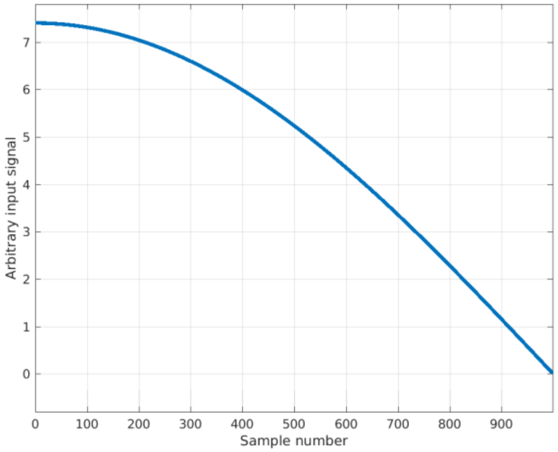
QUANTIZATION NOISE AS A FUNCTION OF WORD SIZE

Quantization Noise Analysis

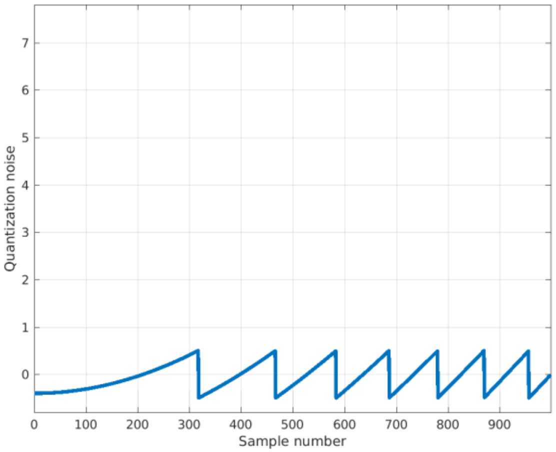
- Example
  - 3-bit quantization
  - 25.2 dB SNR (not very accurate with this waveform)

Input



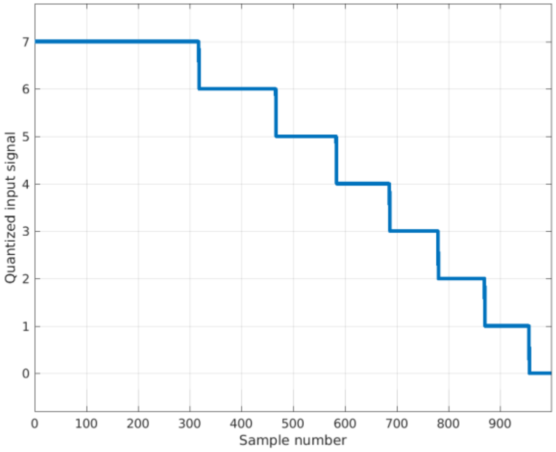
+

Quantization Noise



=

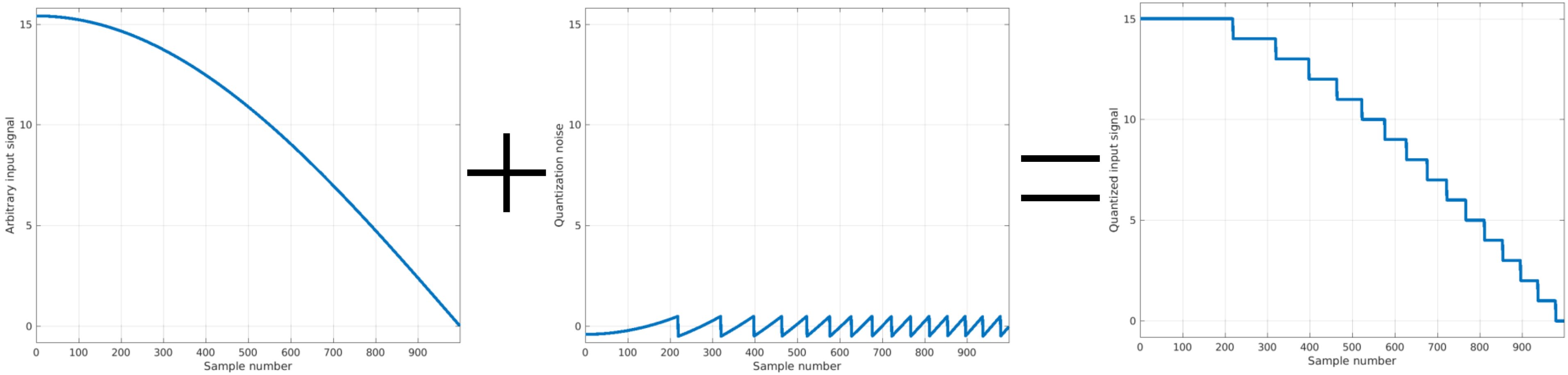
Output



Quantization Noise Analysis

- Example
  - 4-bit quantization
  - 31.5 dB SNR (more accurate)
    Note 31.5 dB = 25.2 dB + **6.3 dB**

