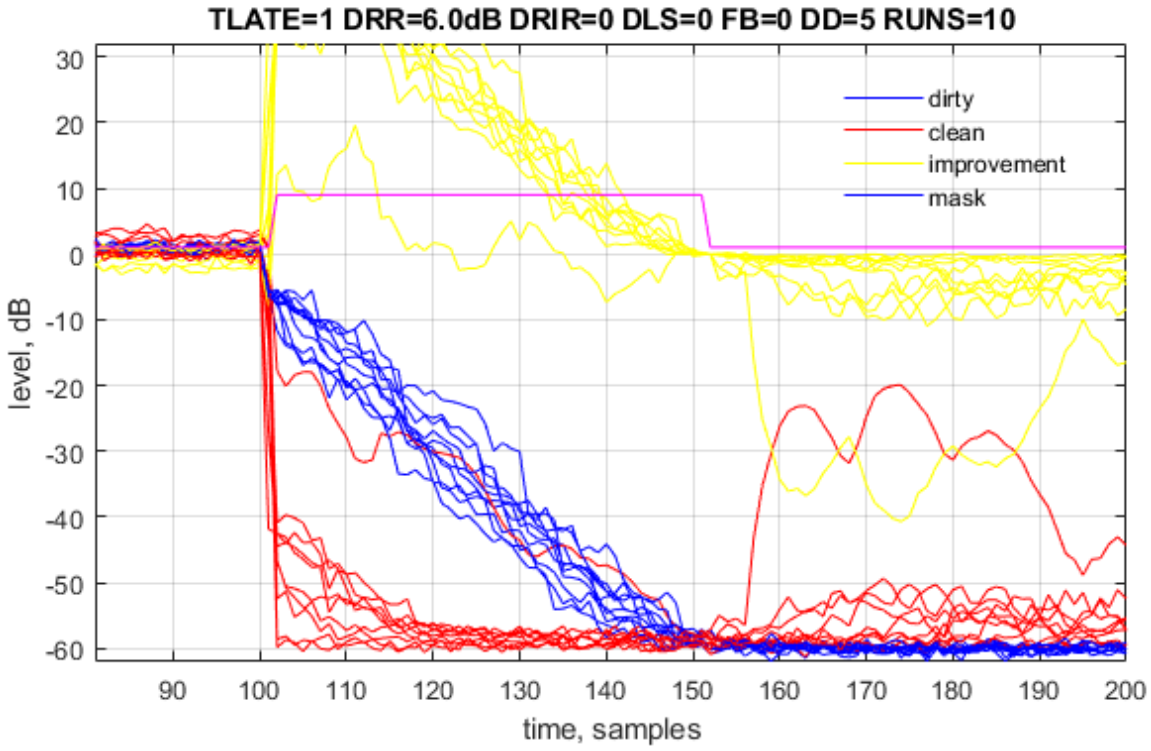
2.5.7 Repeatability of WReLS

Same DRR, but different RIR functions, 10 realisations (RUNS). I suspect that it's very hard to predict how well a RIR with a given DRR is invertible:



2.5.8 Repeatability of DD-WReLS

It's better but the fat tails are of significant concern:



2.6 SUMMARY

It may be a plausible suggestion to

1. identify the *RIR(t)* itself,
2. invert it to $RIR^{-1}(t)$ outside of adaptive filter, with a specific $\tau_{inv} = arg(min_{\tau}(||conv(RIR, RIR^{-1})-\delta(\tau_{inv})||))$ which is not necessary 0, and
3. use $RIR^{-1}(t)$ to de-reverberate speech.

It could be a bit challenging to find τ_{inv} which is optimal for all rooms and positions between a person and a microphone.

Once upon a time, I ported AEC into ARC in about 3 months and was very proud that WER dropped twice. I did not realise at the time that the effect was non-scalable, and that the problem was much harder than it looked. I apologise for not going into the finer details of ARC which are outside of my research interests.

3 LOUDSPEAKER / ROOM MEASUREMENTS [408,409]

FSAF can be put into a good use by DIY enthusiasts for loudspeaker and room measurements because FSAF is much more adequate and precise method than the alternatives.

The traditional methods, MLS and chirp, were developed for radars / sonars, to improve their resolution with limited TX peak power, in mid 1960s, and studied to death.

Convolution of long MLS or chirp with itself produces a nearly perfect delta function; thus, they have nearly perfect identity Fisher matrix; thus, you don't have to invert it; thus, you can simply perform a convolution to achieve reasonable results:

$$h = (X^H X + \gamma D^{-1})^{-1} X^H y;$$

$$\text{if } X^H X = \alpha I \text{ and } \gamma = 0 \text{ then}$$

$$h = \alpha^{-1} X^H y$$

The same methods also been "invented" by "audio-science" circa 1980.

Unfortunately, convolution-based methods excel in producing abundant artifacts; they are excessively sensitive to non-stationary noise, nonlinearities, and other deviations from ideal conditions due to the implicit matrix inversion.

The MLS / chirp applications to audio have been heavily criticized by "subjective" testing proponents who pointed out numerous times that the MLS / chirp measurements results contradict to human perception but these objections were disregarded by "objective" testing zealots.

3.1 BASICS

The end-customer sound reproduction is usually simplified as an LTI (Linear Time Invariant) process to be fully characterized by its infinite RIR (Room Impulse Response), due to our much lower degree of understanding of non-linear time-variant systems rather than to LTI relevance.

Let's start with characterization of a budget ADAM F5[5] studio monitor. Equipment used: monitor with settings: eq = 0, gain = max, flat, 2 channels of Focusrite Scarlett 18i20, no-brand balanced cables, AKG P170, Apex 220[6]. Monitor and microphone(s) are set 0.5m apart in the middle of a living room 11' x 24' x 9', where ~30% of walls are covered by 1" and 3" acoustic foam.

We use MLS ($L=2^{14}-1=341.3$ms), exponential Sine Sweep (20Hz…23kHz), and FSAF (340ms), all on the same acoustic level (80dBC@1m), with the same spectrum, and duration (15 seconds). The excitations are pre-shaped to the same -6dB/oct before DAC and inversely filtered after ADC, to prevent tweeter burnout and maintain ~frequency-invariant SNR re room noise.

To examine 'output' errors, we subtract LTI modelling from the observed data, and plot the spectrogram of this residual.

To examine the 'modelling' errors, we delay the microphone signal by ~80ms relative to loudspeaker, so that these "non-casual" 80ms before the RIR's main spike contain only model's noise.

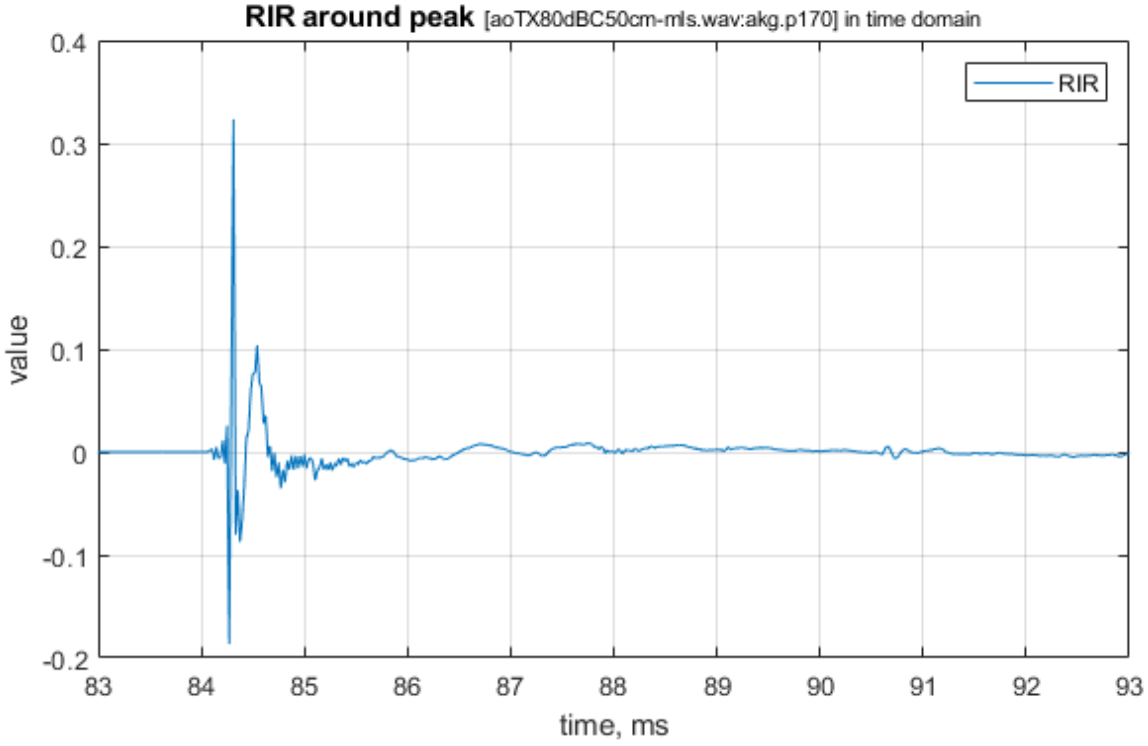
[5] which collected many favorable reviews over the years

[6] You can get away with a 2i2 or equivalent, on-battery laptop, any cheap measurement microphone, and a reliable 'true condenser' ½" pencil cardio mic with < 20 dBA noise and >130dBA distortion limit.

3.2 METHODS COMPARED

3.2.1 MLS observations

MLS calculations were performed using MATLAB audio toolbox functions, and together with the computation of residual, took about 0.8…1sec[7], which is ~7% real-time load.

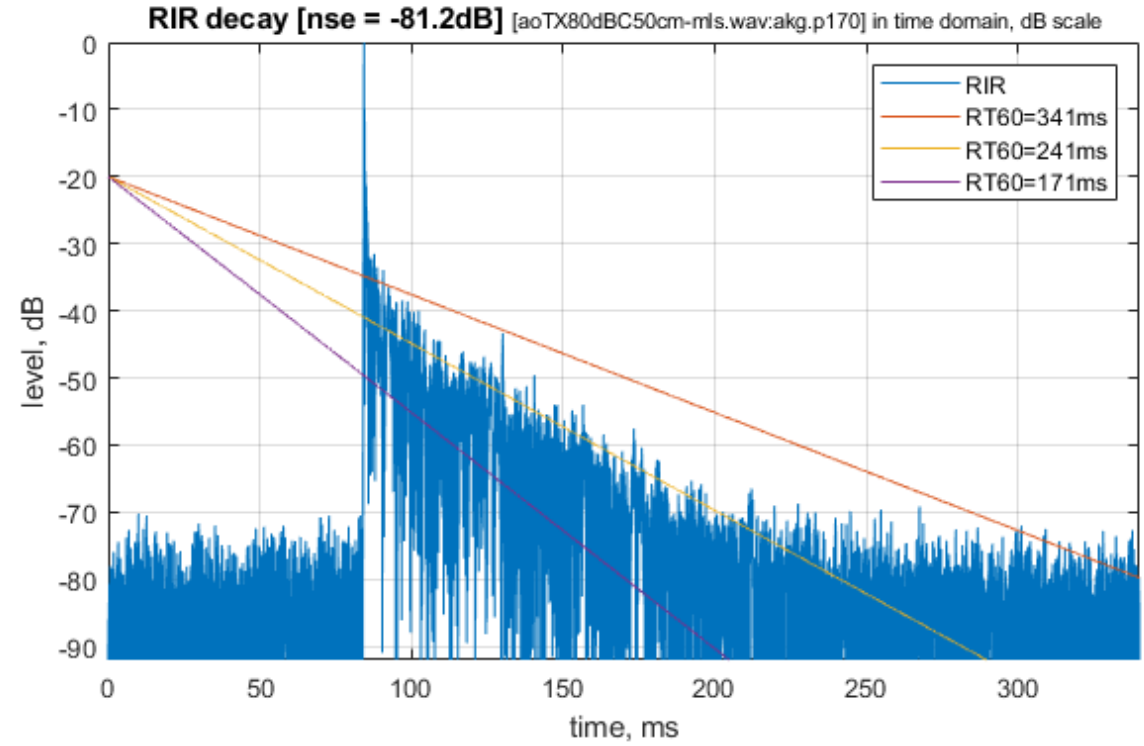


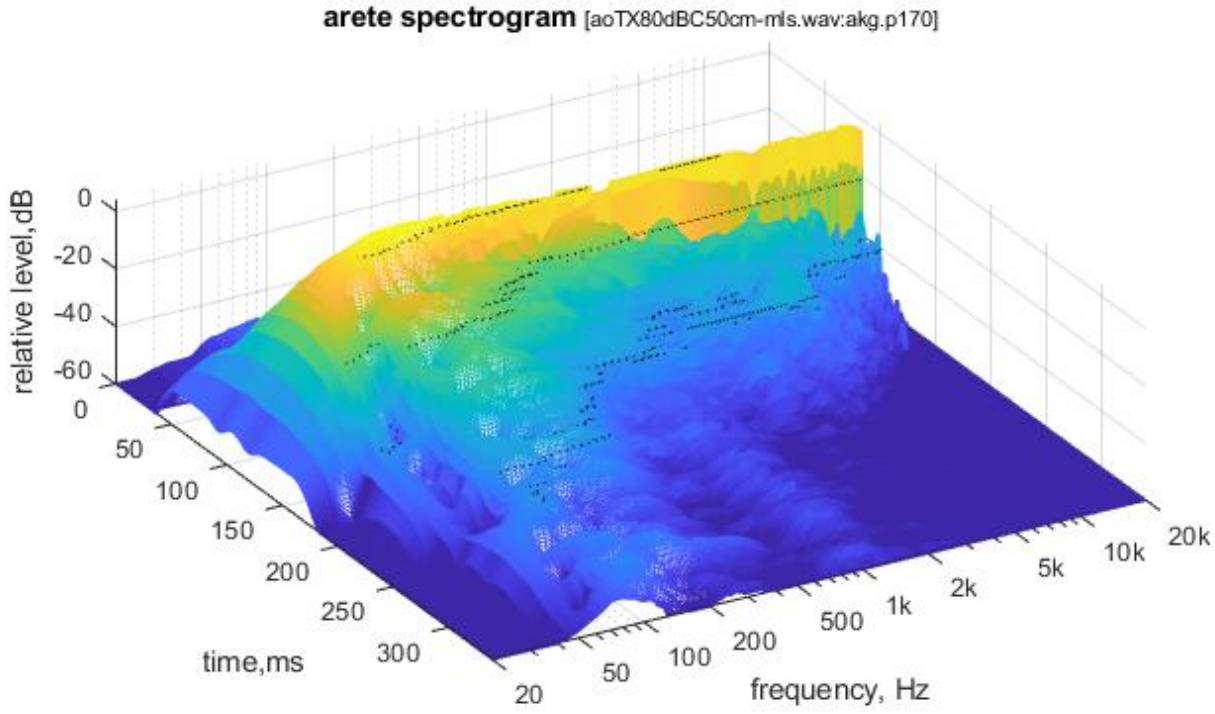
The linear time-domain representation of RIR is not overwhelmingly visually informative.

The log scaled (dB) time domain figure below shows that the modelling errors are uniformly distributed along time axis, in a so rare nowadays full agreement with the theory.

The anticipated modelling errors are 10…20 dB higher than the actually measured residual error. Let's recall that MLS excitation is an exact repetition of the same nearly orthogonal L test vectors which are all reside on a circle in L-dimensional space, to find L unknown variables, thus MLS can "fit" any system to an IR or another. MLS does not provide any basis for evaluation of MLS's validity, therefore LTI violations may result in whatever artifacts.

