Dither and Noise Shaping in Digital Audio: Hows and Whys

Duane K. Wise
Wholegrain Digital Systems LLC
dwise@wholegrain-ds.com
www.wholegrain-ds.com

AUD200: Signal Processing
SAE Institute, Nashville TN
18 August 2021
https://www.wholegrain-ds.com/DigAud_Dither.pdf

Part One: Introduction

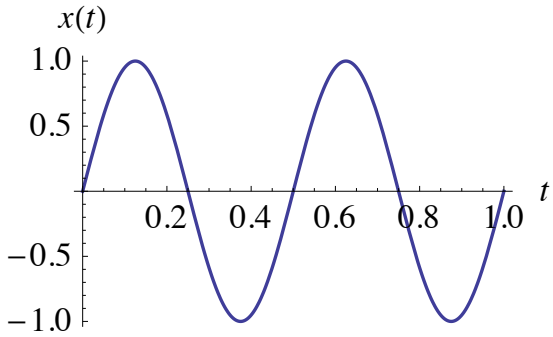
Background and Motivation

- Digital signals versus analog
- Numerical representation of a digital signal
- Quantization of a digital signal
- Side effects of quantization
- Manipulation of quantization operation to improve its output

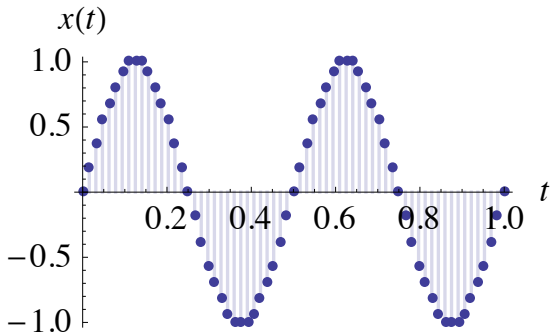
Digital Audio

The Important Differences from Analog

- All time points in the analog duration have signal values
- An analog signal evolves "smoothly" over time



- A digital signal is sampled in that signal values exist only at discrete time points separated by a constant period
- The signal values are coded into a finite-length numerical representation

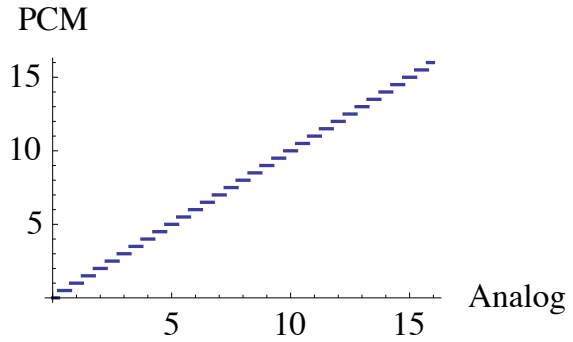


- This presentation focuses on discrete amplitude values, but sampling in time will be addressed when it can be exploited

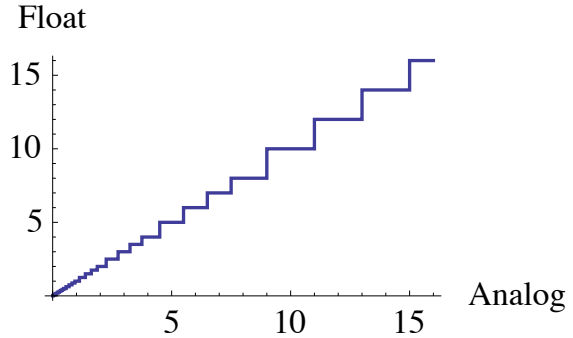
Numerical Representation

Quantizing Amplitude Values

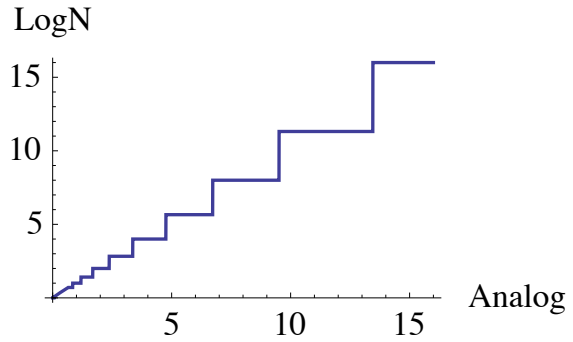
- Linear PCM is a scaled integer, mapped as a step function



- Floating point has a variable scale that is non-decreasing as the value increases



- Mu-law and other esoteric codes are non-linear and statistically motivated

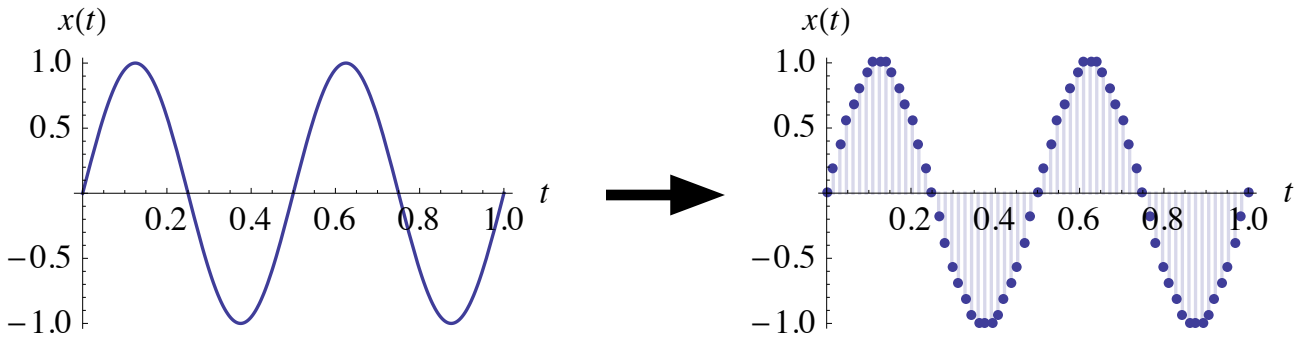


Quantization

Definition and Examples

- Quantization is the conversion of a value into a different precision format
- Implicit in this definition is the loss of accuracy--the best we can hope for is to break even

