legends: Orange (Medium Priority), Red (Critical/High Priority), Blue (Info), Green (Good work!)

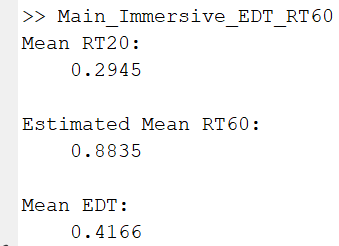
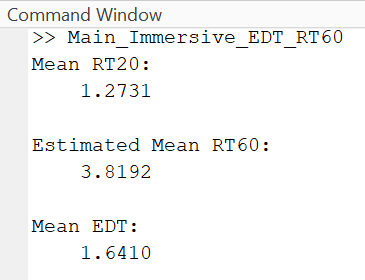
Done, Important

# Monday, 29 July 2024

## **FOCUSED TODO**

1. Use RT30 for RT60 estimation via extrapolation, and generate all graph/value to compare with Kim21, especially for RT60 on each bands as seen on picture above.
2. Try maximise/increase global steam audio parameters to increase accuracy (hopefully?) to take advantage of using baked instead of realtime.
3. Figure out what causing early reflections (high EDT) in ST scene.

Continue with trying to get RT30 working properly with provided matlab code because for some reason simply changing y\_fit to fit only 30 instead of 60 doesn’t work as expected.

1 good news (not really) and bad news. Good news is that the codebase itself it most likely working as expected (y\_fit), atleast it works the same as it is for RT60, the problem is that even on RT30, the best fit line freaks out when its not a good energy decay curve ie one without weird flat noise level. This is proven as using RT20 instead to get estimated RT60 gives very similar value to expected RT60 for KT. The bad news is that this doesn’t work at all for ST… Which means we need to go back to drawing board and find out what is difference between Mona vs mine and see where this weird noise floor level coming from. Ideally there should be no changes to codebase, or as little as possible and still use RT60 instead of extrapolation.

So for now, lets get back to drawing board and mess with Unity/Steam Audio to get rid of the noise level thing.

## **UPDATED FOCUSED TODO**

1. Troubleshoot the RT60 problem in Unity (reduce noise floor level)
2. ~~Use RT30 for RT60 estimation via extrapolation, and~~ back to normal RT60 calc. generate all graph/value to compare with Kim21, especially for RT60 on each bands as seen on picture above.
3. Try maximise/increase global steam audio parameters to increase accuracy (hopefully?) to take advantage of using baked instead of realtime.
4. Figure out what causing early reflections (high EDT) in ST scene.

Hypothesis 1: Due to the fact that my scene mesh are not voxel, maybe that’s why noise level higher?  
Let’s try it on S3A and see how it performs, next lets use 0.5 volume setting on ST.

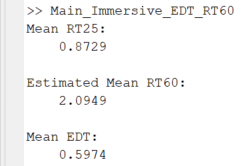
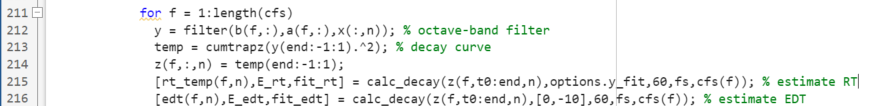
Maybe combination of 0.5 volume + ST30 extrapolation can work? If doesn’t, lets also add on 0.5 direct mix level instead of 1.

Ok both did not improve it by much.. Maybe lets turn off HRTF.. Nope, not doing much either.. Lets try 0.25 volume. Still too high.. ok EDT still soo bad, this needs proper experimentation and testing and troubleshooting instead of shooting into the dark. Lets list out all related component and settings etc. idk my brain fried for today. Lets figure it out tomorrow man.

# Tuesday, 30 July 2024

WFH, went back to drawing board. Studied the matlab code and research about related topic to further improve understanding. Brainstormed some ideas to troubleshoot problems.

# Wednesday, 31 July 2024

After reading the code line by line to figure out whats wrong, found the culprit! It was 2 lines in irStats that presumably forgot to account for y\_fit value changes, they hardcoded 60 instead of using y\_fit value for EDT and RT60 best line fit and calculation… That’s why the graph best fit line don’t really change even after changing y\_fit. THIS IS BIG FIX! Finally fix the bug halting progress for more than 4 days.