[7] On a 5GHz Dell 8940 i7-10700K desktop, single threaded



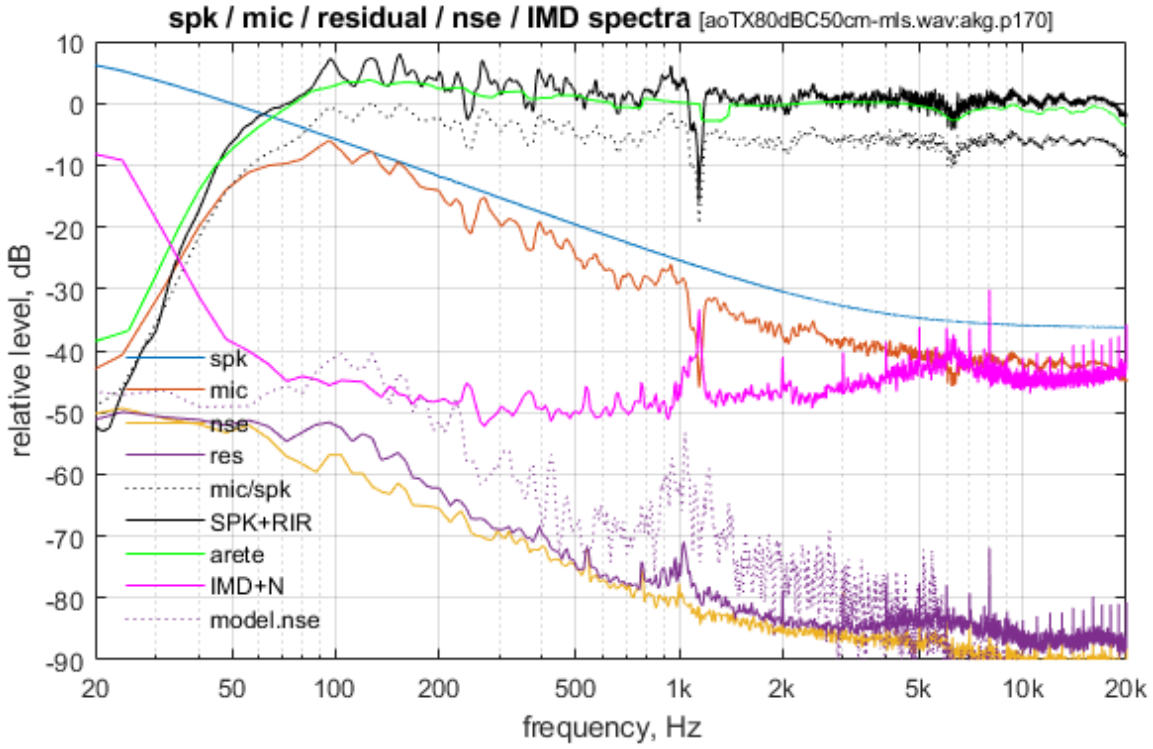


The RIR spectrogram with inversely proportional to the frequency window length (we call it arete[8]) is more informative. The F5 monitor itself is tonically flat, judging by the arete top, which is the property of

[8] Cox, Steven and Fulsaas, Kris (2003). *Mountaineering. The Freedom of the Hills, 7th Edition*. Page 340. Mountaineers, Seattle, WA.

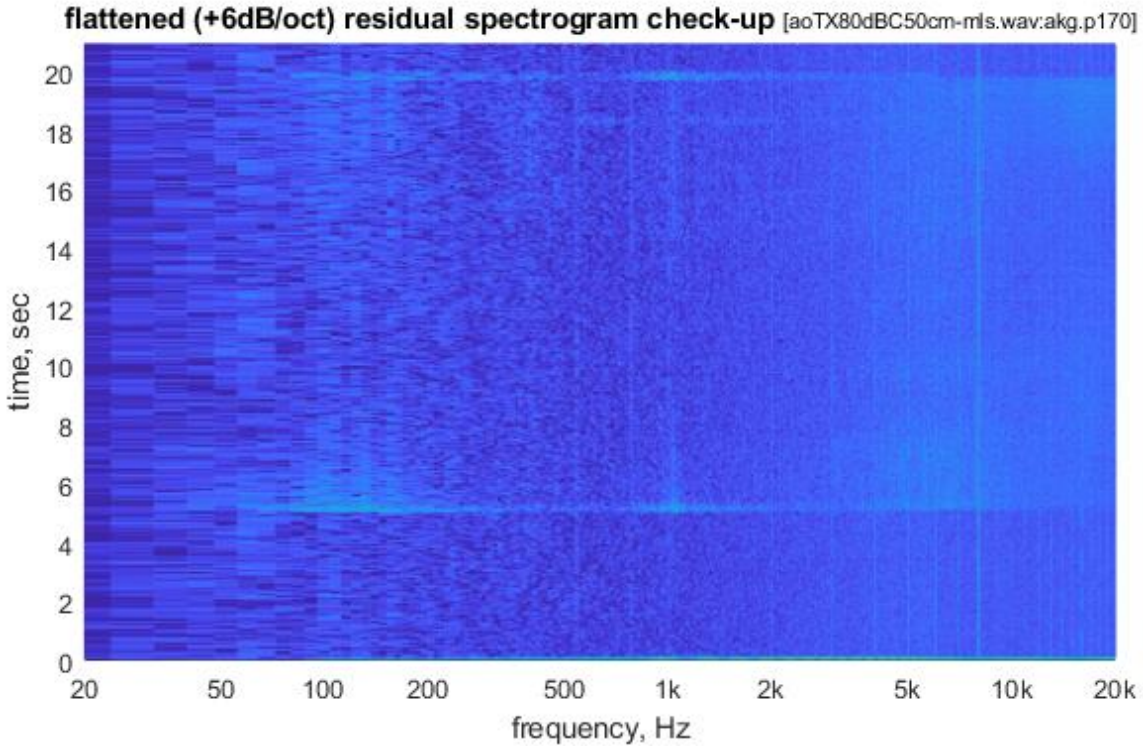
loudspeaker itself – not the room's. The major room resonances, concentrated around 100Hz (λ=3.4m≈11'), are realistic. They are not damp-able by reasonable amounts of acoustic foam.

However, the slow decaying ~1.1kHz resonating mode is debatably a room feature[9].



[9] It decays a way too slow to be of a 'natural' acoustic origin.

The log-log frequency domain plot shows that the -6dB/oct de-emphasis provides nearly spectrally flat SNR. There is a noteworthy +3dB FR bump at ~150Hz[10].



The noise-flattened spectrogram of MLS residual shows that some of the artifacts are sharp, transient, and concentrated on the excitation start/stop edges.

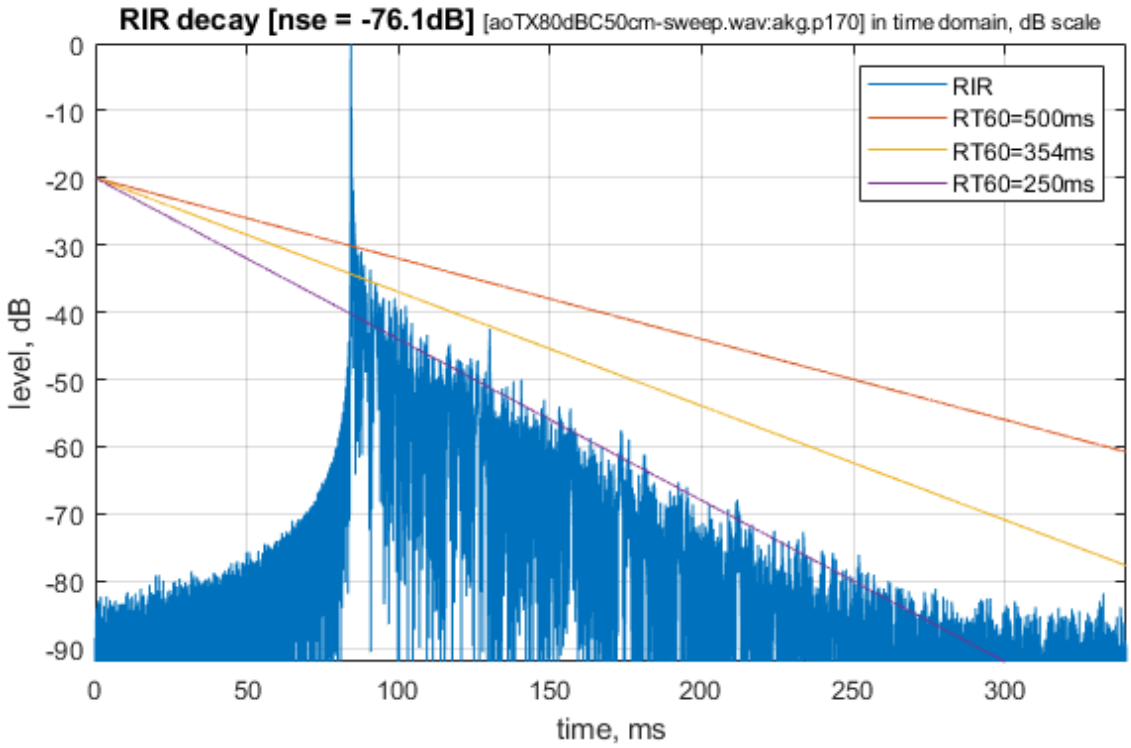
3.2.2 Chirp (a.k.a. SineSweep) observations

Chirp calculations were performed using MATLAB audio toolbox functions and, together with the computation of residual, and took about 0.8…1 sec (same as for MLS).

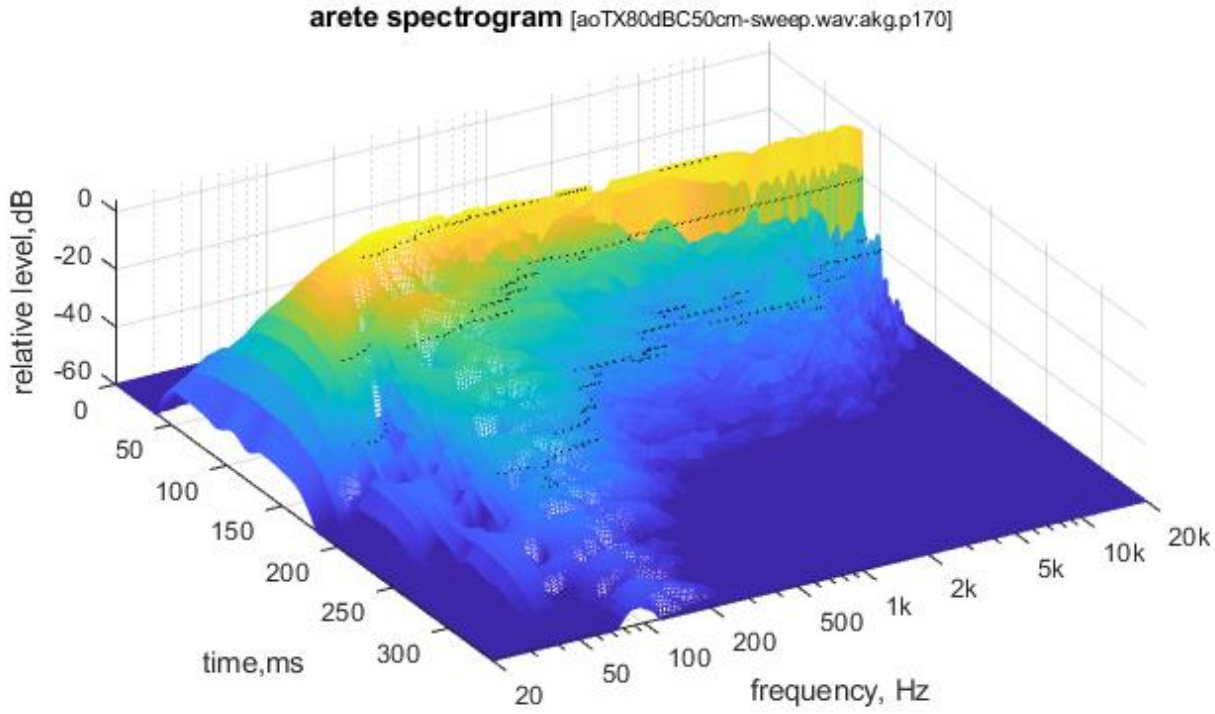
The non-casual 80ms part of RIR contains 23kHz decaying hum, which is due to MATLAB 'brick-wall' implementation of sweep generation and processing. Otherwise, the modelling noise is also time-wise

[10] so helpful to declare a spectacularly low -3dB HPF frequency.

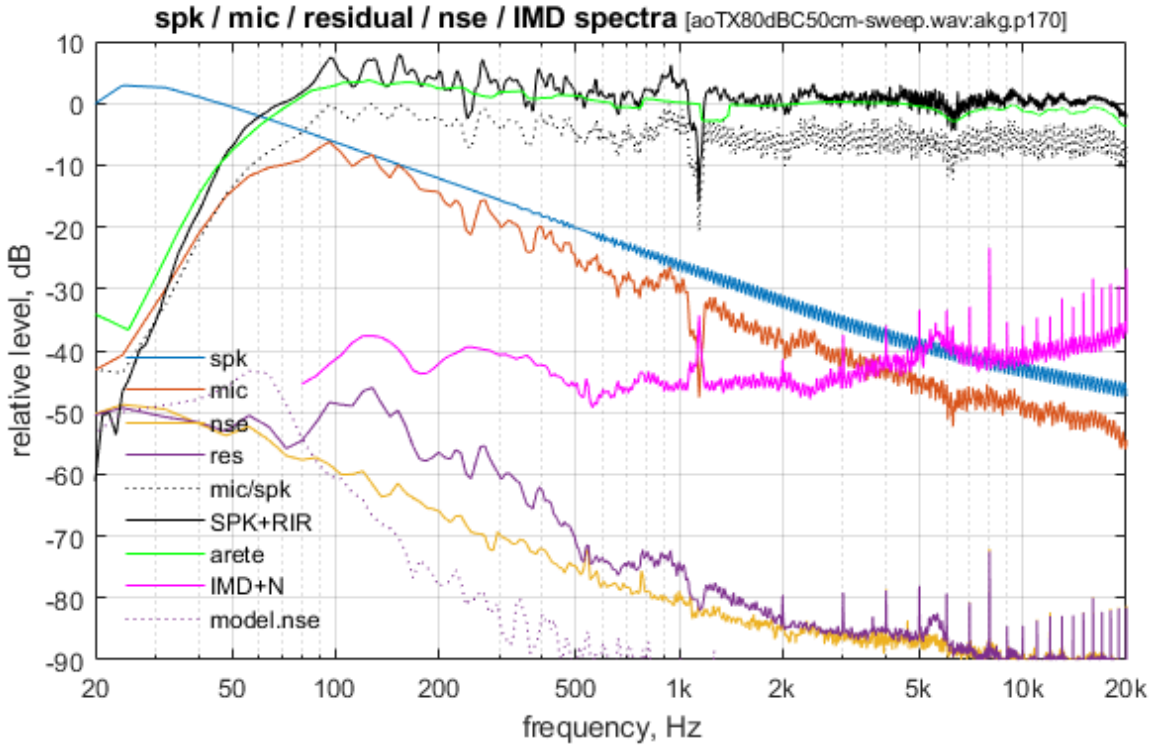
uniform and about 10dB lower than for MLS:



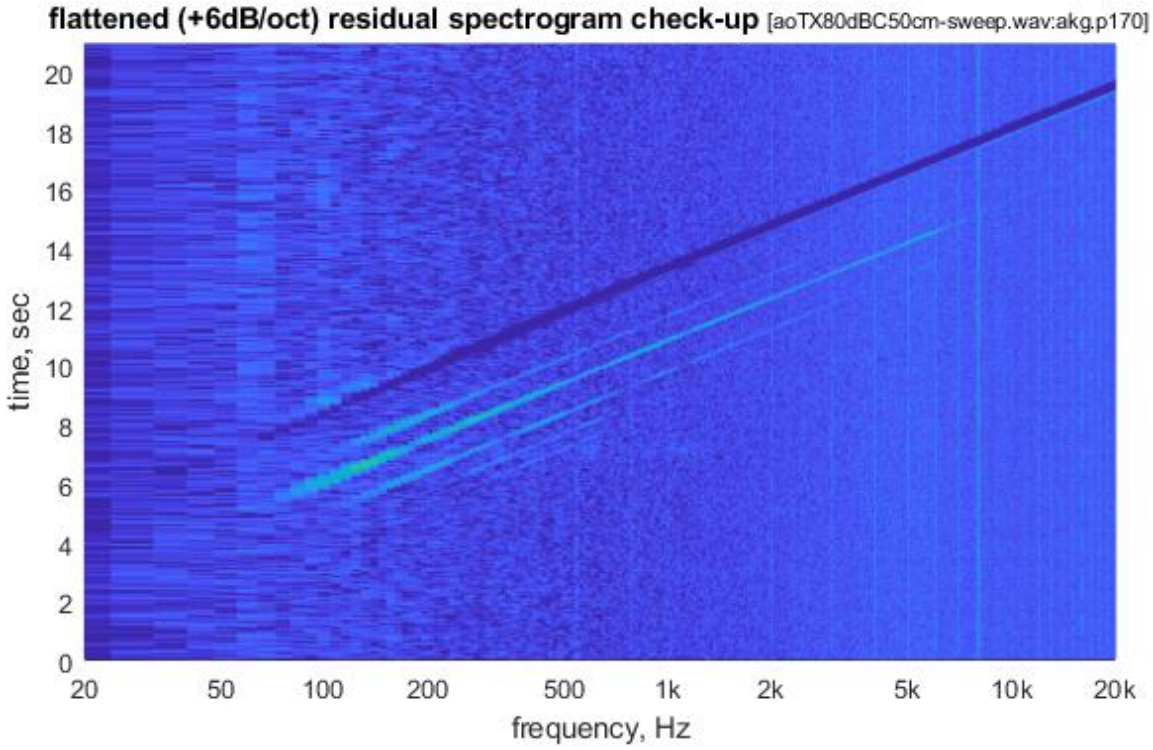
There is no 1.1kHz resonance tail on the RIR arete spectrogram:



The SineSweep RIR's FR is 'visually' nearly identical to MLS' but devil's details differ.