- Quantization examples: analog-to-digital

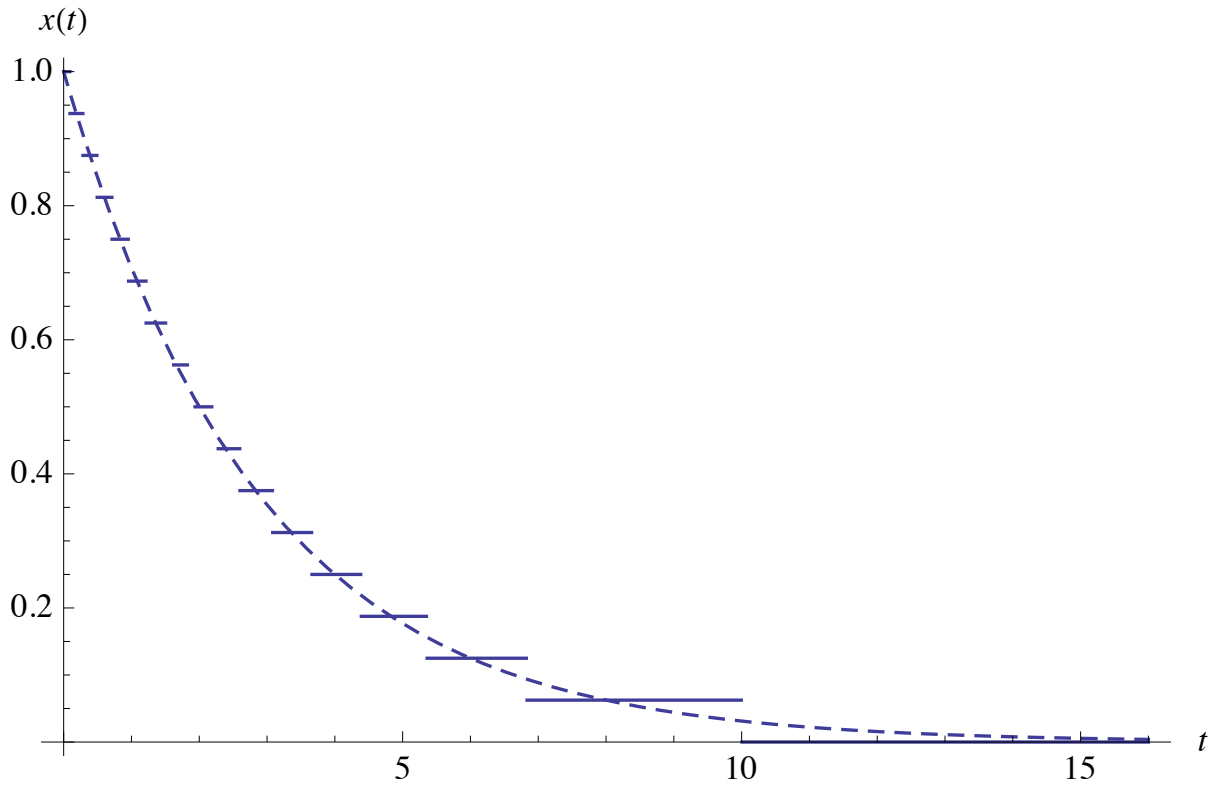


- Within a digital processor: double-to-float, long-to-short, etc.

Illustration of Quantization

PCM Quantization of Amplitude

- The dashed line is a continuous function, and the solid line is its PCM quantization

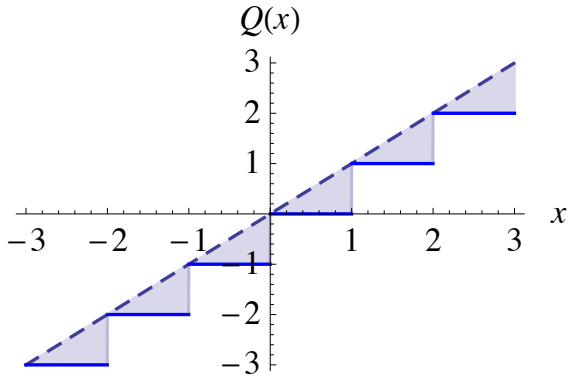


Basic Quantization Methods

Expressed As C Code From Float to Int

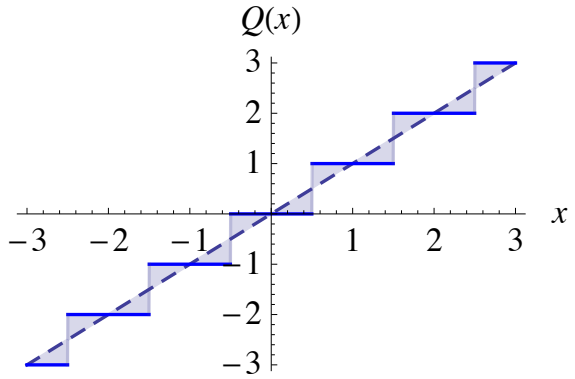
- Truncate $\lfloor x \rfloor$

```
intVal = floor(floatVal)
```



- Round $\left\lfloor x + \frac{1}{2} \right\rfloor$

```
intVal = floor(floatVal + 0.5)
```

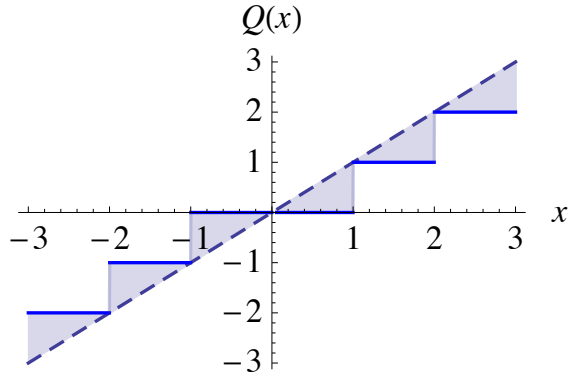


- Truncate-toward-zero $\operatorname{sgn}(x)\lfloor |x| \rfloor$

```
intVal = sign(floatVal)*floor(fabs(floatVal))
```