Input **Quantization Noise** **Output**

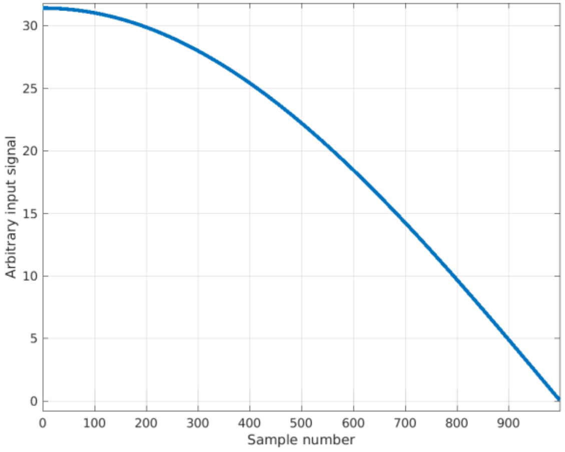


© B. Baas

Quantization Noise Analysis

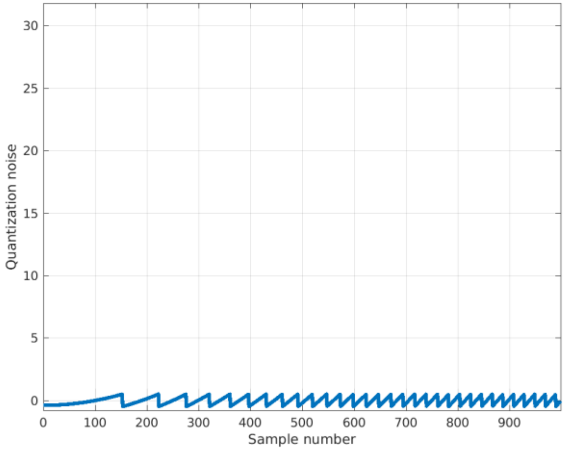
- Example
  - 5-bit quantization
  - 37.7 dB SNR (more accurate)
    Note 37.7 dB = 31.5 dB + **6.2 dB**