Here, modelling noise is [mostly] below not-LTI residual, which is correct.



The residual spectrogram shows how strongly SineSweep RIR measurement method is sensitive to the noise: see the 0.5sec deep blue trench immediately after the chirp. This can be alleviated by making multiple passes, as for MLS – which, however, may take too much time because you cannot shorten the chirp without losing resolution.

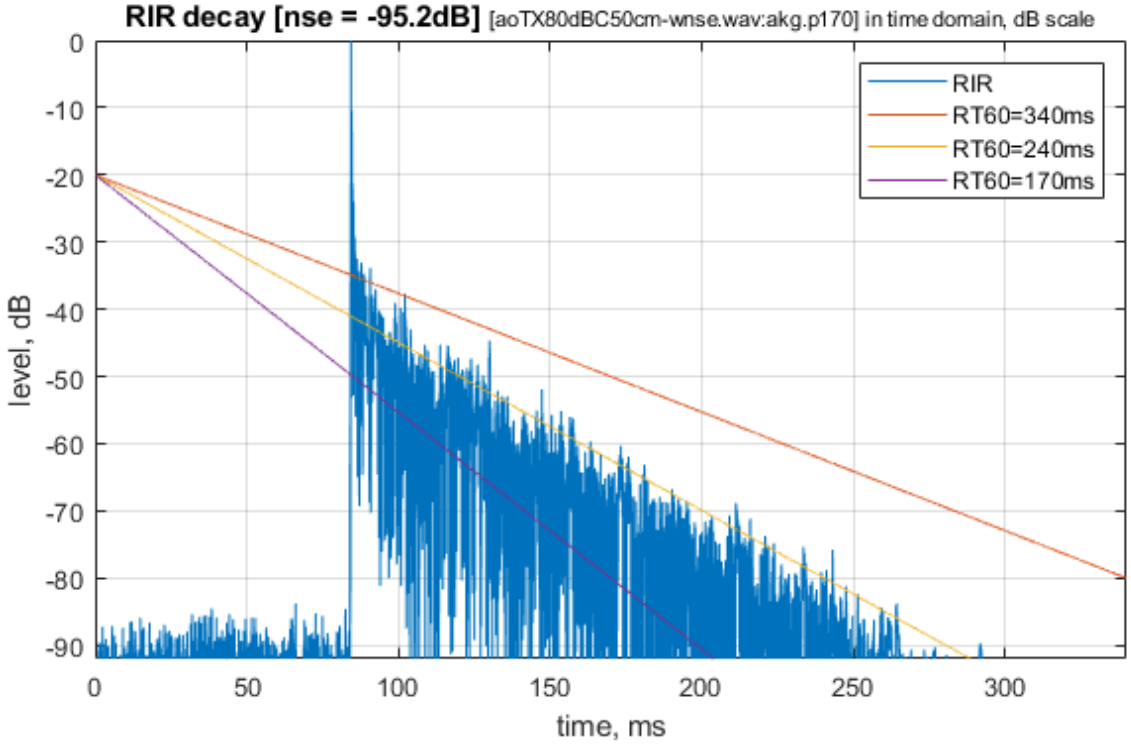
The chirp signal in LADF-dimensional space covers a full-dimensional circle (a thin surface of constant radius), as the last few yards in a traditional yarn ball, which is better than for MLS but still miles apart from comprehensive and/or real-world representable.

There is a clear trace of tweeter heat-up variance and distortions on low frequencies (upper-left corner). There are other indications of non-linear distortions, as the harmonic distortions' lines below the chirp and delayed noise at ~120Hz. The one-to-one relation of these harmonic distortions to subjective loudspeaker's assessment has been covered by many fairytale & fantasy publications.

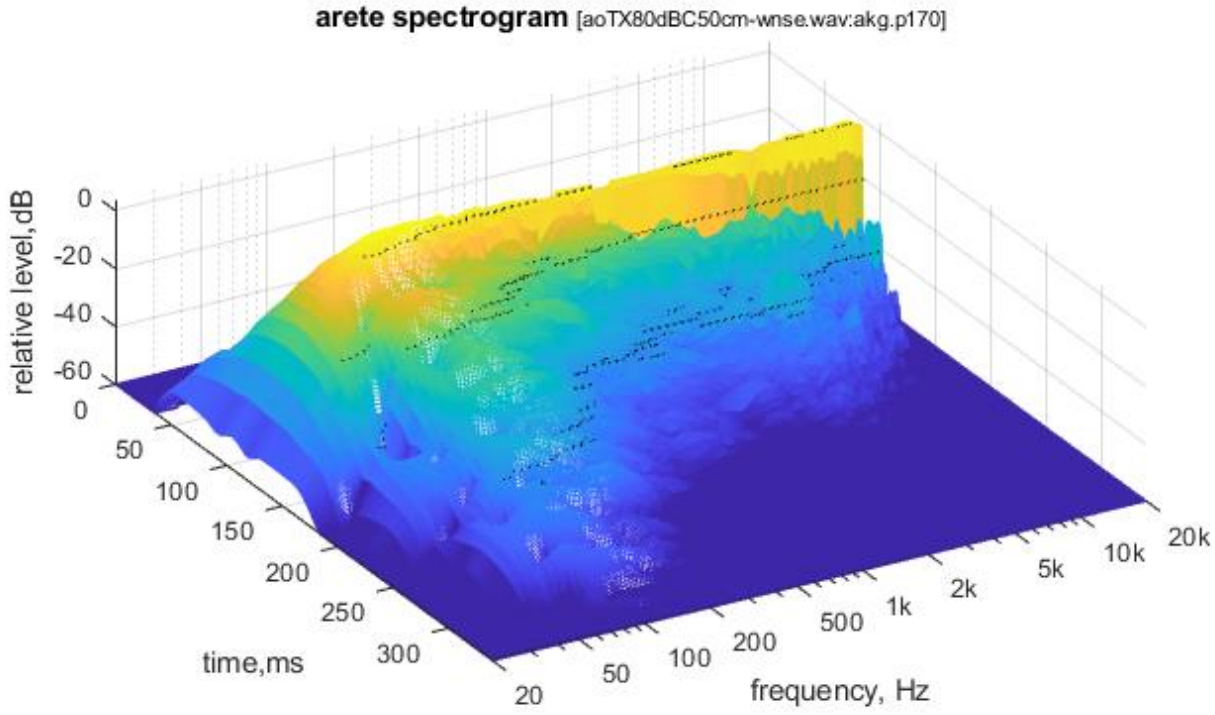
3.2.3 FSAF observations

FSAF uses a 480 bands filterbank with long IN/OUT filters (L=20) and even longer DSF/DSF (LDSF=36). It reconstructs full band IR with precision of about 115dB if no AWGN is present. Due to the length, it takes a long time to design (a night on modern i7). You can redesign the filters or use the existing design as is.

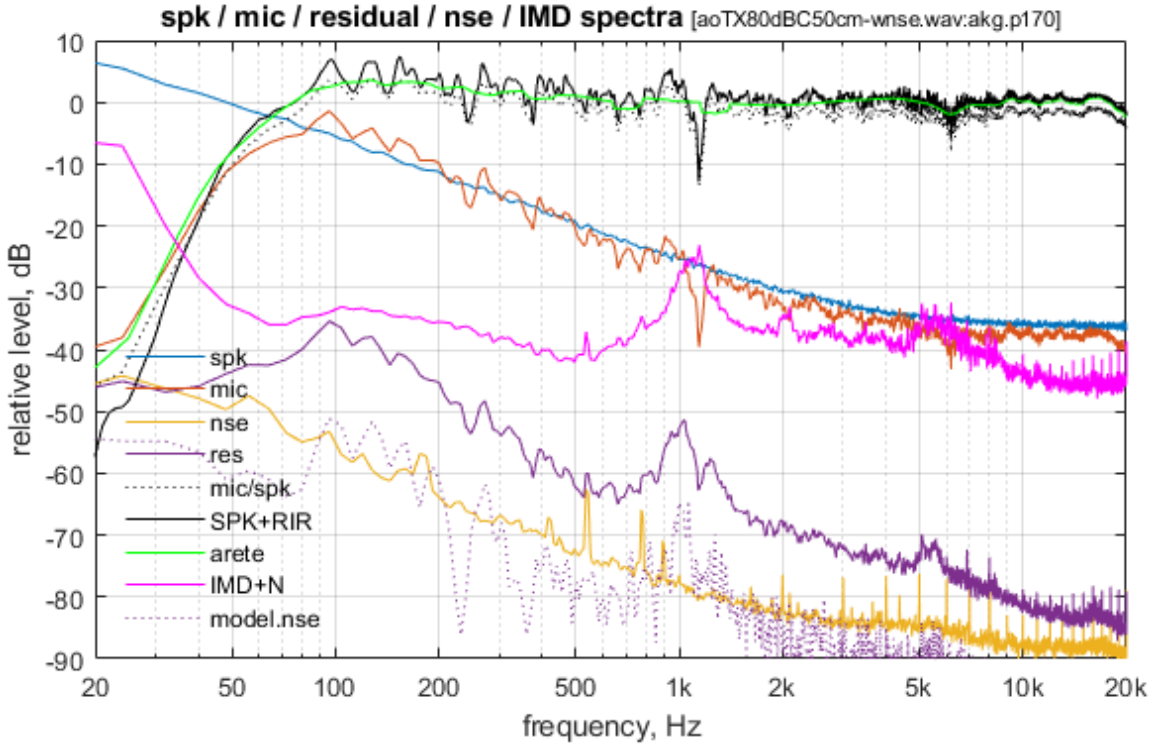
FSAF adaptation on pre-shaped AWGN was performed using MATLAB high-level implementation lacking proper performance optimizations, and together with the computation of residual, it took about 8…9sec, which is ~55% real-time load.



The modelling noise, besides being ~15 dB below MLS's, is not time-wise uniform because FSAF 'knows' that any RIR decays exponentially, and uses that to optimally bias the RIR estimate.

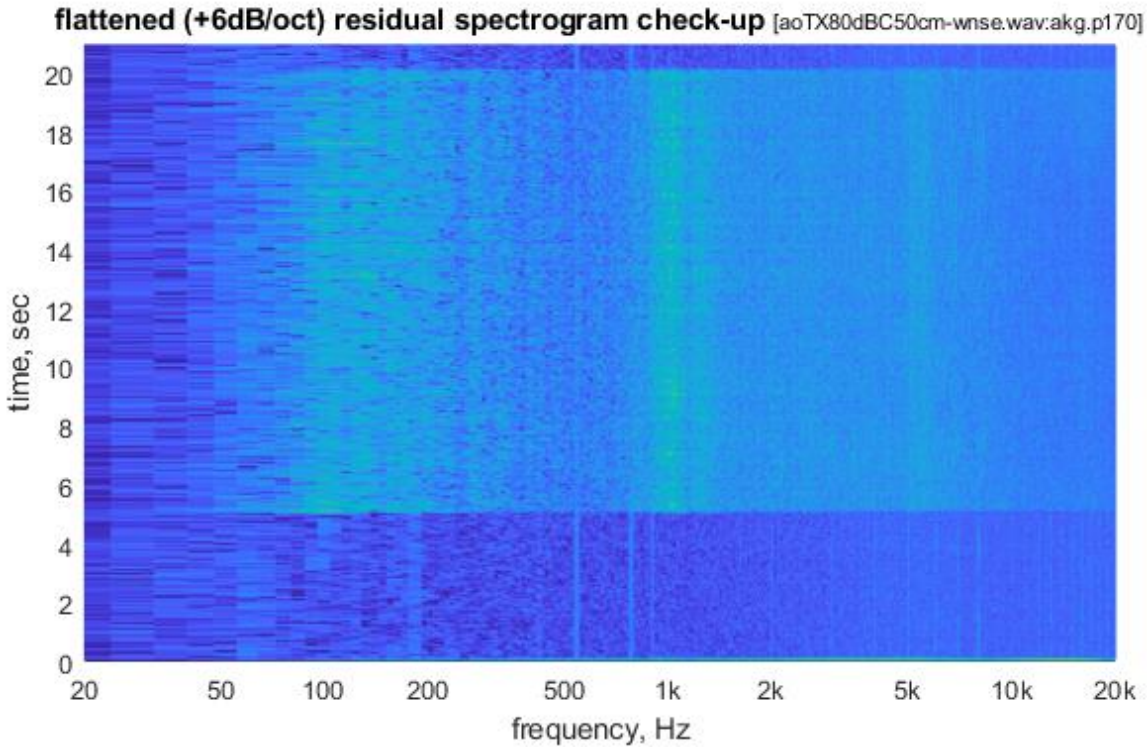


The too slowly decaying 1.1kHz tail of the MLS' arete is [also] absent.



The FSAF residual's PSD is higher than for SineSweep and MLS. AWGN has higher peak to average ratio (~12dB vs 3dB). The AWGN signal in LADF-dimensional space looks like a ball of fog a full-dimensional

circle, thickest in the center and thinning out. It is still not real-world representable.



The FSAF residual' spectrogram is non-stationary at f >10kHz, the lowest in the middle of excitation due to the ribbon tweeters' power [mis]handling. The tweeter efficiency is ~1…2%, and the ability of thin metal film of ~1sq.inch to convert 98% of incoming power into heat radiation shall not be overstated.

3.2.4 Discussion

Ideally, for an LTI system affected by AWGN, all methods shall provide estimates of [asymptotically] the same precision because the corresponding Fisher Information matrices are essentially the same.

In reality, the robustness to the deviations from their assumptions differs. We know that MLS and SineSweep belong to the program control's domain, while FSAF belongs to the adaptive domain and is expected to be significantly more robust.

- MLS's [in]ability to process non-LTI distortions [in]appropriately renders MLS spectacularly deceitful[11] for acoustic measurements.
- Chirp is not useless but very annoying.
- FSAF is inferior by the speed of computation
- FSAF can refer to IMDs as AWGN in each subband because IMDs come from other frequencies.
- FSAF provides estimations of the modelling assumptions.

Note that loudspeaker's odd-order non-linear distortions cannot be modelled by Volterra series approach.

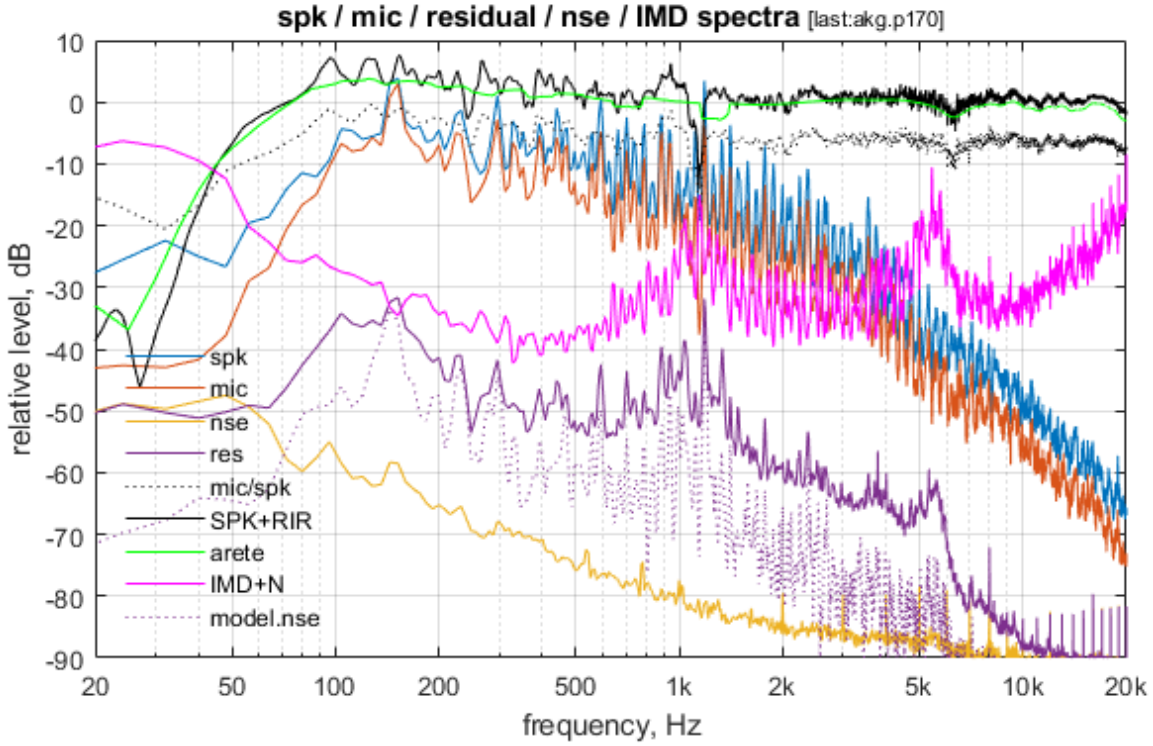
[11] Where the wide-spread belief in insignificance of loudspeaker non-linear distortions is rooted? And, how does it agree with any other facts of life?