OR

```
intVal = floatVal
```



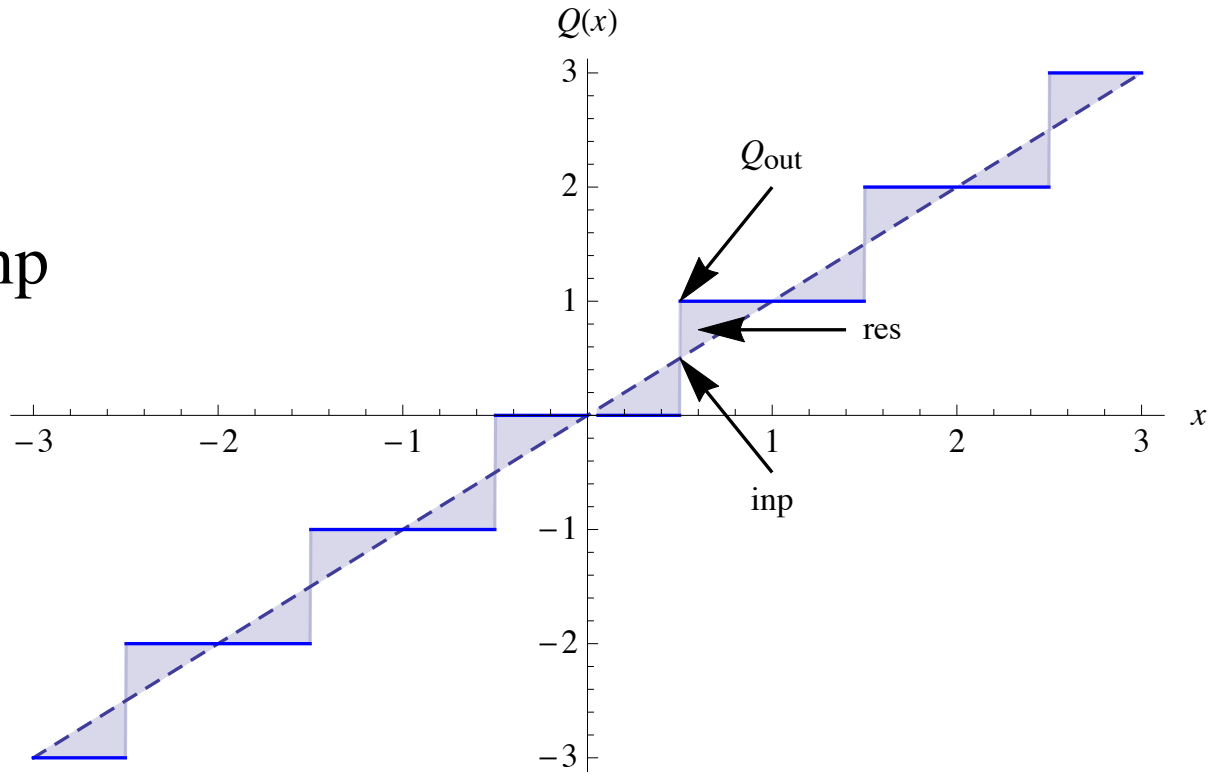
- Truncation will be used in all examples herein, with stuff added in

Residual of Quantization

Representation of Quantization Error

- The difference between the quantizer output and input is the error or residual

$$Q_{\text{out}} = Q(\text{inp})$$
$$\text{res} = Q_{\text{out}} - \text{inp}$$



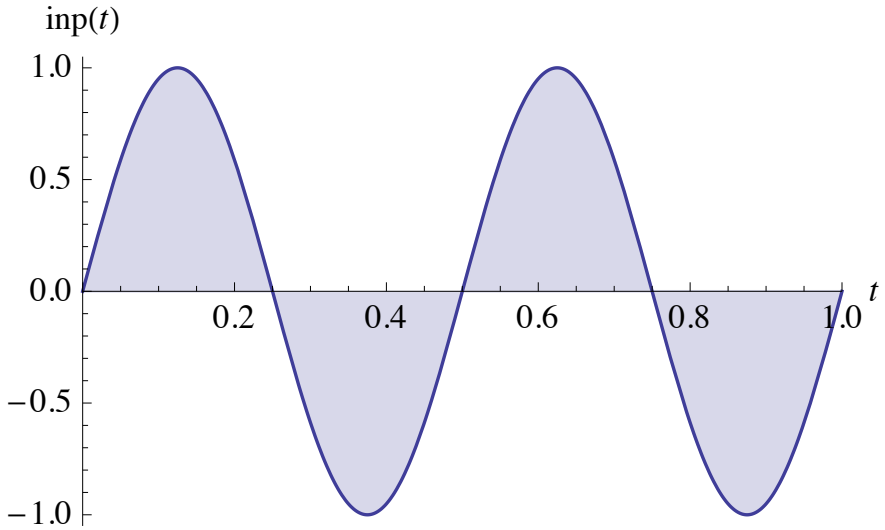
- The residual plus the unquantized value (quantizer input) is the quantizer output