On line 215,216, the value (60) should not be hardcoded but dynamically change depending on value of y\_fit. After testing values on ST using 0.25 volume sound, it seems RT30 is too high and RT20 is too low, thus RT25 is a good middle ground to settle at getting 🡨 following result.

TODO tomorrow is to change the value before fs to dynamically change depending on y\_fit argument given, so the calculation works as expected. After that lets reupdate focused TODO. Done for today.

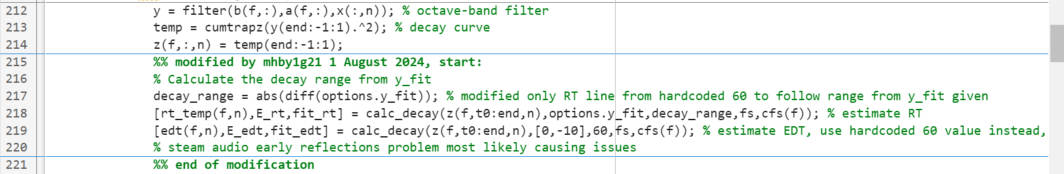
# Thursday, 01 August 2024

## **UPDATED FOCUSED TODO v2**

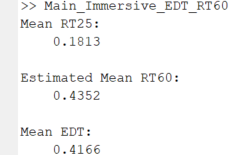
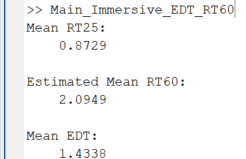
1. Troubleshoot the RT60 problem in Unity (reduce noise floor level) done-ish, presumably the reason is that our scenes are meshes while Kim21 and Mona’s are voxel which means the noise floor level lower as they dissipates faster (? Only theory for now, need verification
2. Modify matlab code (irStats.m) to have the value range change dynamically depending on y\_fit parameter given done!
3. ~~Use RT30 for RT60 estimation via extrapolation, and~~ ~~back to normal RT60 calc~~. generate all graph/value using RT25 extrapolated to RT60 to compare with Kim21, especially for RT60 on each bands as seen on picture above.
4. Try maximise/increase global steam audio parameters to increase accuracy (hopefully?) to take advantage of using baked instead of realtime.
5. Figure out what causing early reflections (high EDT) in ST scene.

Fixed the hardcoded lines by adding dynamic decay range variable instead. Reminder to mind the variables name of RT25/30/60 though in Main matlab script. Tested with KT again and can confirm it works as expected! Nice! Next, let’s regenerate all audio for all scenes on a standardised settings (let’s follow ST that works).

Ok nvm, just realised the EDT for KT last time using RT25 extrapolated to RT60 is absolutely wrong (too low). This means, I just need to use RT25 on case by case basis (depending on how bad the noise floor level is)? Yea, also EDT for KT works best when taken over 60dB range, but not for RT60 as even KT have noise floor around -50dB which mess up RT60 calculation. So, right now best way is to disregard ST EDT value, let it be ver wrong (most likely limitation with recording/steam audio early reflection simulation, thus need further research into this, idea: watch steam audio gdc video etc.)



For reference, current KT and ST calculation:

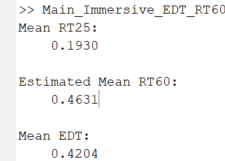
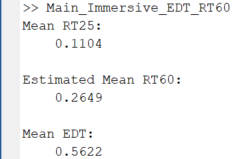
 Anyway, lets standardised the steam audio source parameters first.

Let’s name the audio file for it as best\_v1

## best\_v1 audio source parameter settings:

For baked audio source influence radius, just pick any number that encircle whole scenes. Windows volume mixer level is at 50%.

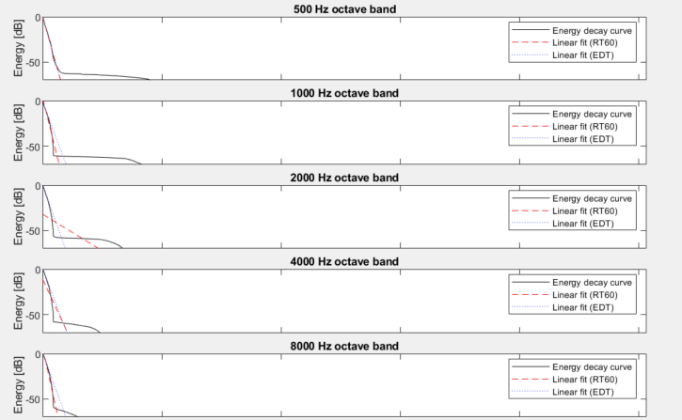
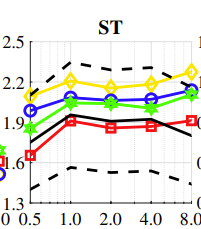
### Results