3.3 FSAF USING MUSIC AS TEST SIGNAL

FSAF is capable of converging on [any] music[12] [with exclusion of singular spectrum non-vibrato wind instruments].

Here we use 60 seconds from a recording of Mozart Piano Concerto N.20, reproduced on the same 80dBC@1m RMS level. The peak-to-average ratio is 23.5 dB, which is about normal for 'unplugged' acoustic performances[13].

The results could look quite different on other music styles.



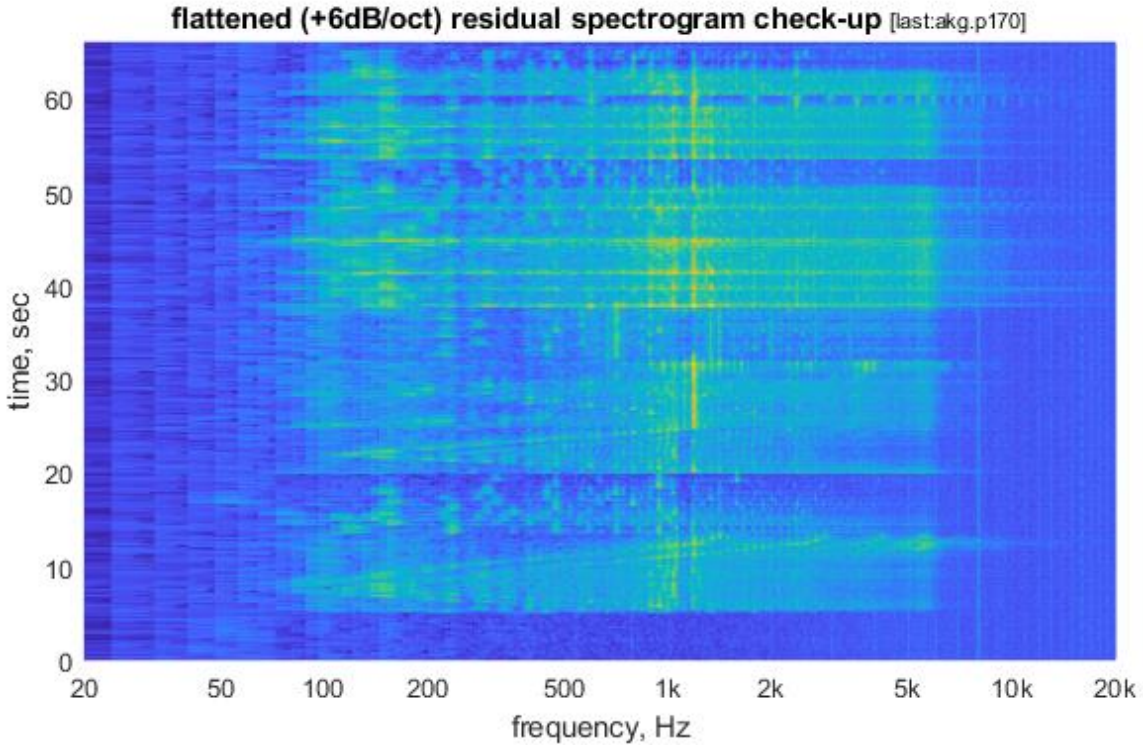
ADAM F5 is not the worst monitor ever, on the contrary, it is relatively decent for its MSRP.

There are few design bugs, like F5's TDA7294 goes into protection even on moderate level sine input in 150…300Hz range (F5 is not alone in doing so) – because the woofer complex resistance drops below 4 Ohms. It is curable by a series 0.5Ohm/5W resistor.

The non-linear distortions indicated by Chirp (SineSweep) are not related to the IMD on music.

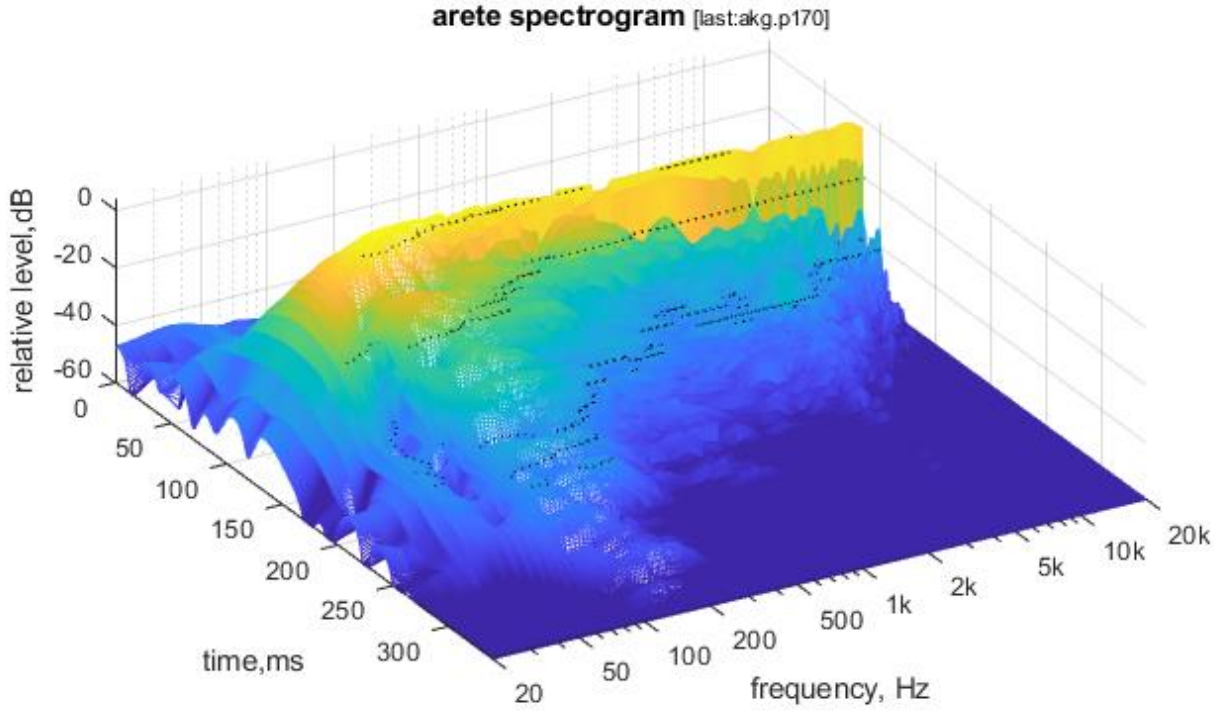
[12] which is the ultimate test signal.

[13] AT&T set telephone reference RMS level at -22dBFS (-19dBm) in 60s, to provide for a natural conversation flow. Nowadays, EU requires TV & radio broadcasts to be normalized to -23dBFS. Internet streaming services… Yet.



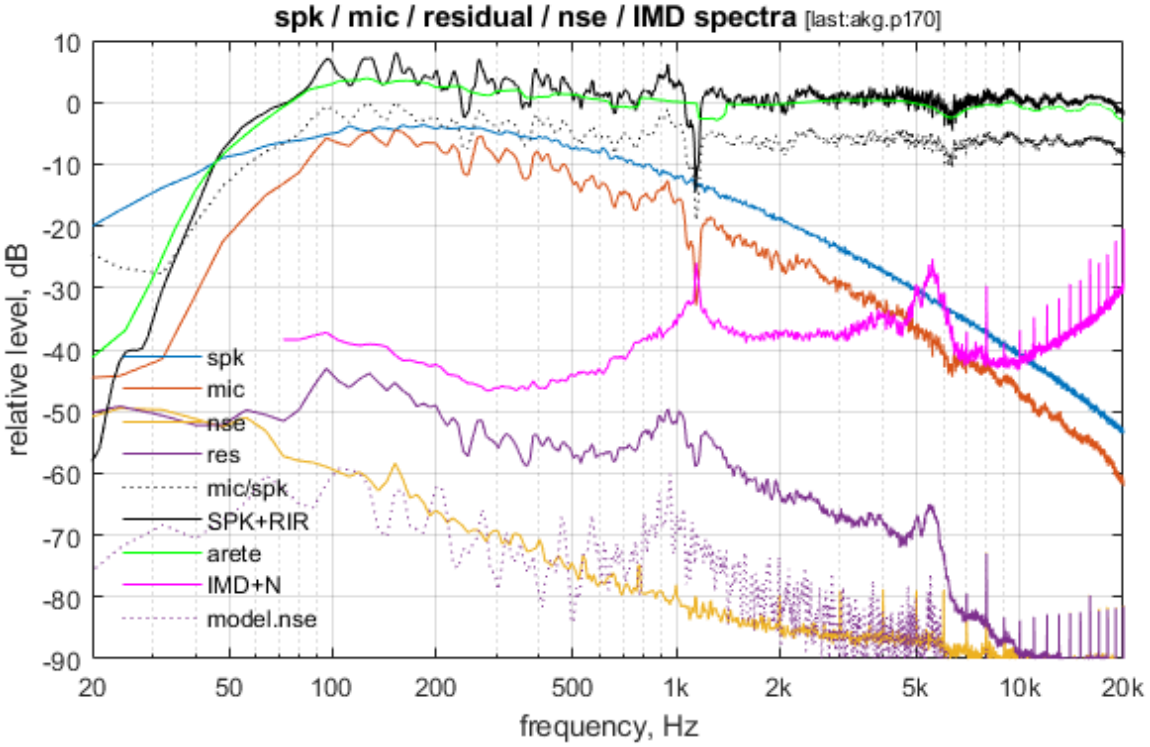
Particularly bad IMD distortion outbursts are due to the peaks in the music dynamic range, when it hits high-water marks, and the worst are on the Steinway piano attacks. Hearing [the residual [yourself [on your kind of music]]] is believing and may help you to realize how poor the 'average' consumer loudspeakers, and why 'wall-of-sound' became de facto recording standard.

The goal of the FSAF LTI modelling of loudspeakers is the extraction and analysis of residual to improve it. Until the residual is satisfactory low, arete plots make limited sense.

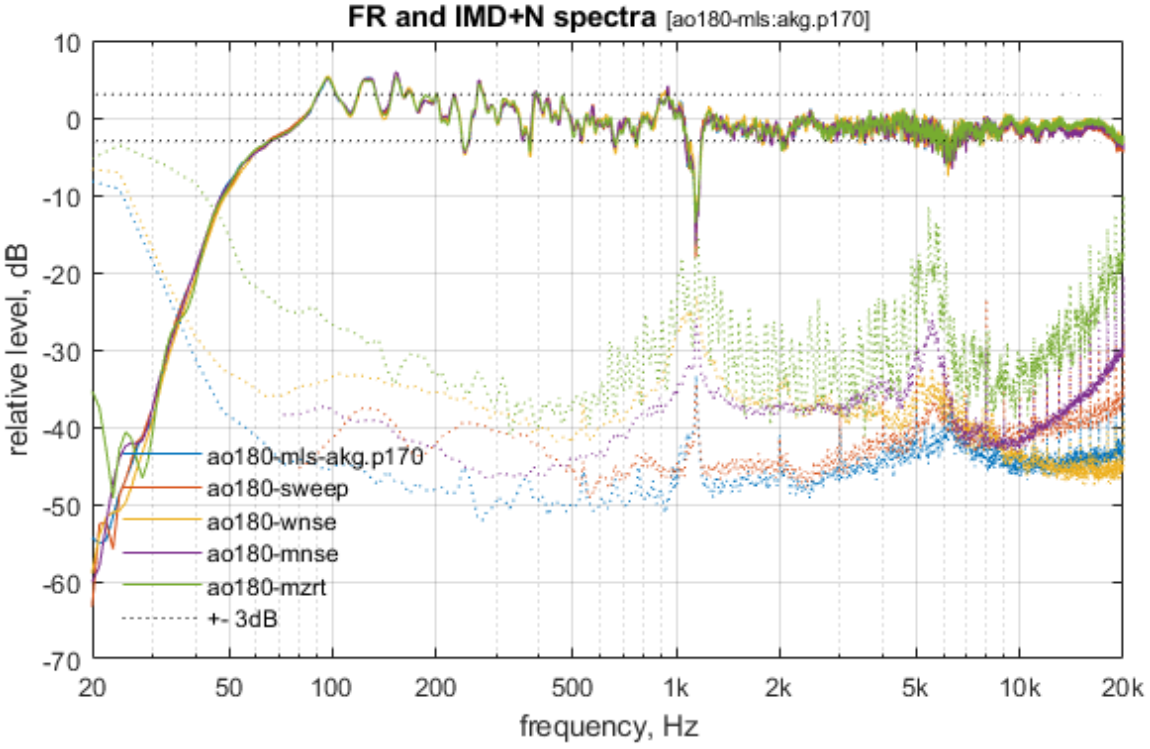


3.4 MNSE AS A QUICK-TEST SIGNAL

Proper music FSAF measurements cannot be replaced by short artificial test vectors because non-LTI distortion's level and spectrum heavily depend on the excitation's spectrogram. However, to quick-test loudspeakers, we suggest using stationary Gaussian noise, spectrally shaped to be music-like.

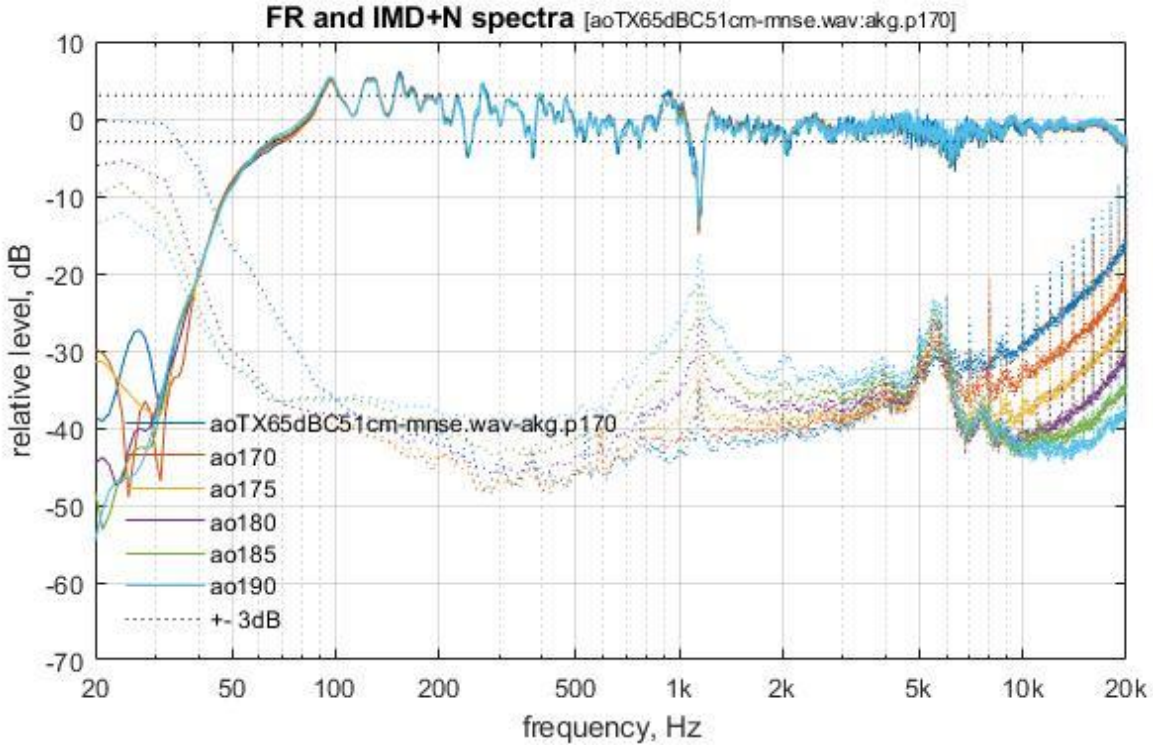


So far, such excitation has produced the non-LTI distortions of the same general shape as for many classical music pieces, but about 10dB lower. I call it MNSE.