$$Q_{\text{out}} = \text{res} + \text{inp}$$

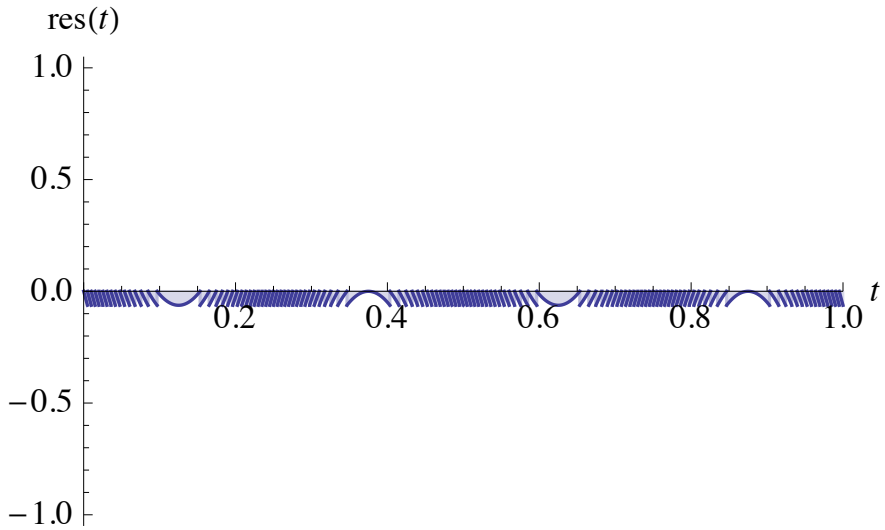
Residual Signal

As An Entity

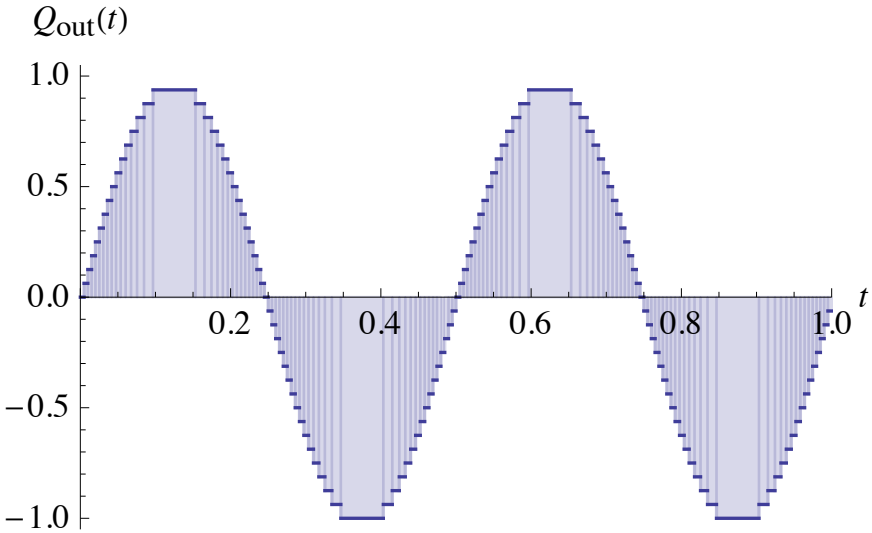
- Quantizing over the duration of a signal yields a residual signal
- This signal can be worthy of analysis on its own



+



=



$$Q_{\text{out}} = \text{res} + \text{inp}$$

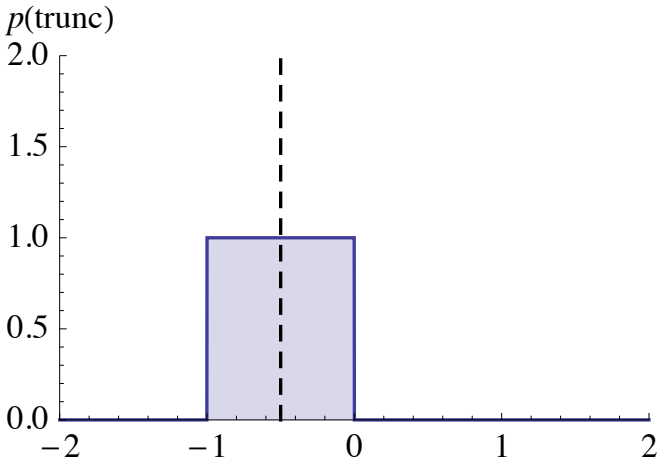
Residual Analysis of Basic Quantizers

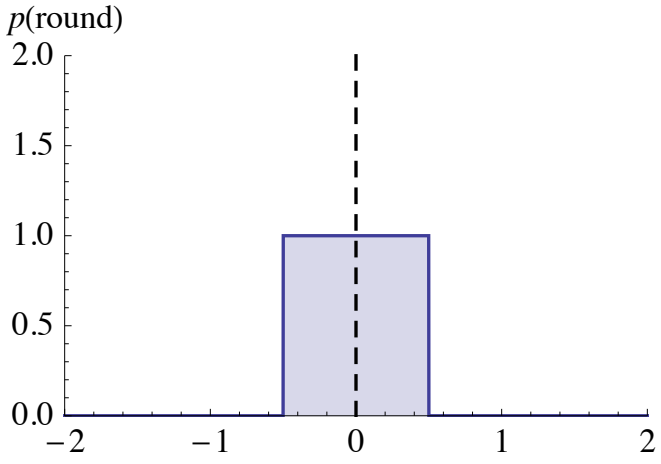
Based on Probability Density

- Range of rounding and truncation residuals

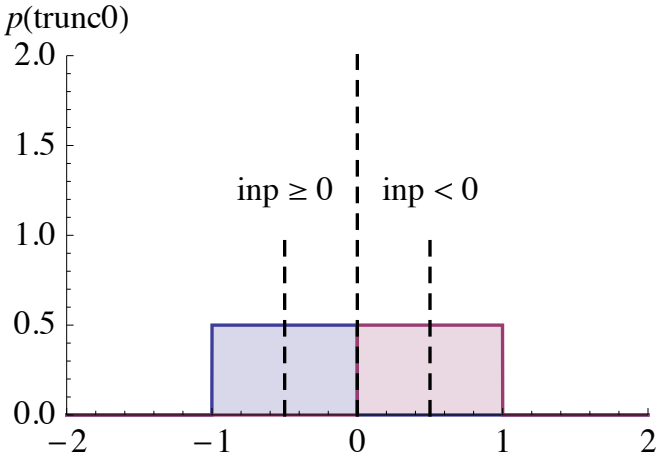
$$\text{range(trunc)} = (-1, 0] \qquad \text{range(round)} = (-\frac{1}{2}, \frac{1}{2}]$$

- Probability distribution of error
- DC offset (dashed line, center of gravity)