🡨 KT LR 🡪

As seen on the right, something is obviously wrong still, so lets use dyanmic value range for EDT as well again…

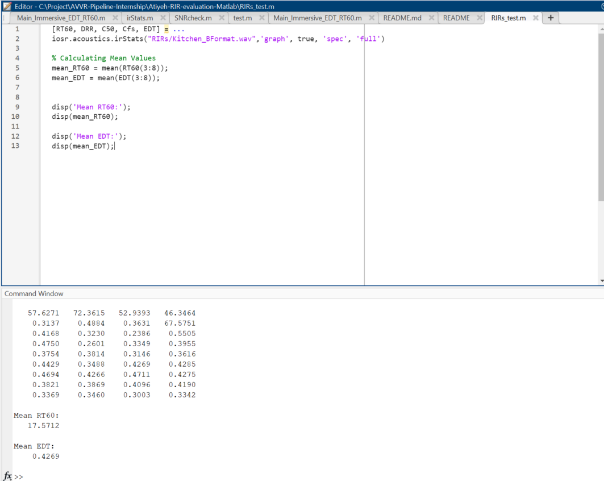
And even then, RT60 is too high (0.26 instead of around 0.18 recorded in real life), so let’s also make best\_v2 parameter, this time using way lower volume (0.10).

NVM, this don’t work either, standardizing one setings and RT calculation for all scenes wont work as the noise floor level is different for all. This gonna feels not so scientific and feels like fitting the datapoint to fit our needs but there’s nothing much I can do unless I want to rewrite the whole RT60/EDT best line fit calculation, instead of defining the range, just find where noise floor level starts earliest in all bands and use that to get ratio for extrapolating RT60, but even then, it still feels a bit weird ngl. So for now lets just create a setting and RT?? Ratio for all scenes….

One thing I can do is analyse the Kim21 audio sources (the gunshot) to see how the graph in that looks (especiall to confirm existence of the weird noise floor). NVM, they don’t have the RIR/sweep sound listed. I maybe should ask Dr Hansung Kim for that.   
Other than that, I need to ask Mona about clarification of the decay curve and how RT60/EDT is actually calculated because rn, it seems to me Mona results are merely kinda coincidental because on some bands, its totally going higher than others, atleast when compared to Kim21 band decay curve graph. Below: Mona vs Kim21 band graph for comparison in ST.

Plan next:

1. Generate manually tuned (both audio setting and RT20/25/30 extrapolation, ie different for each scenes to get as accurate as possible, kinda cheating?) analysis graph for each scene.
2. Email Mona and Dr Hansung Kim about the concerns above, and ask Dr Hansung Kim if there’s raw audio from previous RIR data for reference to double check if the problem exist in that etc.
3. Create/modify existing RT60/EDT best line fit to not be according to y\_fit, instead find the earliest point of noise floor and use that as cutoff point or something like that.

Assuming the .wav files in RIRs folder are the ground truth, even those give out wrong RT60 on base parameter analysis. (Yes’ I’ve reverted the decay\_range value to 60 again.)

OK simulated files are probably in Unity-ogg-files so no need to bother Dr Hansung as it’s aready provided, I just forgot to copy before because I thought it was some random stuff.

OK next plan is to investigate the other matlab scripts more (shouldn’t have assumed the Mona’s one is the best one… Yea it might be the best with her setup and sine sweep she generated, but better to confirm the older working one first… Rookie mistake by me ngl.)

Especially investigate plotting\_rir\_compare and difference between all the RIRs audio files naming etc.

Plan next frfr:

1. Investigate the other matlab script and get them to run, verify older audio files and scripts methods etc.
2. Replicate those with our scenes.
3. Email if needed and follow through below if problem still persists.
4. Generate manually tuned (both audio setting and RT20/25/30 extrapolation, ie different for each scenes to get as accurate as possible, kinda cheating?) analysis graph for each scene.
5. Create/modify existing RT60/EDT best line fit to not be according to y\_fit, instead find the earliest point of noise floor and use that as cutoff point or something like that.