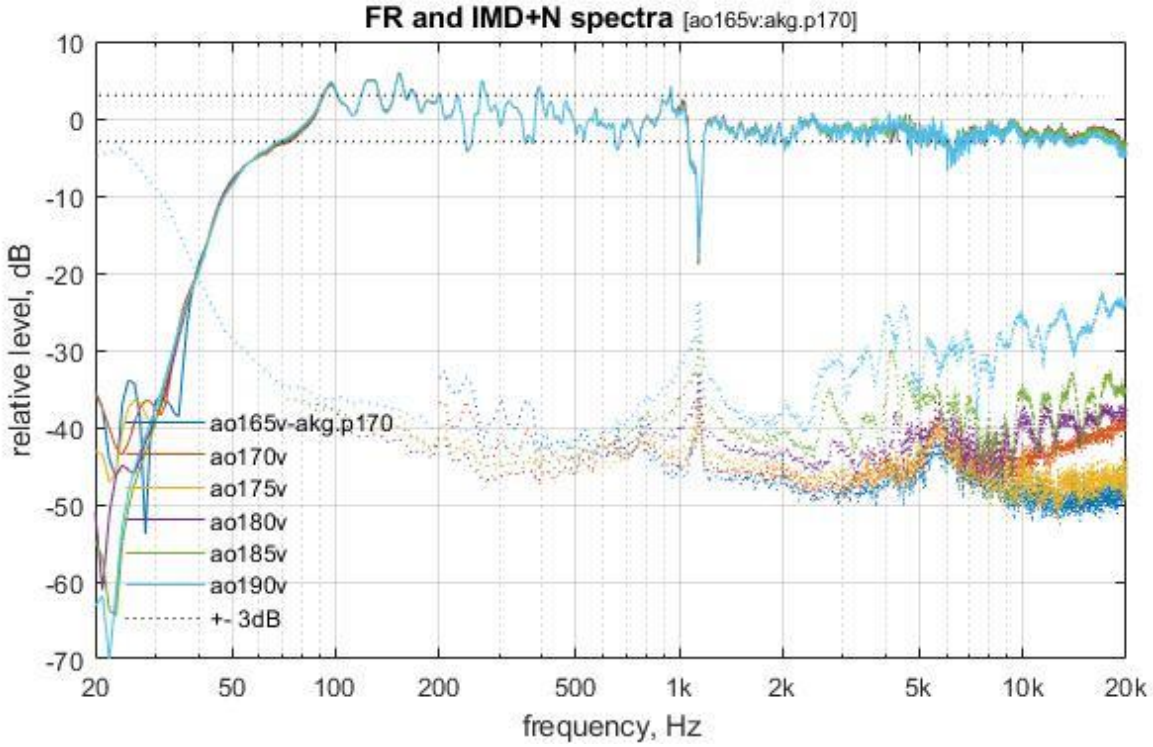


3.5 LOUDNESS VARIATION OBSERVATIONS

MNSE excitation for 65:5:90dBC@1m:



The seemingly illogical MNSE IMD+N rise above 6kHz is proper – distortions drop below the noise floor.

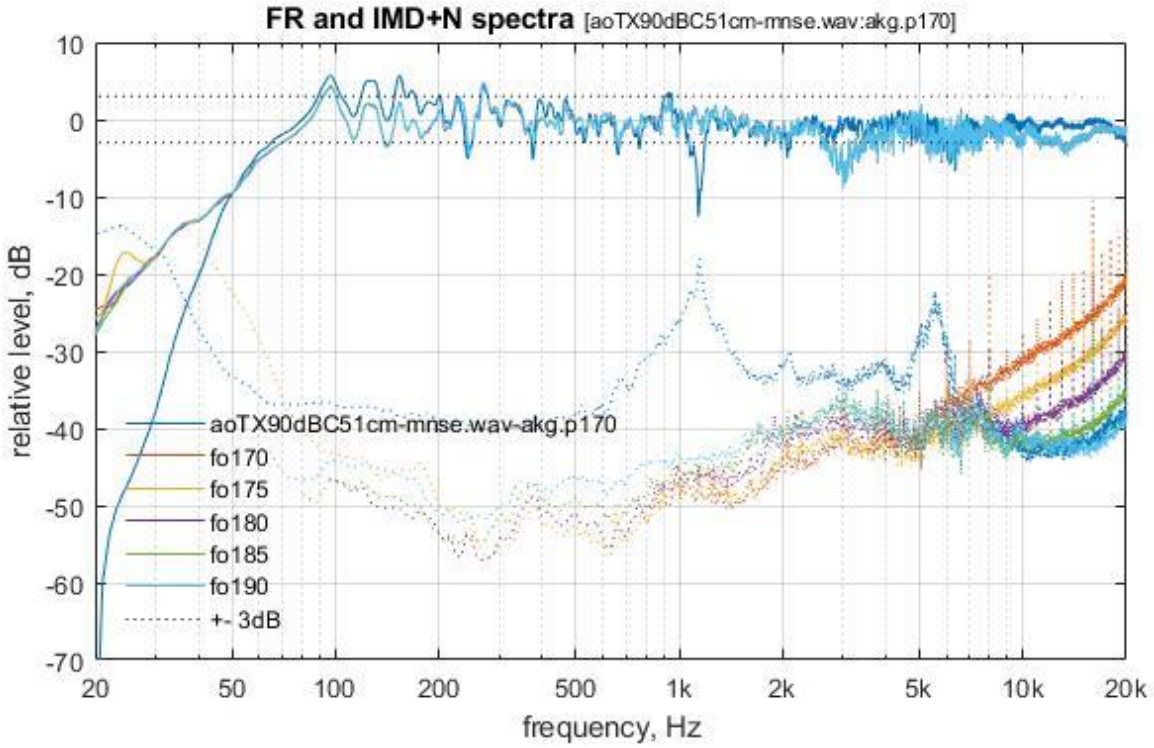


ADAM F5's tweeter fails to handle white noise excitations louder than 75dBC.

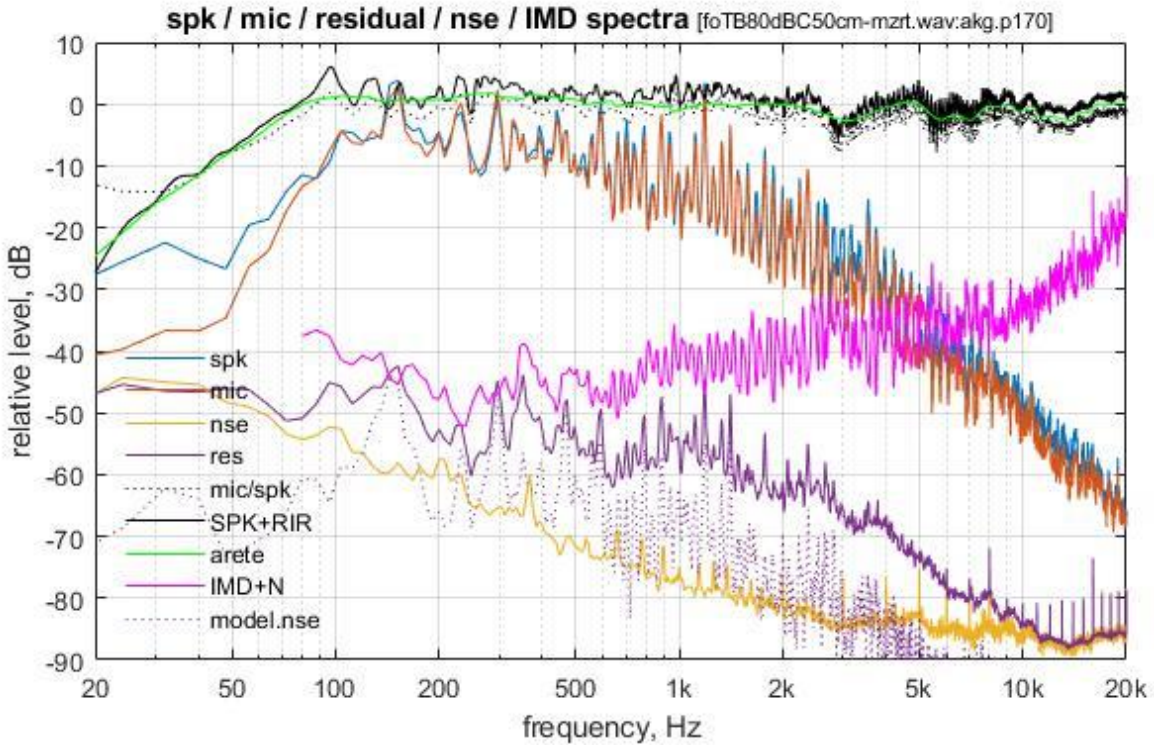
You could have expected the IMD+N distortions to fall at [at least] the double rate with decrease of the excitation level. This happens only around 1.1kHz resonance mode. The rest of the IMD ratio curve falls at the [most] half-rate, ~2.5dB [or less] per 5dB of the excitation decrease.

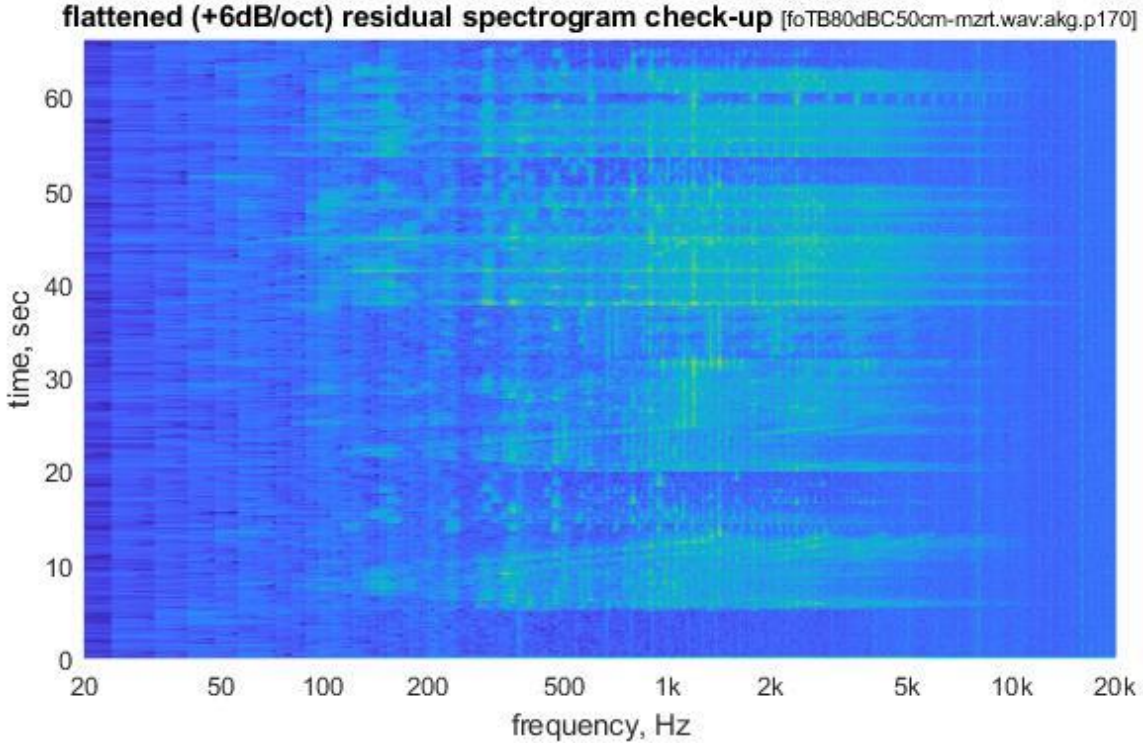
3.6 FOCAL SPECTRAL 918.1 OBSERVATIONS

Here is a [scratched, beaten up and half-broken] 27-years-old Focal's classical 3-ways 4-drivers design evaluated on MNSE excitation at 70:5:90dBC@1m loudness, measured at the same 0.5m distance (which is too close for the massive 918.1, low frequencies don't sum up properly yet). ADAM F5 at 90dBC@1m is plotted for comparison in <blue>:



The Focal 918.1 performance on [the same as ADAM F5] music:



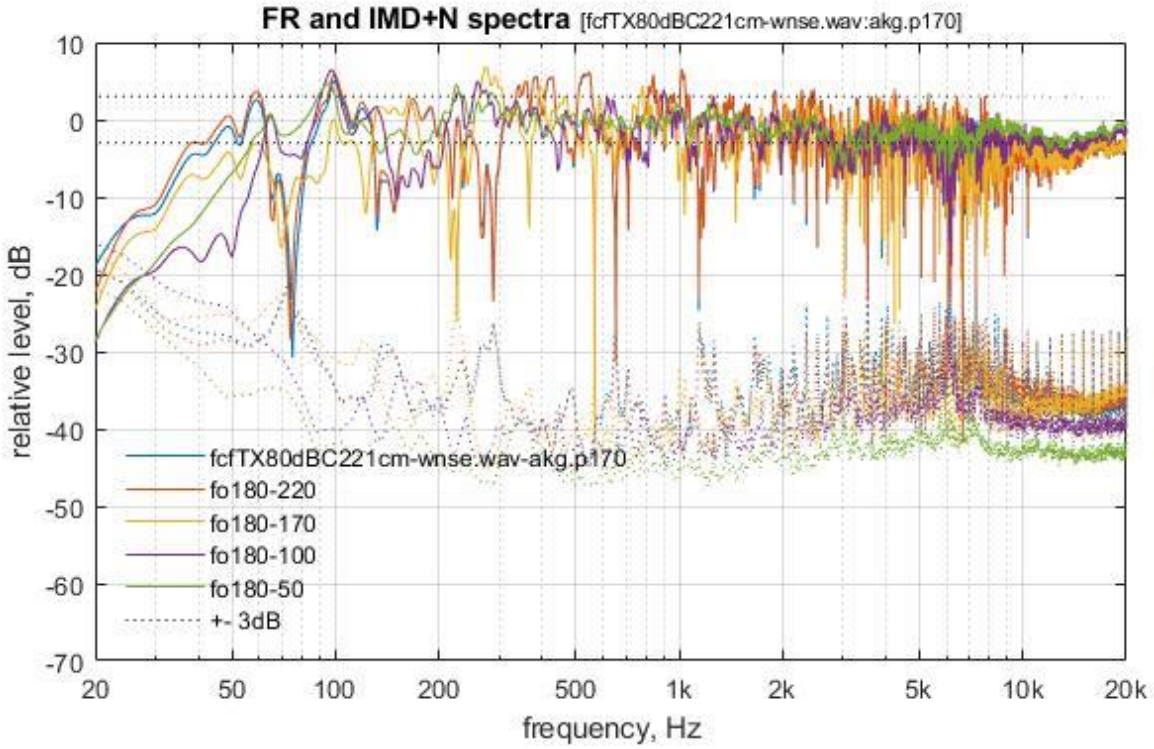


3.7 LOUDSPEAKER LOCATION VARIATIONS OBSERVATIONS

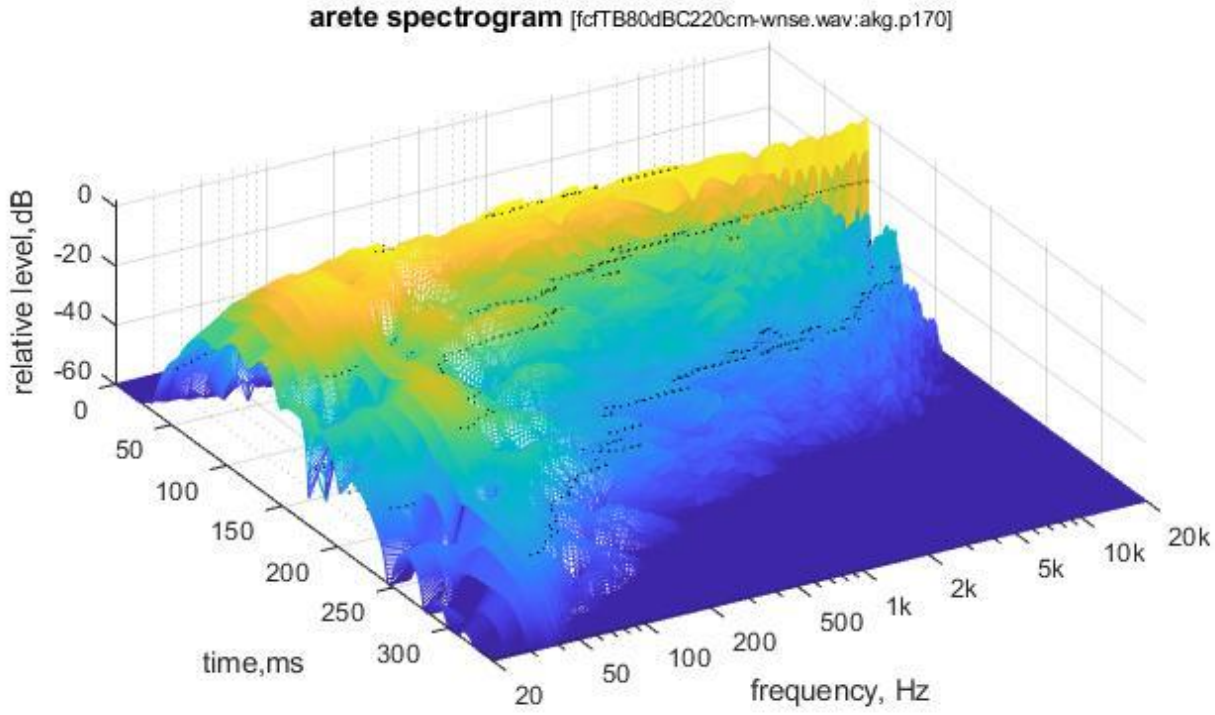
Here, microphones are located at a 'normal' sitting listener position, 80cm from the floor, and loudspeaker is gradually moved away, from initial 50cm to 100cm, 170 cm and finally to the corner of the room at about at 220cm distance. The maximal distance and stereo staging are unfortunately shortened by

unfitting loudspeaker's 30^0 rotated footprint into the room's 90^0 corners, and not being flush with the walls.