- Truncation towards zero

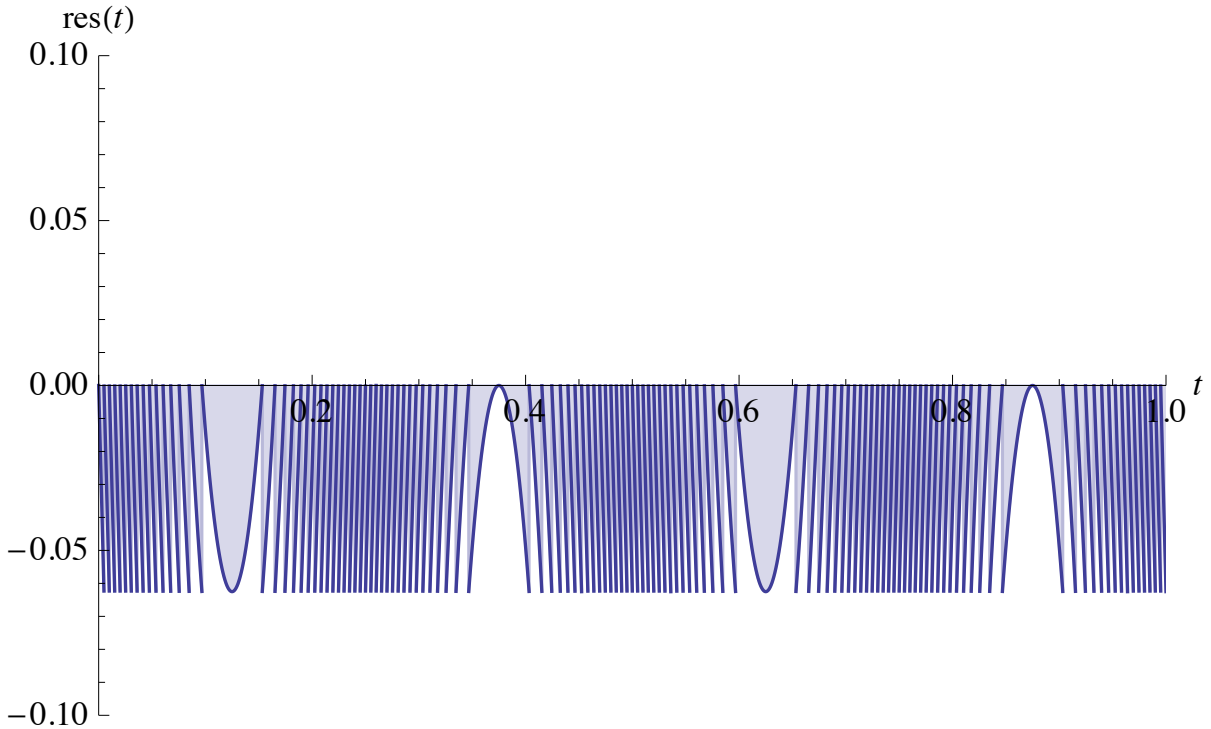


Statistics of Residual and Input

As Random Variables

- The residual signal can be considered a random variable separate to the input signal

- It is not safe to assume that the residual is statistically independent from its input--correlation is likely if input is not pure noise



Dither and Noise Shaping

Statistical Helper for Quantization

- Add a noise signal to the pre-truncation input signal

$$Q_{\text{out}} = Q(\text{inp} + \text{noise})$$

- Optionally filter the residual signal and feed it back to the quantizer input

$$\text{inp} \mathrel{+}= Q_{\text{out}} \otimes H_Q$$

- While an extraneous signal is being injected, the goal is to improve the quality of the quantizer output--remember that breaking even with the input is the ideal

Part Two: Dither

Implementation and Analysis

- Expected value and simple examples
- Digital noise signal generation
- Adding noise to input signal pre-quantization
- Context to which dither is applied

Statistical View

Expected Value

- Formula using probability distribution

$$\mathrm{E}(X) = \int_{-\infty}^{\infty} x\, p(X = x)\, dx$$

- Example case: analog input of constant non-integer value
- Case one: truncation quantizer with no added noise

$$Q(3.4) = 3$$

$$\mathrm{E}(Q(3.4)) = 3$$

- Case two: truncation quantizer with added noise

$$Q(3.4 + [0, 1)) = Q([3.4, 4.4))$$

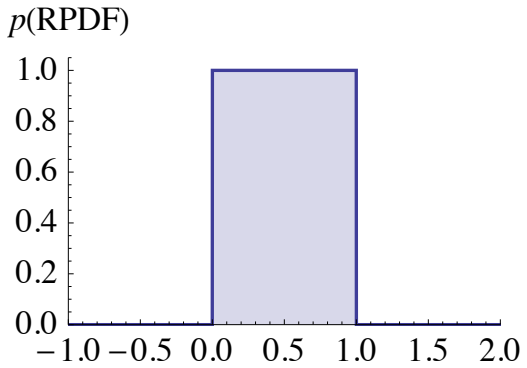
$$Q([3.4, 4.4)) = Q([3.4, 4.0)) + Q([4.0, 4.4))$$

$$\mathrm{E}(Q(3.4 + [0, 1))) = \mathrm{E}(Q([3.4, 4.0))) + \mathrm{E}(Q([4.0, 4.4))) = 3 \times 0.6 + 4 \times 0.4 = 3.4$$

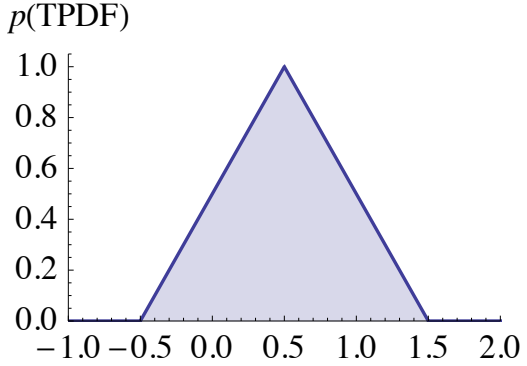
Noise Generation

By Probability Density Function

- Rectangular Probability Density Function (RPDF)



- Most, if not all, software random number generators yield an RPDF
- Adding random variables, i.e. noise signals, convolve their PDFs
- Triangular Probability Density Function (TPDF) is the sum of two independent RPDFs



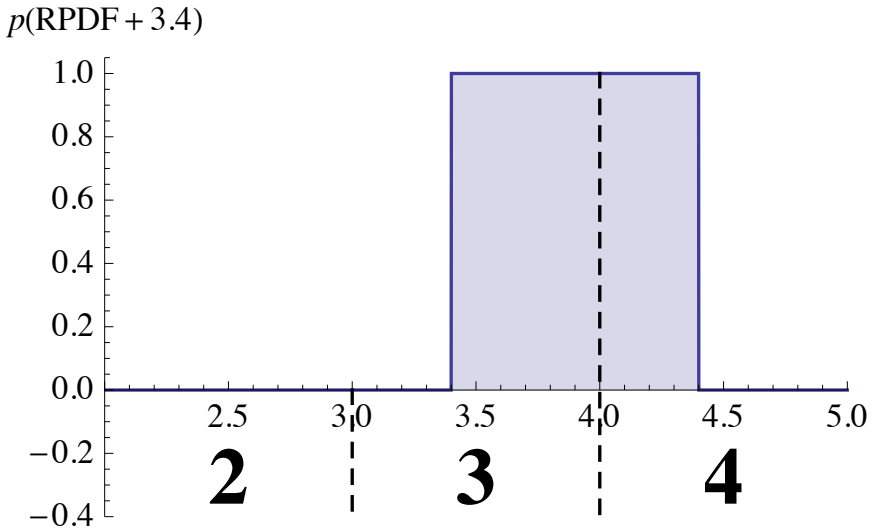
- A nonlinear function or allpass filter can be applied to adjust the noise PDF post-generation

