A fast approximation for decibel to amplitude conversion on embedded systems.

Introduction and motivation

Many audio applications, such as dynamic range compression, require the conversion of a signal measured in decibels (dB) to amplitude (Y). This transformation is straightforward to implement on many systems.

$$Y = 10^{\frac{dB}{20}}$$

However, on an embedded system this may be too slow and the memory requirements for a lookup table may be prohibitive. Therefore, a fast and accurate approximation is required.

Combining lookup tables with interpolation.

To reduce the lookup table memory requirements, we precompute a subset of the values in the desired range and then perform interpolation to obtain the "in between values". We can separate the integral component(I) of this transform from the fractional component (F) of this transform.

$$Y = 10^{\frac{dB}{20}} = 10^{\frac{I+F}{20}} = \left(10^{\frac{I}{20}}\right)\left(10^{\frac{F}{20}}\right)$$

We use a lookup table for estimating $10^{\frac{I}{20}}$. This requires less than 400 bytes for 16-bit integers. For the fractional part we use a cubic interpolation based on Newtons divided differences

$$10^{\frac{x}{20}} \approx f[x_0] + f[x_0, x_1](x) + f[x_0, x_1, x_2](x)(x-1) + f[x_0, x_1, x_2, x_3](x)(x-1)(x-2)$$

Where $f[x_0 \dots x_n]$ is the notation for newton's divided differences. The divided differences can be defined recursively as

$$f[x_0 \dots x_n] = \frac{f[x_1 \dots x_n] - f[x_0 \dots x_{n-1}]}{x_n - x_0}, \qquad f[x_0] = f(x_0)$$

For the case of $10^{\frac{x}{20}}$ and 0 < x < 1 We note that

$$f[x_0 \dots x_n] = \frac{\left(10^{\frac{1}{20}} - 1\right)^n}{n!}$$

Making the substitutions

$$a = f[x_0], b = f[x_0, x_1], c = f[x_0, x_1, x_2], d = f[x_0, x_1, x_2, x_3]$$

We get

$$10^{\frac{x}{20}} \approx a + bx + cx(x-1) + dx(x-1)(x-2)$$
$$= dx^3 + (c - 3d)x^2 + (b - c + 2d)x + a$$

When we calculate the coefficients of this polynomial we get

$$0.0003027786x^3 + 0.006535915x^2 + 0.115179760x + 1$$

The final approximation is

$$Y = 10^{\frac{I+F}{20}} = \left(10^{\frac{I}{20}}\right)(F(F(0.0003027786F + 0.006535915) + 0.115179760) + 1)$$

The accuracy is better than 1 in 120,000 and the approximation runs in 0.23 microseconds using only 400 bytes of memory. Job done!