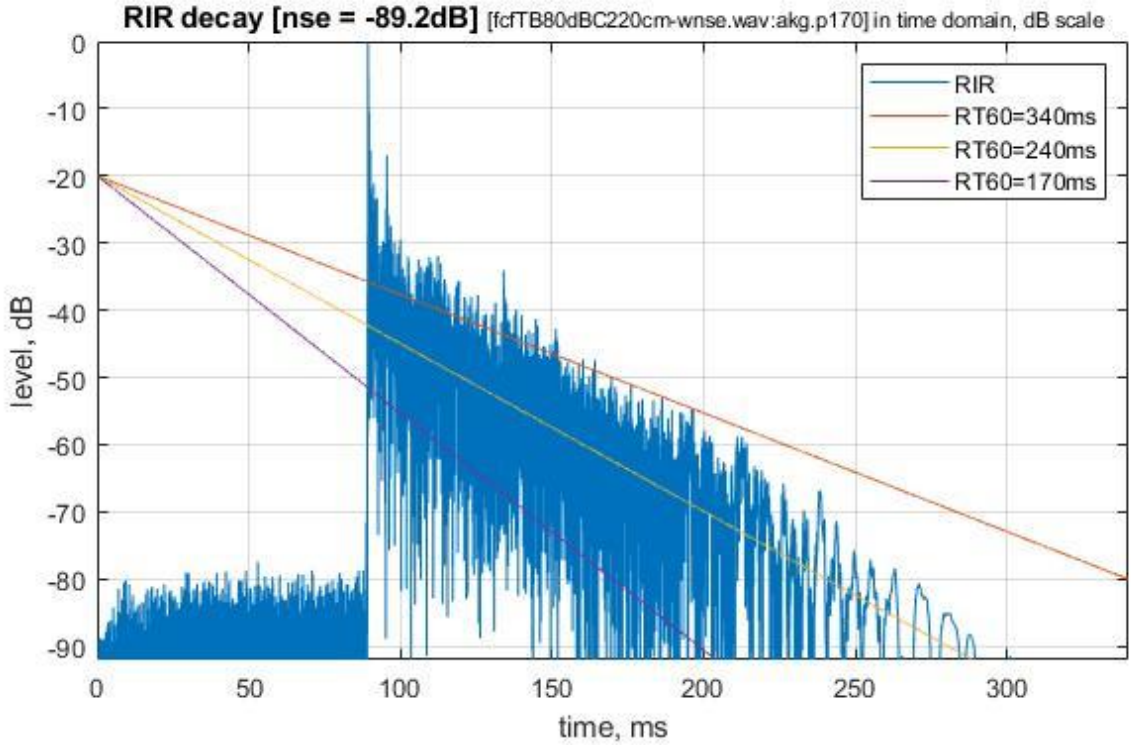
The first <blue> curve corresponds to the 918.1 with the port plugged, which may help to comprehend that in a real living room, loudspeaker's anechoic performance is not of major relevance while the loudspeaker location and the acquired room gain are. The impact of the port is harder to distinguish on the FR plot than to hear it as unnaturally booming (150ms RT60 for 918.1's ~40Hz, even longer for larger ported designs) and distorting (~20dB IMD+N at the port is common)



Below, RIR arete plot for a closed-box 918.1 at the corner. Adequate loudspeaker positioning may help to minimize exciting of not so exciting room resonance modes.



The RIR decay plot starts to make more sense when measured by FSAF:



There is no white [modelling] noise at the RIR tail [as in MLS or Sine Sweep cases] obscuring the reality - which is a special beauty of kernel-based System Identification approach.

3.8 SPKID.M USER GUIDE

You don't need expensive equipment or anechoic chamber to accomplish it. Regular USB audio interfaces have 120+ dB dynamic range which is more than sufficient for the loudspeaker measurements, so fix gains once found.

3.8.1 Living Room vs Anechoic Chamber

The focus on loudspeaker's anechoic performance was due to the inappropriate program control paradigm for sound recording and reproduction. It was assumed that different good recording studios are interchangeable; if good loudspeakers can deliver 'perfect' sound in a controlled environment as an anechoic chamber, they can pass it to the customers, who can, if they wish, get sufficiently good source/DAC, amplifier, and loudspeakers and hear essentially the same sound as in the original recording studio.

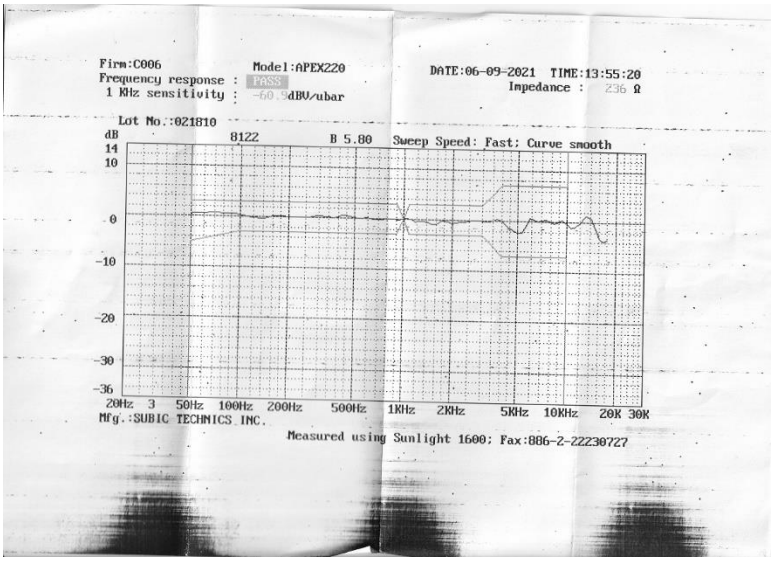
None of those oversimplified assumptions are valid. Moreover, it is absolutely impossible to make a perfect voltage-driven passive loudspeaker.

Loudspeakers are routinely set at the room corners – but they have not been designed for that. The common ported design disallows exploiting room gain effect. Major low frequency room resonance modes are not [easily] controllable, in non-billionaires' living rooms. Good anechoic performance does not translate into a good performance even in a well-damped living room. Moreover, the recordings produced in one studio do not sound much, if any, similar in another good studio.

You don't need 'universal' loudspeakers for ideal anechoic chambers, instead, adjusting them to your specific room and specific locations makes much more sense.

3.8.2 Microphone Calibration

Reference B&K measurement microphones are out of reach for most DIYers. Cheap measurement microphones use 6mm omni electret capsules, with ~30dBA noise level, max input well below 105dB SPL, regardless of the vendor claims. Any attached calibration plots (like one below) are most likely fake.



You cannot use electret measurement microphones for FSAF, they are too noisy and non-linear. So, you'll need a second, normal condenser cardio ½" pencil microphone, and to calibrate it to match previously calibrated [your] measurement microphone. You don't really need to cover its side ports to convert it to omni but mind the proximity effect.

A 6mm omni physically cannot have FR bumps with say, 1kHz period. That frequency corresponds to wavelength of 340mm. There is no room inside the microphone to have any resonances on such frequencies. These microphones are perfectly flat in low frequencies and have a sole bump around 10…15kHz. The position and level of this bump is not repeatable for any omni capsules' vendor, +-3dB variations shall be expected. There is no real difference between $20 and $200 measurement microphones.

The only realistic reference is a loudspeaker measured with a true measurement microphone by a reputable vendor[14] or reviewer. Decent loudspeakers are highly repeatable and reliable devices, you can trust them. To calibrate your measurement microphone, you need only adjust the aforementioned high frequency bump with a 2nd-order IIR filter, without touching low frequency response, intentionally avoiding any high order filters. The windowed measurement technique shall work perfectly.

You will need to correct fsaf_spkid.mic_eq_sos() function manually for your microphones. Use spkid.mic_eq_plot() function to see the results. The doc_p409.m script contains the entire ~10 lines of the test code you'll need.

3.8.3 Noise and timing

Room noise is overstated. Later in a day, street noise almost dies out, and your refrigerator becomes the loudest noise source. Between fridge cooling wake-ups, microphone's internal electronic thermal noise becomes the only noise. Sometimes, a loud car passes by, and you may need to repeat that particular measurement.

50/60Hz hum is almost always a problem.

USB noise at 1kHz*N may be a really bad problem in some setups and not a problem in others, especially with some laptops running from a battery.

There are many advices how to test for and avoid PC latency problems. Most relevant are written by your audio interface vendor. Audio interface shall be set to 48KHz / 24 bit, 10 or 20ms buffer size.

You need to always observe the residual spectrogram plot. If any timing problems with skipping frames are still present (as they will from time to time), you'll see too bright yellow strips.

3.8.4 Repeatability

The task is to ensure the repeatability of the measurements, and ability to recalibrate quickly and easily at any time, is always much harder than you anticipate. Every detail matters, and there is always one you forgot about.

3.8.5 Functionality

All user-callable functions are "help"-ed in the code.

Config:

```
% config
% MicDly    = delay microphone signal so that at least 80ms
%            preceed the RIR main spike [0]
% FS         = sampling frequency [48000]
% Tstart     = length of pre-silence, sec
% Twarm      = do not adapt for X, sec
%            do not use it with MLS/sweep because Matlab does not
%            work properly with it
```

[14] like B&W / Focal / Genelec

```
% Tend      = -Tstart = length of active excitation
% TADF      = length of adaptive filter
% Color     = de-emphasize on spk and
%             inversely pre-emphasize on mic.
%             usually, white noise as the src.
%             param is:
%                 'flat'
%                 'brown' -6dB/octave
%                 'pink'  -3dB/octave
%                 'class'  - classical music like spectral shape
%                 'rock'  - as in queen's ... dust
%                 'p50' - speech, as in ITU-T P.50x
% RoomNseF  = pre-emphasis for adaptation for noise flattening
%             param is max freq of pre-emp. always 6dB/oct
% AudioIf   = how your audio interface is named in ASIO
% AudioX2   = some Audio IF do not use a proper ani-aliasing
%             filters, and it messes up the measurements. with
%             this option set, the spk excitation is upsamples to
%             96kHz and properly filtered manually
% MicCh     = which mic to use
% MicName   = name of the mic on this channel
% XSPK      = if char: file name where spk exciation is saved
%             else: excitation itself [vector]
% HPN, LPN  = order of filters used for crossover emulation
%             [3]. can be set in real time in o.playrec();
% all other configs go down class hierarchy. if you mistype,
% you may (or may not) enter a valid config for a sub-class.
```

Some audio interfaces (as Focusrite 2i2 G3) do not apply proper ani-aliasing filters when they operate on Fs (say, 48kHz) less than maximum (say, 384kHz). You notice it by 1) noise floor not going down at fs/2 2) a weird increase of residual towards fs/2.

This is an illustration of essentially the same problem FSAF is addressing. The easiest workaround is to do explicit anti-aliasing by ourselves.

There are self-explanatory MLS, Sine Sweep, and White Noise generators saving their output to a file. Do not regenerate them into the same directory.

play_rec:

```
% the function plays out excitation preceeded by silence
% and records mics attached to Audio IF
% by default, 2 channels (=stereo). there shall be possible to
% record more mics because ASIO is used.
%
% the mic selected by micCh is 'remembered' and pre-processed
%
% options:
% - 'filename'='fname'='file'='name' -> where to save mics
%   the recordings is saved (default=work_path/lastmic.wav)
% - 'gain' -> set gain in dB of the reference excitation
%   so that you don't have to adjust any volume knobs and
%   recalibrate
% - 'HPF'='FHP' - frequency of optional high-pass filter
% - 'HPN'='NHP' - order of that filter (butterworth)
% - 'LPF', 'LPN' same for low-pass filter
%   default filter order is 3
% - 'DC' - 5th order HP on provided frequency. usually 30Hz
```

```
% 'ChMap' – which channels of AudioIf to use. They may be [1 5]
%     as plugged in (to avoid phantom power on second mic).
%   these filters can be used to choose crossovers for a
%   loudspeaker under test
```

mic_read:

```
% read previously (or externally) made mic recordings
% micCh = mic to use
% fname = file name contatining the recordings, default =
% lastmic.wav
% same options as for play_rec();
```

mic_make: for self-testing

% fname = file name to contain the recordings,

% nseDbFS - db(std(added brown noise))

% rirname – use RIR from a processed recording

fsaf_adapt:

```
% adapt using FSAF (Fast Subband Adaptive Filter)
% default in-subband method - kernel-based ReLS
% NseEq = equalize spk and mic to make mic nse flat,
%   1st order Butter hipass,
%   NseEq=high-pass filter frequency or 0 if no eq
% tailAtt = what do we expect RIR level will be at the end,
% relative to the peak, for the chosen TADF [60]
%
% the function assumes that non-LTI effects are small and
% that noise floor is the same as in pre-silence
%
% the function provides the RIR estimate and residual error
%
```

fsaf_readapt

```
% adapt using FSAF (Fast Subband Adaptive Filter)
% and knowledge from the previous adaptation
%
% the function considers non-LTI residual error as noise
%
% N - number of passes to readapt (default = 1)
% idx1 - the start index in the subband respesentaion of
% adaptive filter to start with to learn RT60 on per-band basis
% idx2 - the end index ...
% to identify idx1 and idx2, call o.rir_fsaf_plot()
% make idx1 = idx(max) + 2-3
% make idx2 < idx when echo tail flattens
% experiment with idx1, idx2 for the best results
% idx2 must be > idx1 to activate the weighting
% judging by the min residual error
% default (idx1, idx2) = 0 (inactive)
```

The similar functions for MLS and SineSweep (Chirp)

Plotting:

- Mic eq
- RIR in time domain, and up to 9 RIRs in time domain compared
- RIR differences
- RIR (Y axis in dB) in time domain, and up to 9 RIRs (Y axis in dB) in time domain compared
- Arete
- RIR in frequency domain: FR itself, spk, mic, nse, residual, top-of-arete, IMD, and model noise, and up to 9 RIRs in frequency domain compared, you can choose which details to plot in solid and dotted lines.
- Save % wavs - save computed curves, residual, linear response, and excitation for any future use.
- Signals: mic, lti and LTI distortions
- Crossover compared: residuals of up to 9 pairs of speakers (or frequency divides)
- Sub-band ADF in Y as dB, helps to troubleshoot
- Residual spectrogram
- Sine sweep plots and comparisions

Music analysis:

```
% the function analyzes music given in <fname> and plot the
% results on figure <figno> and <figno+1>...<+3>
% left channel only
%
% results:
% 1: spectrogram
% 2: selected quantiles from the spectrogram's histogram,
%    such as average, 95%, 99%, 99.8%, max
% 3: tweeter [relative] energy, for various crossover
%    frequencies and orders (butterworth)
% 4: woofer -"-
% 5: mid-woofer -"-
% the 3:5 plots can be disabled if <stop_early> is set
```

3.9 SUMMARY

Generally, it's worth checking any method of measurements for the appropriateness of their underlying assumptions, either implicit or explicit, in the given circumstances before such methods are brought in common use. Math is not universal and, ideally, you should understand the math you are using.