Input



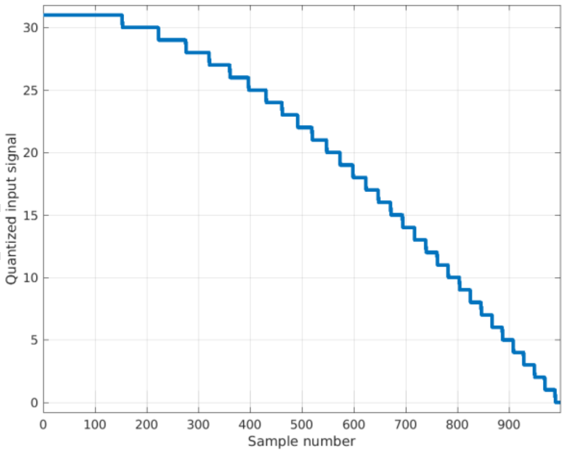
+

Quantization Noise



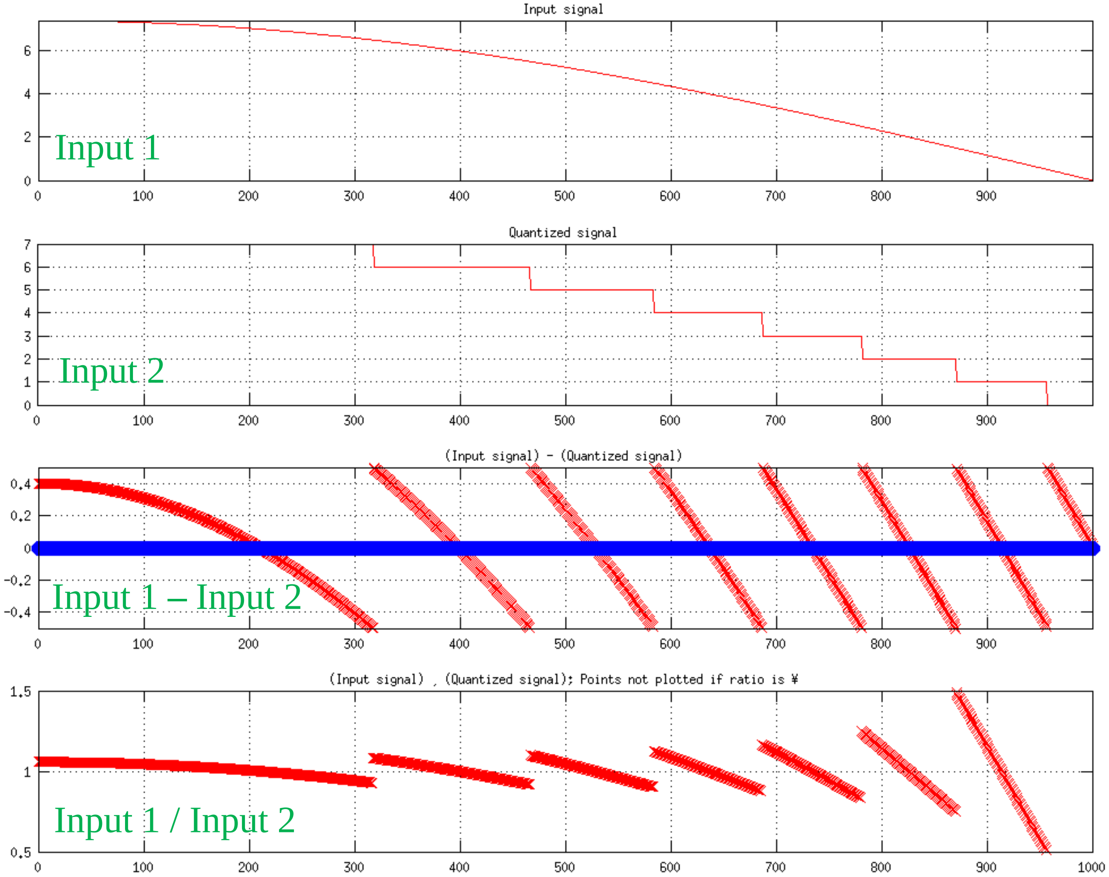
=

Output



`difff().m` matlab script

- Very useful in finding the character of the differences in two waveforms
  - Works with complex data
  - Also prints SNR and other calculated data
- Example
  - 3-bit quantization
  - 25.2 dB SNR (not accurate with this waveform)
- Posted on the 281 web page under “matlab: simple functions useful for 281”



Quantization Noise Analysis

- Matlab code that generates the plots on the previous slides
- Copy, paste, run, and modify it yourself!

```
% quant_plot.m
%
% 2020/01/09  Added linewidth, xlabel, larger plots (BB)
% 2017/01/10  Fixed out = in + noise (BB)
% 2011/01/11  Written (BB)

% Set these
NumSamples = 1000;
NumBits    = 3;     % resolution of quantization of input signal
PrintOn    = 1;     % 1 = print plots to .png files; png is 4x smaller than
tiff
lwidth     = 3;     % line width in plots

% Calculate some things
MaxValue      = 2^NumBits -1;
x             = 0:(NumSamples -1);
y             = (MaxValue+0.4) * cos(0.5 * pi * x /NumSamples);
y_quantized   = round(y);
y_quant_noise = y_quantized - y;        % output = input + noise

% Print and plot
fprintf('Valid range of output values = %i - %i\n\n', 0, MaxValue);

figure(1); clf;
plot(x,y,'linewidth',lwidth);
xlabel('Sample number');
ylabel('Arbitrary input signal');
grid on;
axis([0 NumSamples-1 -0.8 (MaxValue+0.8)]);
if (PrintOn)  print('-f1', strcat('1.',num2str(NumBits),'bits.png'), '-dpng');
  end

figure(2); clf;
plot(x,y_quantized,'linewidth',lwidth);
xlabel('Sample number');
ylabel('Quantized input signal');
grid on;
axis([0 NumSamples-1 -0.8 (MaxValue+0.8)]);
if (PrintOn)  print('-f2', strcat('2.',num2str(NumBits),'bits.png'), '-dpng');
  end

figure(3); clf;
plot(x,y_quant_noise,'linewidth',lwidth);
xlabel('Sample number');
ylabel('Quantization noise');
grid on;
axis([0 NumSamples-1 -0.8 (MaxValue+0.8)]);
if (PrintOn)  print('-f3', strcat('3.',num2str(NumBits),'bits.png'), '-dpng');
  end

figure(4); clf;
difff(y, y_quantized, 'Input signal', 'Quantized signal');
if (PrintOn)  print('-f4', strcat('4.',num2str(NumBits),'bits.png'), '-dpng');
  end
```