

Introduction

Brick was designed with quality control in mind. One of the most important aspects of a sample rate converter would be the test suite used to verify that it is working correctly. Since Brick uses filters with noise floors up to a billion times quieter than human hearing, it is not possible to rely on listening tests alone to verify the correctness of the software. Therefore, Brick comes with a test suite enabling you to verify that it is working correctly. This is important because if you were in a situation in which your audio did have a problem, you would want to rule out each part of your processing chain and you would want to do it as quickly as possible.

It is very unlikely that Brick will give you incorrect output. If there was a bug causing problems in your output, then it would most likely have a cascading failure leading to very obvious output problems. In other words, if Brick does fail, it should fail hard. This has been my experience in developing the program. Unless everything is working correctly, the output will not be even close to valid.

However, this practical advice is not enough to verify the quality of Brick's output.

Brick's test suite falls into three parts: extremely precise input files generated using Mathematica, spectrogram analysis of sample rate conversion output (via Brick), and frequency response plots of the filters used (via Mathematica). You do not need Mathematica in order to prove to yourself that Brick is working. You can download the input files here and use the spectrogram test. (The spectrogram feature of Brick is extremely accurate and sensitive.) Using the Mathematica scripts will simply allow you to look under the hood. In any case, I have published here the results of the input files for all to see.

Since the spectrogram used is a part of Brick, there is also a need to verify that the spectrogram itself is working. Thus there are input files specifically for the testing of the spectrogram, for example: ensuring that dB levels are reported as accurately as possible.

Terms

Active Region: this plot shows

Transition Region: shows the transition from passband to stopband so that you may verify the steepness and smoothness of the transition.

Passband Ripple: expressed as deviation from unity in Decibels. This should in theory be as low as the stopband attenuation.

Phase Deviation: indicates the deviation in phase from the group delay. Sample rate conversion introduces a delay in the signal proportional to the length of the filter used. A linear phase filter will have no deviation from the group in the passband. This graph shows deviation in millionths of a radian.

Test Suite

- Spectrograms for dB test, near Nyquist tests
- Filter FFTs for: 200 dB, 0.0

- Difference between Mathematica precise, C++ sin function, C++ Chebyshev recursion, wavetable
- Varying dithers, levels of dither

FAQ

Why does DC/Nyquist always measure at approximately 6dB louder than the actual signal?

This is due to the effect of there being positive and negative frequencies for all but the highest frequencies.

Can the spectrogram use non-power of two sizes?

Yes. FFTW is doing the FFT calculations for the spectrogram, and it is capable of using

Why are my graphs not exactly the same?

Depending on the architecture the trigonometric functions used may evaluate to different things.

I need a noise floor in my calculations lower than -320dB—what do I do? For example, I need a reference sample rate converter to show the best possible result in a float64.

The best option for you would be to compile using the MPFR library.