Expected Value of Dither Application

Using a Truncation Quantizer

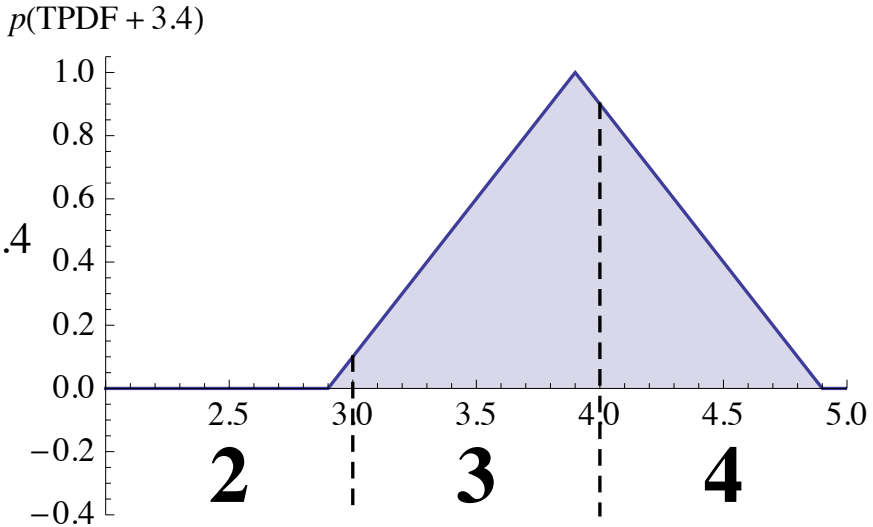
- Addition of RPDF noise of range zero to one to analog constant non-integer value

$$\mathrm{E}(Q(3.4 + \mathrm{RPDF})) = 3 \times 0.6 + 4 \times 0.4 = 3.4$$



- This is also true for TPDF noise provided that the crown is centered at +1/2

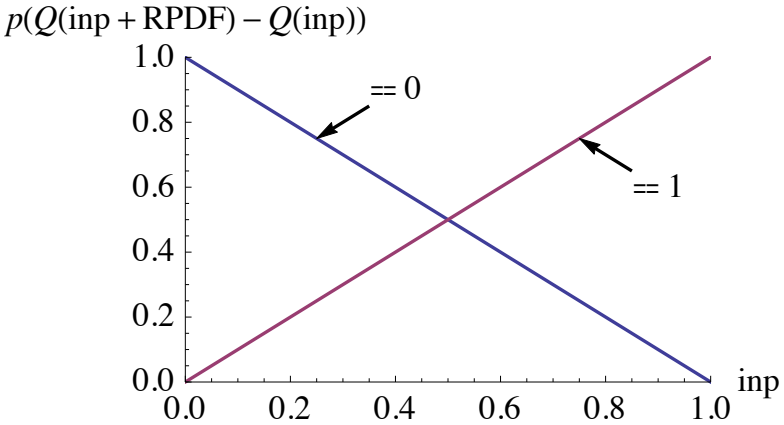
$$\mathrm{E}(Q(3.4 + \mathrm{TPDF})) = 2 \times 0.005 + 3 \times 0.59 + 4 \times 0.405 = 3.4$$



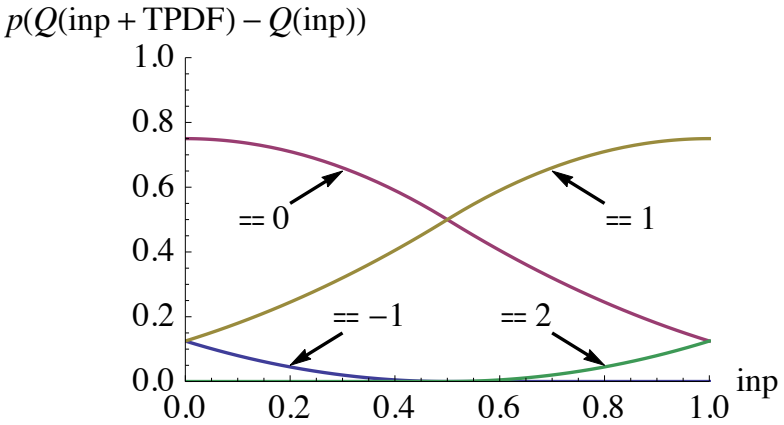
Treatment of Correlated Residuals

When Expected Value is Not Enough

- While RPDF does efficiently interpolate between quantization points, it is often not enough to fully modulate the residual completely out of correlation with the input



- TPDF provides nonlinearity and extra range to further kick the resudual out of correlation with minimum noise energy growth



Application of Expected Value

In a Recursive System

- A recursive digital signal processor such as an IIR filter will require quantization of its delay state due to multiplication bit growth

- The expected value of a quantizer within the recursive loop will affect the accuracy of the transfer function

- Paying attention to the expected value of a dithered quantizer can help avoid limit cycles, which are unintended correlated residuals on the order of the PCM increment

Part Three: Noise Shaping

Implementation and Analysis

- Spectral masking
- Equal loudness curve
- Applying masking and equal loudness to quantizer output
- Designing and employing a noise shaping IIR

Masking Spectral Buzz

Defeating Unwanted Tonal Components

- The residual signal from a quantizer is considered a separate random variable, but it may be correlated with the input signal

- The energy of this signal is bounded by the maximum magnitude of the residual--if most or all of the energy is correlated, the result is not pleasant

- The addition of a noisy signal before quantization discourages the residual from falling into a pattern correlated to the input

- All spectral components of the added noise contribute to this masking--no subband can claim more credit over the others

Equal Loudness

The Frequency Sensitivity of Hearing

- The human ear is not uniformly sensitive over the audio spectrum

- The sensitivity peaks at around 5 kHz
- To either side of the peak, residual noise would be less detectable

- It would be nice to reduce the level of the residual near the peak of hearing sensitivity

Principles of Noise Shaping

Filtering the quantization residual

- It is indeed possible to filter the residual signal to emphasize the residual in bands where our ears are less sensitive

- Reducing the residual level over a particular band can effectively lower the quantization floor beneath the LSB of a PCM sample in that band

- Applying noise shaping to an oversampled signal can push most of the residual noise outside the critical band, which will be filtered away down the processing line
- This technique is also known as Delta-Sigma Modulation, and is widely used in digital audio CODEC chips

Designing a Noise Shaping IIR

Transfer Function Manipulation

- The noise shaping system centers on two nodes, the system input and the quantizer output