In the first half of XX century, it was very fashionable to call any study a "science" due to enormous success of physics and mathematics[15]. If you presented a few pages polluted by formula you were surely a scientist, and all the honor to you! It was not important that a discipline was unscientific whenever it contradicted to a single fact of nature or another proven scientific theory.

In "audio science", people even succeeded to define an "absolute threshold of hearing" and measure it. Well… hearing is adaptive to background noise level. After a few hours in an anechoic chamber, people start hearing how blood streams along their veins and become ashamed of it. The hearing is a subconscious process[16], you cannot ask questions aimed at the conscious level and think that the answers will make any sense.

[15] Similarly, it's fashionable today to call anything a "system". Even a combination of a sleeping bag and a sleep mat became a "sleeping system".

[16] Once upon a time, being a student, I rented a room in a century-old building. After a while, I started to get precognitions that somebody is going to call me on a [land line] phone, a few seconds prior to the actual phone

However, imho, so far it has not been constructive to criticize the more than questionable acoustic measurement methods without proposing something better instead. Also, it has not been constructive because “audio science experts” have been intolerant of opposing opinions and reacted aggressively with Lysenko-styled accusations, humiliations and personal attacks on the bearers of such opinions - even on Esa Merilainen with “Current-Driving of Loudspeakers”. Excuse me, Lorentz actuators with sub-nanometer precisions are the work-horses of semiconductor fab industry, and they are adaptive feedback-controlled current driven.

Unfortunately, incorporation of modern Lorentz actuator-like technologies into loudspeakers is well beyond the capabilities of “audio science”. You cannot simply glue an accelerator on a diaphragm without running into diaphragm resonances on higher frequency and the feedback loop ringing; the sensor has to be integrated into the driver, closer to the voice coil and untied from the resonating diaphragm on high frequencies. But that requires true understanding of the science involved...

I am profoundly confused by the very existence of “audio science” which disregards contradicting to various experimental data reported by abundant independent observers. Audio industry is of the same kind but even worse. AES. Audiosciencereview.com. Stereophile. Directional oxygen-free loudspeaker cables for $1,500. Differences in digital audio cables - YouTube . Volume wars. Recordings of baroque music with different pitch of solo mezzo-soprano voice in left and right channels, etc. I fail to distinguish it from quackery.

It seems that normal science exists only in the areas (alas, audio is not one of them) relevant to military. Then the state supports long term research activities, and they result in, say, digital imaging sensors. Then commerce is good at making money from such byproducts of fundamental science: consumer digital cameras, 4K video, 4K HDR monitors and TVs. Otherwise … A few chief R&D managers of pro audio companies told me, explicitly: “There will be no research in this company while I am here”. And it was not.

I am not aware of a foundation for any optimistic expectations about commercial audio’s future but I hope that FSAF/fsaf_spkid.m could help audio DIYers in their endeavors, and professional[17] audio retailers to prescreen[18] studio monitors and/or loudspeakers to serve audio community with integrity.

---

ringing. A bit later, I even started to “foresee” who is about to call me - if I knew that person. It genuinely puzzled me because my precognitions were 100% correct (while knowing that all claims of paranormal were found BS so far). I even started to answer the calls with “Hello, X” before X introduced him/herself, startling and frightening X. Once I joked about my “superman abilities”. Alas, that joke costed me dearly.
The precognitions stopped after I moved to another place. Only years later, I discovered that the central telephone station in that district was experimenting with Caller ID sent before the first ring as DTMF tones. Somehow, these tones must have been be leaking into handset. Consciously, I did not hear a thing. Subconsciously, besides mere hearing the tones, I was able to distinguish DTMF “melodies” from people who called me often.
Also, I learned that I was not unique in any sense.

[17] It was said long time ago that because ignorance is bold and knowledge is reserved, the objective truth, as the world goes, is only in question between the wise, while the marketing does what they can and the sciences suffer what they must;-(

[18] Purifying selection is the most prevalent form of natural selection as it constantly sweeps away deleterious mutations that are produced in each generation (c) ncbi.nlm.hih.gov

4 OTHER APPLICATIONS [407]

Sometimes, we need to recover a signal well hidden under an "echo" of a known excitation. We may not be interested in the signal itself but only in its general properties.

- In a restaurant, we need to measure the people babbling level to adjust the background music level so that nearby tables are somewhat private.
- In a car, we need to measure road noise level, separate from the car audio music, to adjust the music volume appropriately.
- In airports or other public area, we may need to adjust paging volume automatically per room so that people hear the announcements but not blasted by them.

… And so on. FSAF can remove the echo from the mix as far as decision-directed step size control is set-up appropriately.

Acoustic Feedback Canceller (AFC) has much affinity to ARC. The in-out latency can be nullified by running adaptation in subband (whatever way) and using DCF to convert subband RIR translations back to full-band.

Acoustic Noise Canceller looks as a prime field for the application of OLD FSAF.

The acoustic properties of a theatrical venue depend on the number and distribution of people there. You don't know it until people come but you don't want to bombard them with loud annoying MLS or chirp. FSAF can identify RIR on soft background elevator music.

Let's reiterate that FSAF audio performance depends on applicability of LTI assumption which almost solely depends on the loudspeaker quality. Better XXI-century-worthy loudspeakers are long overdue.

FSAF may help with real-time adaptive filtering problems related to radar / sonar processing where a typical IR is sparse, with few small spikes along a very long IR, on high FS, in presence of high noise levels.

Obviously, FSAF can lower the LS's MIPS requirement quite drastically, *~1/M²* and proper noise-weighted ReLS is much more robust and BLUE than convolution-based approaches.

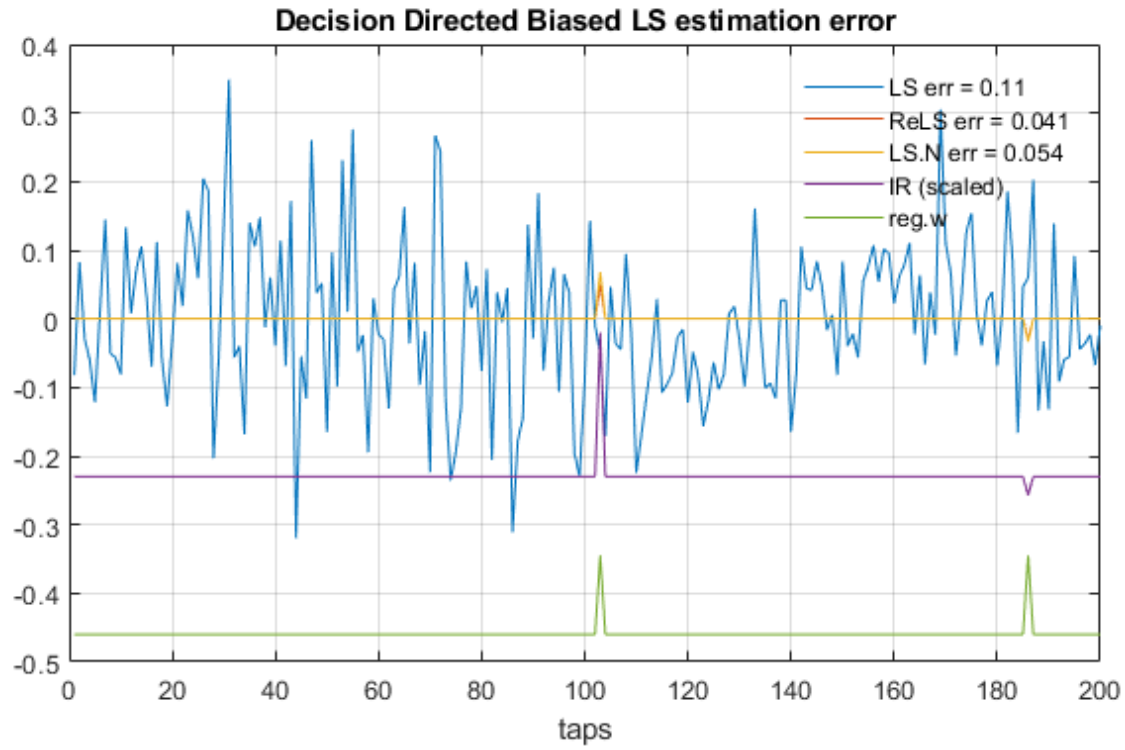
Additionally, if after the first run of LS / MLS/ chirp, we get a noisy IR estimate with N_r spikes – candidates, then using ReLS:

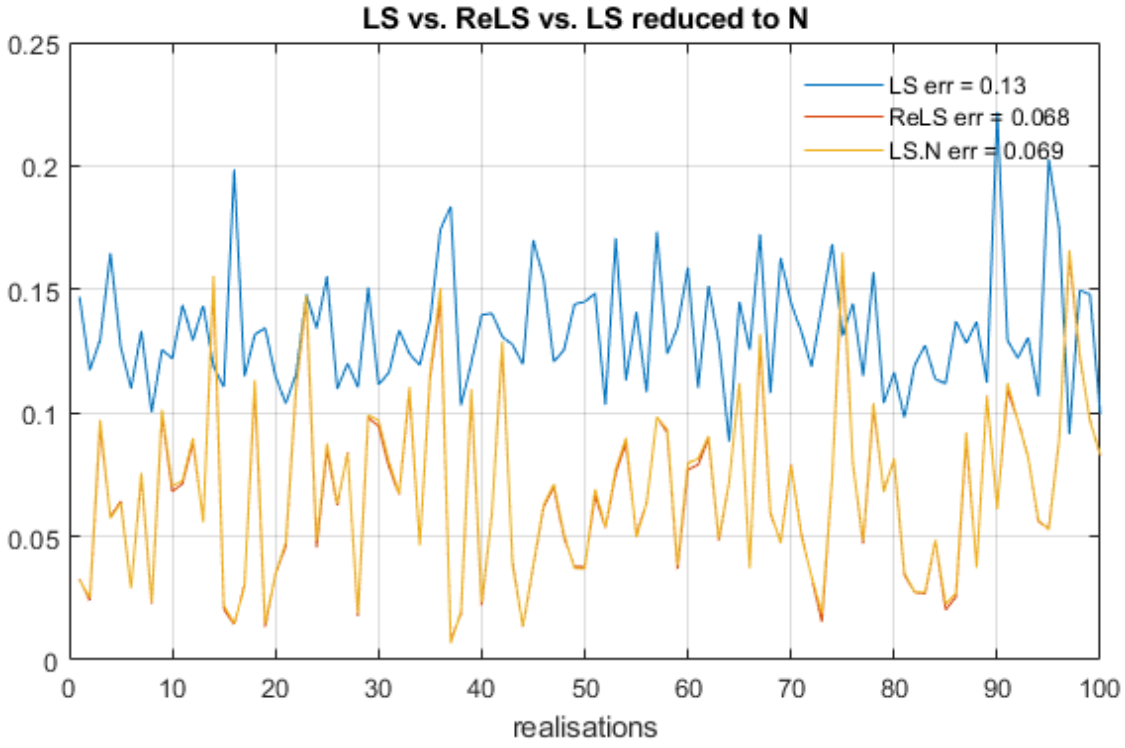
$$h = (X^H X + \sigma^2 D^{-1})^{-1} X^H y; \text{ where}$$

*D=δ*diag(zeros(idx₁-1,1); 1; zeros(idx₂-idx₁-1,1); 1; zeros(…,1));*

… is equivalent to inserting zeros between candidates which is equivalent to reducing the dimension of Fisher matrix from *LADF* to N_r. The model noise variance shall drop (no CR bound broken), if we have located (by iteratively narrowing down the candidate regions) the correct candidates.

FSAF can be implemented in cheaper low-power and compact radar processing SoCs which may help in the drones-against-drones' combat, etc. One who locates the enemy (while being below visibility threshold) first - wins.





I am not a radar expert. I could have reinvented a wheel or made a mistake; if not, I would like to assign all relevant Intellectual Property rights exclusively to the MoD of the USA[19].

[19] as my gratitude to the USA President Joe Biden saying "For G-d's sake, this man cannot remain in power!". If a nation loves dictators, they shall be free to live that way - but any podonok who have threatened WMD shall be removed [into a psychiatry ward on a tiny island in the middle of an ocean,] away from a red button.

5 ADAPTIVE FRAMEWORK GUIDELINES

5.1 BASICS

The task of developing adaptive filtering apps is particularly hard here because proper theoretical analysis of Fast Subband Adaptive Filtering applications in their entirety (performance, convergence, stability, numerical robustness, on all possible RIR and RIR variations, real-life excitation classes of inputs and outputs, double talk scenarios) is somewhat challenging.

This problem may be facilitated by augmenting FSAF with a collection of checkpoints for particular aspects of FSAF performance under various conditions. These checkpoints are set at every "location" where we can compare the actual, observed performance with theoretically expected, and collect appropriate statistics. Let's denote such collection of feedback sensors as a framework.

The framework is not a replacement of proper theoretical analysis because even superficial, non-exhaustive testing of adaptive systems is an extremely lengthy process. Once upon a time, it took me about 3 months to perform and document ITU-T G.167 & P.34x AEC compliance testing.

The framework's meaning is to provide feedback on how adequate is our theoretical understanding of stochastic adaptive processes, and to guide in improving it. In a certain sense, framework of checkpoints is an adaptive approach to the research of adaptive algorithms.

The biggest obstacle to Adaptive Applications' development is, by far, a habit to think within the Program Control paradigm.

5.2 PROGRAM CONTROL (PC)

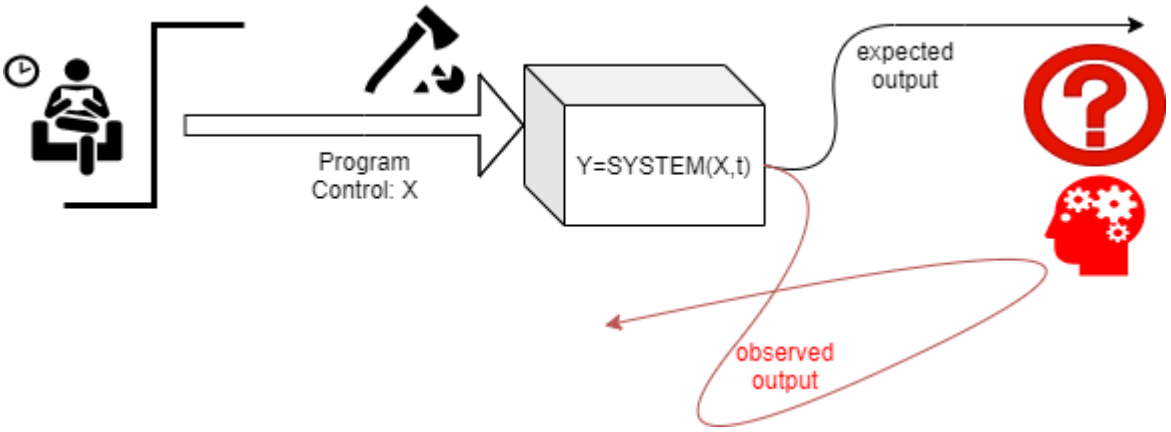
5.2.1 Principles

Program Control is the most common approach to manage in life, working best in well known simple systems in well known repetitive circumstances, when all mistakes have been already done and the lessons learned.

Program control is based on observations of what have worked in similar situations in the past, and what did not, and evolution-like selection of best practices. Any recipes, either culinary or numerical, belong to Program Control domain. Such practices are passed through generations, often become ritualized and sacred. Sometimes, nobody could recall when and why such and such rule was introduced.

Usually, a plan A is worked out in great details[20] prior to its execution, and then enforced on the system. At the end, the results are measured or demonstrated. Meanwhile, any observations are ignored[21]. If plan A falls short of a target, usually more (and more) efforts are applied. When people (nearly) run out of time, they panic and start a search for a magical silver bullet. Traditions of program control are unbelievably stubborn[22].

[20] Banks and investors continue to request entrepreneurs to submit 'hockey-stick' business plans for hi-tech start-ups while a "plan for conquering unknown" is an oxymoron, obviously.
[21] Semmelweis' discoveries and solution were completely ignored for 20+ years till Louis Paster, et al had proved the then-dominant theory of miasmas causing all illnesses to be a nonsense.
[22] In military academies of all countries, generals continue to teach methodically crafted plans based on the experience of prior wars. Time and again, such plans dissolve in the "fog of war" bringing catastrophic losses to both sides.