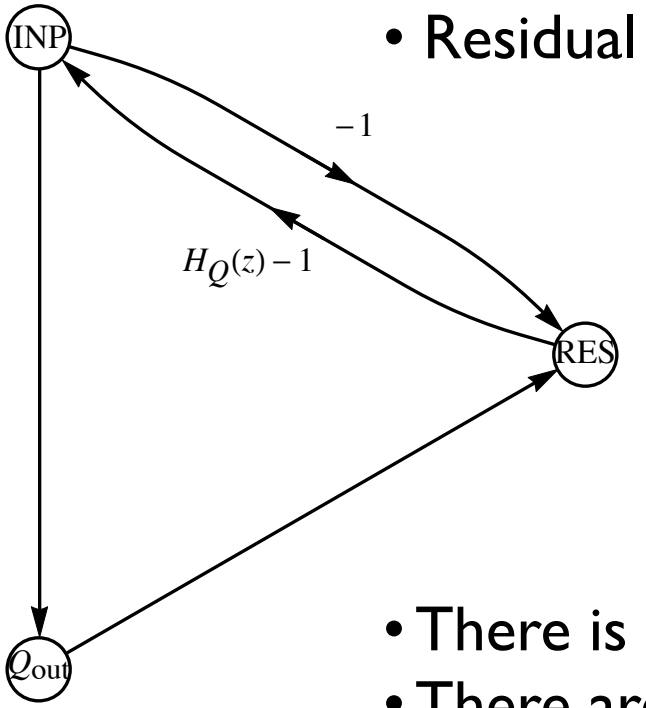
- Add a node that represents the residual signal, the difference between the output and the input

- The goal is to devise a signal processing system where the transfer function between the input and output is unity, yet the transfer function between the output and itself is a filter function

Designing a Noise Shaping IIR

Three-Node Configuration

- Three nodes: input, quantized output, and residual
- Input is sent to output and negated to residual
- Output is added to the residual node
- Residual is fed back and added to the input node



- There is one path from input to output with unity gain
- There are two paths from output to itself: one that does not leave the node and one that does the circuit
- The sum of the two output to output transfer functions is $1 + (H_Q(z) - 1) = H_Q(z)$

Designing a Noise Shaping IIR

Removing a Delay-Free Path

- H_Q is the desired transfer function applied to the residual signal, which is added into the quantization output
- The feed from the output back to the input, because this is a digital system, cannot have a delay-free component, else the system cannot be realized

- Here is the generic transfer function for a second-order filter

$$\frac{b_0 + b_1\, z^{-1} + b_2\, z^{-2}}{1 + a_1\, z^{-1} + a_2\, z^{-2}}$$

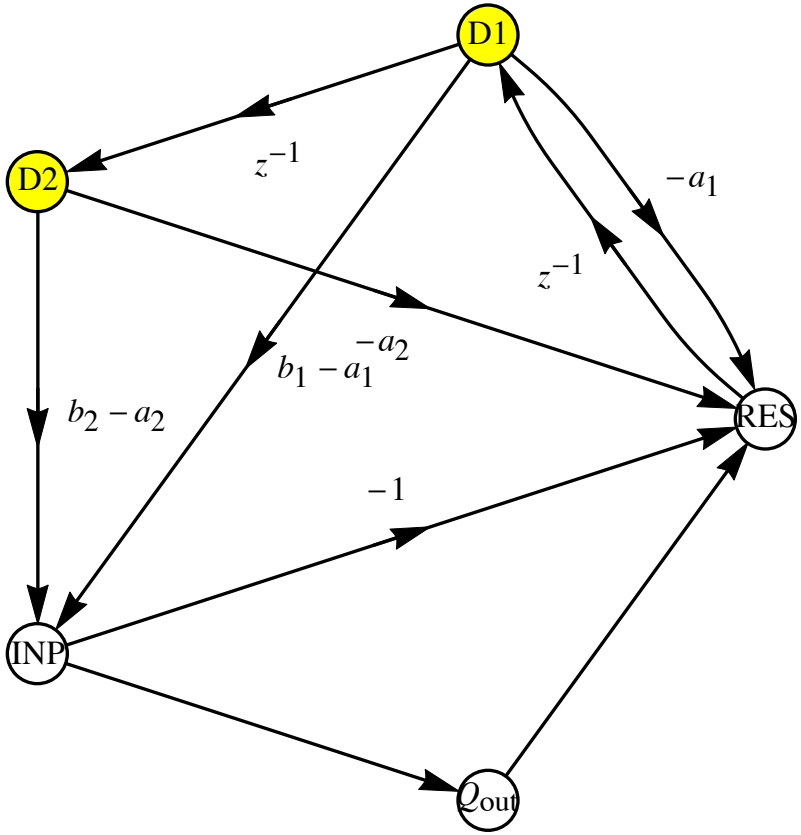
- The goal is to eliminate coefficient b_0, the multiplier for the feed-forward line with no delay

The Recursive IIR

Eliminating the Delay-Free Path

- Since the feedback multiplier is 1 - H_Q, there is a way to cancel the delay-free coefficient
- If the numerator of $H_Q(z)$ is normalized so that $b_0 = 1$, the effective feedback transfer function is

$$\frac{1 + b_1\, z^{-1} + b_2\, z^{-2}}{1 + a_1\, z^{-1} + a_2\, z^{-2}} - 1 = \frac{(b_1 - a_1)\, z^{-1} + (b_2 - a_2)\, z^{-2}}{1 + a_1\, z^{-1} + a_2\, z^{-2}}$$



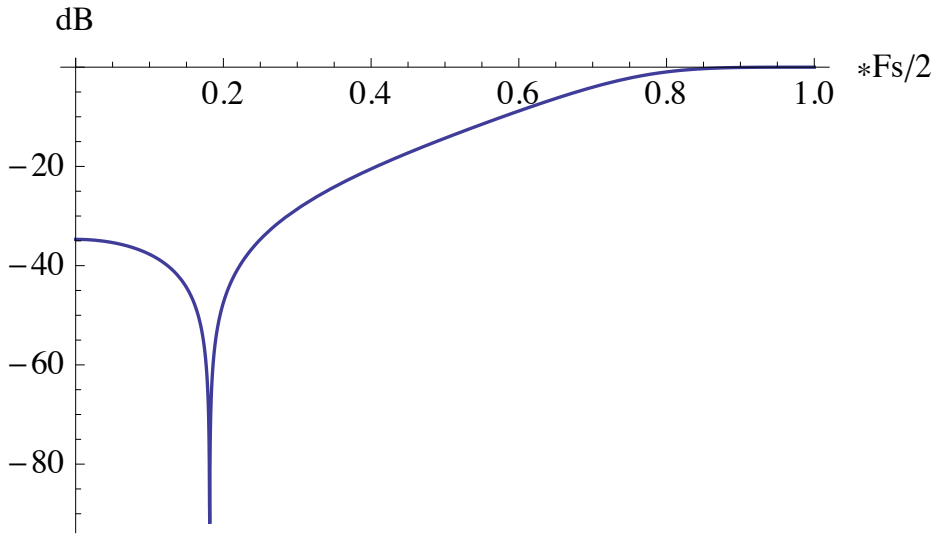
- This implies a trade-off in making a noise-shaping IIR
- The spectral shape can be arbitrarily defined, but the overall gain of the resulting filter is constrained by $b_0 = 1$

The Recursive IIR

Example Transfer Function

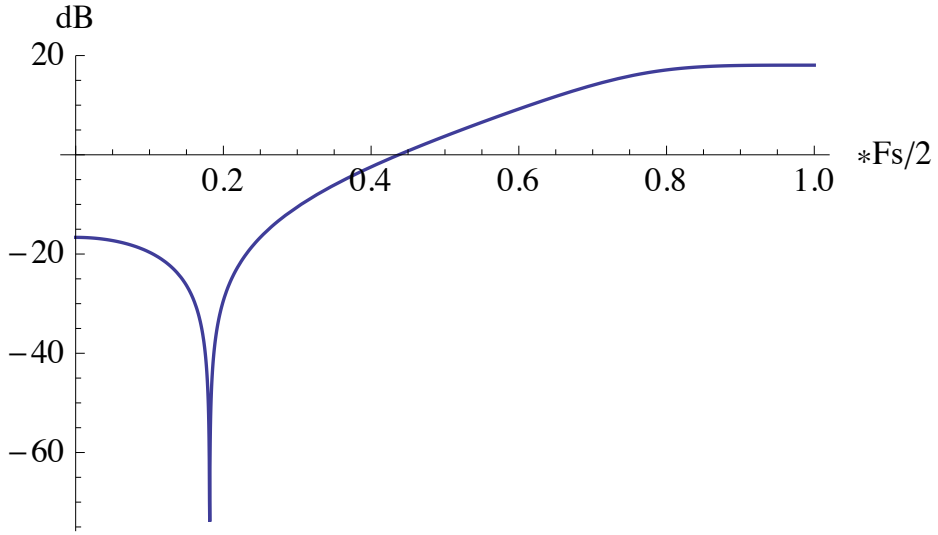
- Second-order Chebyshev highpass filter

$$\frac{0.125 - 0.210496\,z^{-1} + 0.125\,z^{-2}}{1 + 0.842994\,z^{-1} + 0.303489\,z^{-2}}$$



- With normalized transfer function numerator

$$\frac{1 - 1.68397\,z^{-1} + z^{-2}}{1 + 0.842994\,z^{-1} + 0.303489\,z^{-2}}$$

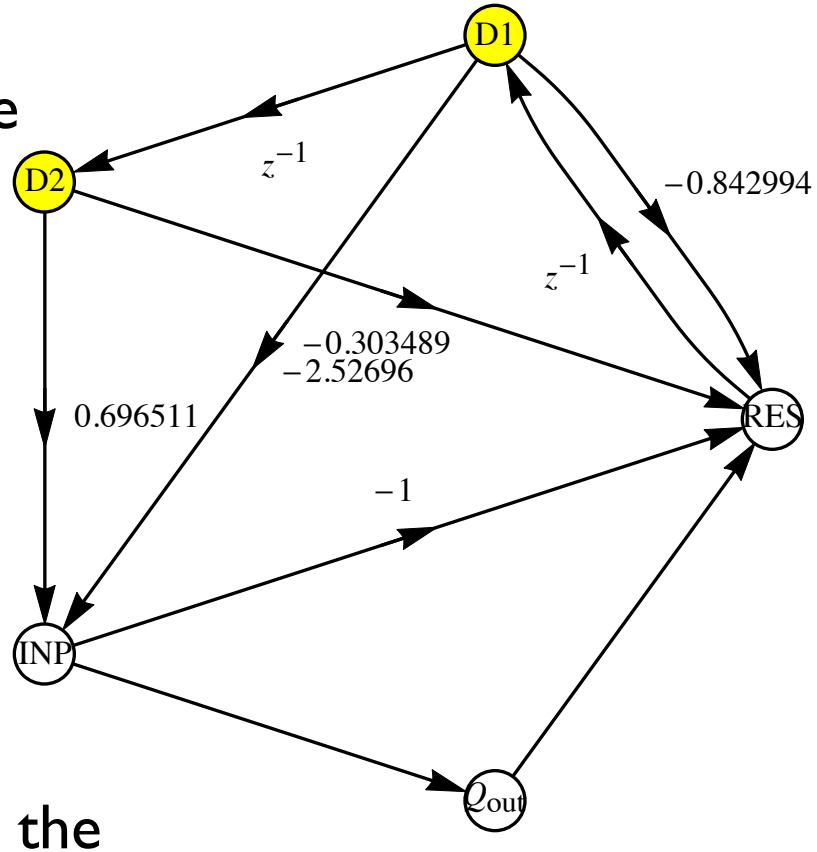


The Recursive IIR

Example Implementation

- Substitute $a_1 = 0.842994$, $a_2 = 0.303489$, $b_1 = -1.68397$, $b_2 = 1.0$ into the generic noise shaping system

- Put audio input into the INP node
- Inject optional dither and quantize at the Q_{out} node



- The residual signal that produces the quantized output will be spectrally shaped

Final Note on Noise Shaping

Accommodate for Residual Gain

- The restriction on the noise shaping transfer function may apply gain to the dither and quantization signals
- The PCM word that holds the quantizer output must contain enough headroom to clear the addition of the gained residual
- If the output word size is only a few bits, the transfer function of the noise shaping filter is restricted to have low normalized gain
- Too high a gain in the noise shaping filter can kick off instability of the quantizing system

Thanks for Your Attention

For Further Study

- James A. Moorer, "Whither Dither: Experience with High-Order Dithering Algorithms in the Studio", presented at the 95th AES Convention, October 1993, preprint #3747

- Robert C. Maher; "On the Nature of Granulation Noise in Uniform Quantization Systems"; Journal of the Audio Engineering Society; Vol. 40, No. 1/2, January/February 1992; pp. 12-20

- Norsworthy, Schreier, and Temes, eds.; *Delta-Sigma Data Converters*; IEEE Press; 1997