Our default behaviour is essentially Program Control, driven by our limbic system, without any "*Dubito, ergo cogito, ergo sum*[23]" prefrontal cortex-based supervision but with a belief that whatever we are doing is all right and the results are just behind the corner. When the results fail to materialize… time and again, we easily find someone else to blame. Then, to avoid future failures, we take courses on rising self esteem, discovering a g-d/dess inside, etc.

Program Control is based on many implicit assumptions, such as repeatability, predictability and controllability.

5.2.2 Deficiencies

Often, Program Control fails when the conditions (it was developed for) are no longer true. In a certain sense, Program Control becomes a victim of its own success when it is extrapolated far beyond the area of its successful applicability.

The systems are often under-modelled, and the limitations of Program Control applicability are forgotten and trespassed. When a system is complex, dynamic, stochastic and changing, or has latent feedback loops[24], etc, no plan survives contact with reality. The optimal Program Control is especially dangerous. It often drives inputs to the extremes, displaying either destructive overreaction or complete ignorance of stimuli. As Dr. A.A. Pervozvansky taught: "There is no system worse than optimal"

- ✓ *A few times at work, my managers were asking me to help other scientists, who worked for several years on issues but could not bring them to conclusion. Time and again, I've looked at pages covered with formulae, pages of MATLAB code with 120+ character lines, 10 operations per line, and asked the same question "When you wrote all that, did you think how you were going to debug and test it?" Time and again, the answers were that the idea seemed excellent, and the results should have been spectacular. Time and again, I suggested splitting the code into one operation per line, and to put a couple of plots after each one. Time and again, I learned that scientists did not know what to plot, nor what to look for at those intermediate plots.*

[23] Contrary to the predominant at the times, religious "I [perfectly] believe [in one and only true G-d] therefore I exist [in the [[original] [heavenly [Platonic cave]] true] world]".
[24] Like any novel strain of any virus have always done.

5.3 Overcoming PC Deficiencies with Feedback Control (FC)

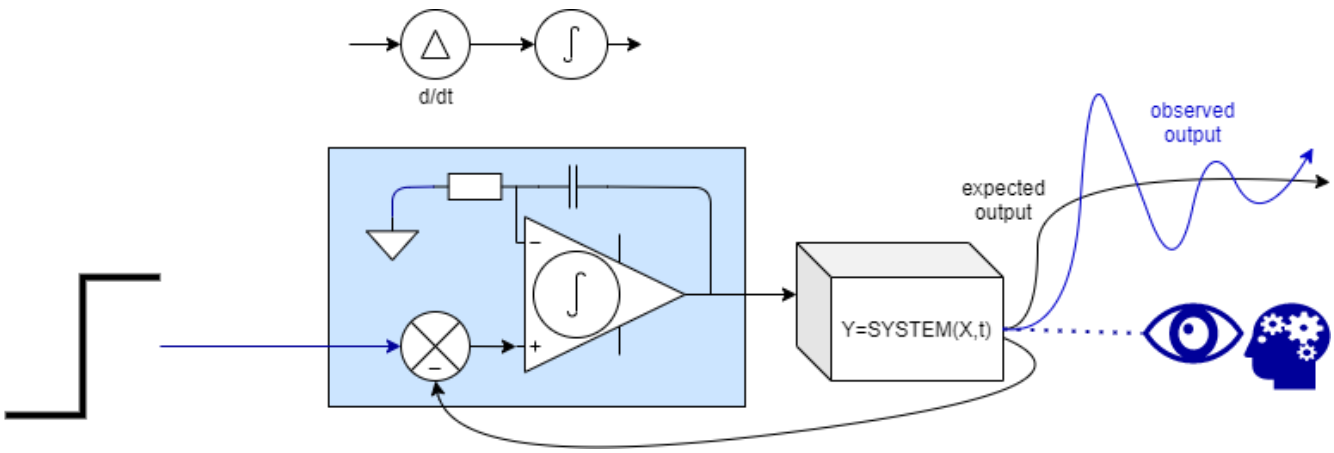
5.3.1 Principles

Instead of a big leap forward of Program Control, Feedback Control is a snail climbing Mount Fuji, but slowly, slowly![25]

Instead of following a undoubtfully genius strategy, Feedback Control anticipates that we will err: err badly, fail, completely screw up, many times in a row. The ideal system architecture is not one without mistakes; it's one where mistakes are easily locatable and fixable.

Instead of searching for a silver bullet (which is assumed non-existing), the problem that FC recognizes is: Where have we already erred? How to fix these inevitable mistakes? How to recover each time (not if) we fail?

FC is not an improved PC, it's principally different. While Program Control ignores measurements of the system's intermediate output, Feedback Control is decisively continuous measurement-centric. FC adds a differentiator and an integrator, in quite a peculiar way: the differentiator, instead of *out(t)=in(t)-in(t-1)* calculates *out(t)=in(t)-system_out(t-1)*.



Feedback control takes frequent measurements of the system output, calculates the differences between them and the desired values (so called error signal), then integrates this differential and adjusts the system's input ***X*** based on the result. Instead of elaborated plans, only a general strategy is worked out prior to execution. The rest is "play by ear".

The Feedback Control is much less dependent on the accuracy of mathematical models of the system: the need to know the entirety of (non-linear or non stationary) system's latent variables precisely is weakened. We can work with a somewhat "gray box" description.

- ✓ *While helping other scientists, the hardest was to convince them that their ideas were not without some hidden imperfections. That there were obstacles, difficulties, and contradictions. That we had to measure intermediate results at as many test points as viable, and as frequently as possible. That we had to have good observability of the algorithm's latent variables, and review each midpoint before progressing to the next.*

[25] By Kobayashi Issa

5.3.2 Deficiencies

The system must be unconditionally stable, and the speed of adjustments shall be an order of magnitude slower than the dynamic of the system. When the system is not stable, and/or it changes a lot during operation, feedback control would likely dysfunction.

If we make severe under-modelling assumptions, and/or don't measure the feedback signal properly, the closed loop may turn out positive. Then, a fluctuation would kick the system over the discretization threshold and cause ringing or out-of-phase exponential saturations.

There is another major deficiency: Feedback Control is quite contradictory to the way a brain used to operate. Most of otherwise perfectly normal engineers and researchers simply do not understand how Feedback Control works[26]. I mean, at all… even if they sincerely believe they do.

Solid understanding of what is involved in Feedback Control started to crystallize only in 1930s, primed by H.S. Black, H. Nyquist, and H.W. Bode. FC still remains a kind of esoteric discipline: some do it extremely well, why others keep on creating systems that that either don't work or blow up, time and again.

However, Program Control has its place in life. If a project is simple and predictable, has been done many times, and can be somewhat satisfactory managed with PC, it should not be overcomplicated by FC.

[26] Once in my experience, while skiing in the mountains, a friend of mine told me that an optical system [he worked on for a few years] had been frequently going out of control.

That optical system was an optical last mile, with a pair of transceivers A and B, each having a laser and receiving photodiode. The initial working point was set by a technician. The system was designed to remain in the photodiode's working zone in any conditions by adjusting the laser power, driving it lower when it's clear, and higher when it's raining or foggy. As the fog affects both transceivers, the indication of atmospheric losses was read locally, from A transceiver 's photodiode to affect the same A transceiver's laser's average power (same on B). However, a way too often transceivers were jumping out of phase: A was transmitting on the highest power, and B on the lowest (or vice versa).

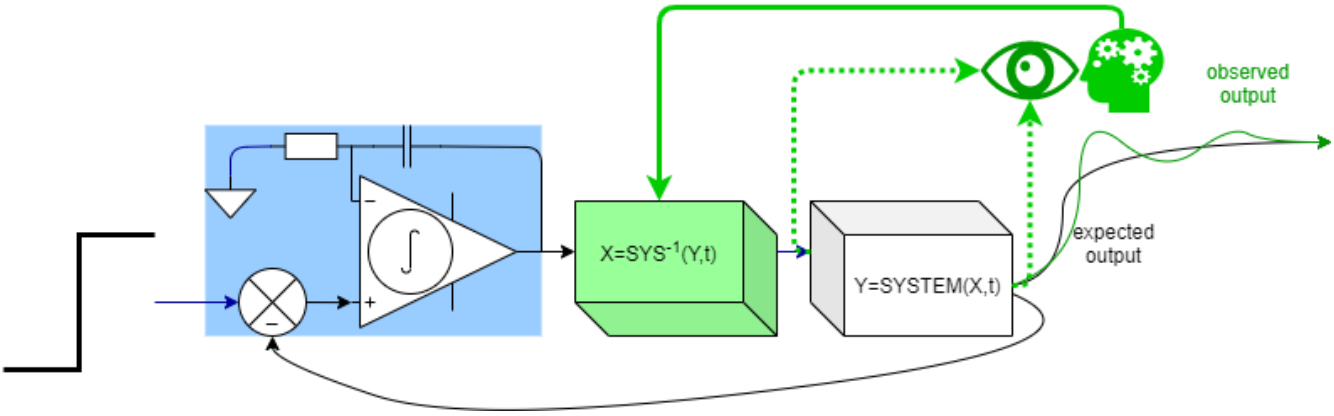
To me, it was immediately obvious that while the feedback was negative for common mode, it was positive for a differential mode. Of course, they had to use "A" photodiode to adjust "B" laser - but that was not possible within their architecture.

To them, it was not obvious at all: the company fought with the problem for at least a couple of years, with nearly 200 engineers and endless consultants and professors. All of these engineers and scientists learned control theory in Universities, everyone passed the exams, but no one could apply the theory to practice. Eventually, the company went belly up.

5.4 Overcoming FC Deficiencies with Adaptive Control (AD)

5.4.1 Principles

The Adaptive Control, introduced in 1950s, uses the measured signals not only inside the feedback loop but also to adjust the system's model to fit the real world as it reveals itself through observations.



The Adaptive Control (here model-indirect, MIAC) is even less dependent on the accuracy of mathematical models of the controlled system than the Feedback Control.

The controlled system may be non-linear and time-varying, it may undergo both rapid and slow changes, be affected by various unpredictable and non-measurable factors. There are usually several additional implicit feedback loops on measuring different parameters, such as noise level, to be used inside adaptation loop. These loops may operate in various domains, such as in energy (squared variables) or in log (dB) domains. All that makes the theoretical analysis of an entire stochastic AC system overly challenging or impractical.

The proper functioning of Adaptive Control systems is not assertable by magical architectures and universal algorithms but through tedious objective testing against an elaborated strategically designed network of checkpoints. Such network is a kind of testbed to ensure controllability, observability, and identifiability and other abilities of the adaptively feedbacked Feedback Control.

Generally, the rate of updating the model needs to be an order (or more) of magnitude slower than the feedback loop bandwidth product, unless a rapid system change has been detected. Under-modelling is a major sin but finding a right balance between adequate- and over-modelling is difficult, etc.

5.4.2 Deficiencies

Here, the main problem of AC is our brains. The counter-intuitive nature of AC overruns any technical issues. Adaptive Control is not what we, as human animals, do and how we live. Learning-while-acting, a quintessentially prefrontal cortex activity, is a sure no-go for us – it's "thinking slow", a way too slow for everyday life.

First, we spend years mimicking our parents in recognizing, classifying and reacting to events. We are required to always ask parents first, act later. Then we go to schools and are instructed to follow rules, to do that and not to do this, to obey the authorities unconditionally[27]. After all that, how can anyone grow up to become an active, independently thinking, creative human being?

[27] … pretty much exactly the way Frederick I of Prussia created the first European education system to ensure that all his subjects understood and obeyed their corporal's orders

It should not be a surprise that the word "adaptive" is almost always misused. You can read many academic articles and books, dedicated to Adaptive Control, and see that the authors thought in a Program Control way. They talk about the approaches, architectures, plans, rules, and strict procedures to follow. But can we build an adaptive control / filtering system using a non-adaptive design? Should we reframe our minds first, to fully understand what adaptive control is about? But … how?

- ✓ *While helping other scientists, I noticed that, after setting up a framework of intermediate control plots, we usually run into a showstopper within 1…5% of the total work. Then, I usually suggested to restart a research project with an idea-testing-framework, with Murphy-law thinking on potentially everything going wrong, on how to locate a problem, how to prove that it's been fixed, instead of dreaming of inventing an ideal 1-2-3 plan to conquer the world. My advice was never heard.*

5.5 Overcoming AC Deficiencies with Adaptive Framework

5.5.1 Principles

- The customer-visible quality of FSAF and/or other Adaptive Filtering applications is defined by how well and how thorough they are tested.
- It is the testing that takes lion's share of research and development resources. Therefore, in accordance with LS principles, the development efforts shall be redistributed, in a squared proportion.
- We shall consciously reject the Program Control's assumption that a predefined plan to solve a complex problem may exist, or that it's worth pursuing at all.
- We shall expect ourselves to make mistakes. Lots of them.
- We shell start with setting up a testing framework, with a collection of checkpoints, creating a network of feedback-on-error loops, and a library of test vectors. It's much easier to say than done. It may take years to setup – but it is reusable from project to project.
- Such framework is a cradle of algorithmic research and development, or a cauldron where ideas are cooked over, verified, fine tuned, mixed and entangled together, and sometimes even well tasting understanding is born.
- To identify a solution / model is only a half of work, to assess the solution / model's dispersion matrix (or another measure of convergence fidelity and robustness) is another.

5.5.2 Deficiencies

- This book is not a set of bullet proof recipes on how to build rock-solid[28] Adaptive Filtering algorithms.
- AC is pretty difficult, takes lots of time and requires stamina.
- It may disagree quite strongly with project deadlines, a need to publish annually, etc.
- It's much harder to publish, generally
- It's very hard (or plain impossible) to describe the AC project in the terminology of Program Control: to defines waterfall-like stages, to estimate the resources and time needed, the final performance, etc.
- AC projects shall be made to last because they are not maintainable because those, who are capable of maintaining someone else's AC system, are not interested in non-creative jobs.
- If a FC system works, don't ruin it by piling an AC on the top.

[28] If you practiced mountaineering you know what the angle of recluse is and that "rock-solid" is a joke.

5.6 SUMMARY

Working applications of Adaptive Control and/or Filtering are deeply rooted in the in understanding that we operate with extremely simplified models of reality such as band-limited FIR. Any simplified models are myths. When we forget about mythical nature of models, there will be troubles. It's important to acknowledge and accept imperfections of models everywhere, including acknowledging the gap between real and imagined knowledge, skills and competencies, and adjust accordingly to the errors, the difference between expectations and observations of actions and outcomes.

This gap has been laughed at many times. It's 95% of drivers believing themselves to be above average, the Dunning-Kruger Effect, etc. Unfortunately, it's not something only anecdotical silly people do. There are very many gaps in our knowledge, seeded by imperfect educational systems. We may not notice them because we all share them… but they do exist[29].

"While I have endeavored to describe things as they appeared to be, I am conscious of having been unable to avoid the usual proportion of errors, for which I beg indulgence, and which I leave for others who shall pursue the same path of investigations to correct[30]

[29] Once upon a time, I worked in a company working on Internet-over-ATM routers. Obviously, they needed to calculate how many ATM cells they need for an Ethernet packet: (1500+47)/48. The processor they used (MIPS) did not have an op-code for division (obviously). So, the lead developers, a group of engineers and PhDs from all over the world, including those who had worked in CERN, started to meet and discuss the "division by 3" problem. Once I commented that 3 is a constant. Therefore, they did not need to divide, but multiply by 1/3 with rounding. The answer was "Don't you see our wisest people talking? Shut up and get lost". I departed the company soon but kept relations with a girl who worked there.
There years later she told me of bad troubles with VxWorks on their workhorse processor(s). She explained that they did not need the VxWorks but had to have it because the library for division was not separatable. As you can guess, that was solely for the abovementioned division by 3.
I laughed, sat down and in 15 minutes wrote a simple code to find a multiplicator, a rounding constant and a shift to make sure that for all *0 < x <= 1500: (x+47)/48 = (x*mult+rnd)>>shft*; It turned out later that MIPS compiler used left shifts and adds instead of multi-cycle multiplication op-code. The bandwidth of their workhorse processor jumped 5 or 6 times, afaik; the girl was promoted to team leaders; etc.
Why a group of not-untalented engineers and PhDs could not manage with such a basic math? No one from them was capable of admitting gaps in their math education, that s/he did not understand the relationship between * and /, nor the algorithm of division they learned in pre-school. That algorithm was given to us without much explanations because we were too young to understand them (actually, multiplication and division with Roman numerals were taught only in universities at their times, not in schools). It is not the only gap, unfortunately. Knowing and understanding math are two very different disciplines.

[30] Joseph Leidy, Fresh-water Rhizopods of North America, 1874.