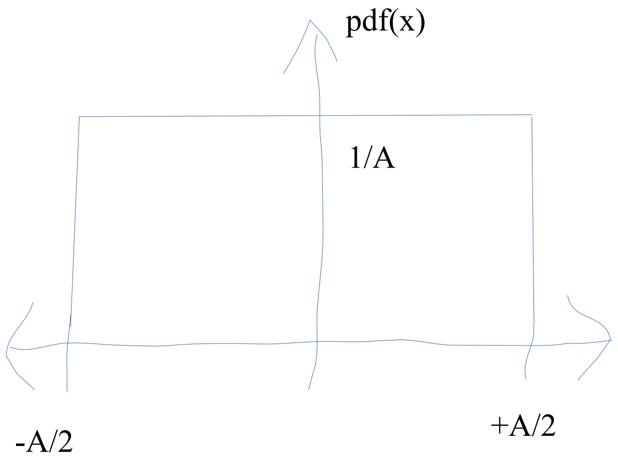
Digital Signal Processing 2/ Advanced Digital Signal Processing Lecture 2, Quantization, SNR

Gerald Schuller, TU Ilmenau

Quantization Signal to Noise Ratio (SNR).

Assume we have a A/D converter with a quantizer with a certain number of bits (say N bits), what is the resulting Signal to Noise Ratio (SNR) of this quantizer? The SNR is defined as the ratio of the expectation of the signal power to the expectation of the noise power. In our case, the expectation of the noise power is the expectation of the quantization error power. We already have the expectation of the quantization error power as $\Delta^2/12$.
So what we still need for the SNR is the average or **expectation of the signal power**. How do we obtain this?
Basically we can take the same approach as we did for the expectation of the power of the quantization error (which is basically the second moment of the distribution of the quantization error). So what we need to know from our signal is its **probability distribution**. For the quantization error it was a uniform distribution between $-\Delta/2$ and $+\Delta/2$.

A very **simple case** would be a **uniformly distributed signal** with amplitude A/2, which has values between -A/2 up to +A/2.



So we could again use our formula for the average power, but now for our signal x:

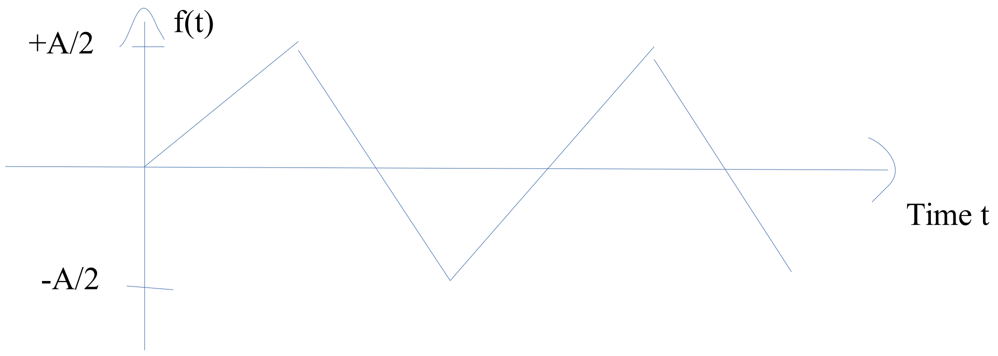
$$E(x^2) = \int_{-A/2}^{A/2} x^2 \cdot p(x)\, dx$$

So here we have the same type of signal, and the resulting expectation of the power (its second moment, assumed we have a zero mean signal) is obtained by using our previous formula, and replace Δ by A. The resulting power is: $\frac{A^2}{12}$

Which signals have this property? One example is uniformly distributed random values (basically like our quantization error). Matlab or Octave produces this kind of signal if we use the command: rand()-0.5 (the 0.5 is to make the distribution centered around 0). In Python, the command for a random number is: *scipy.rand()*.

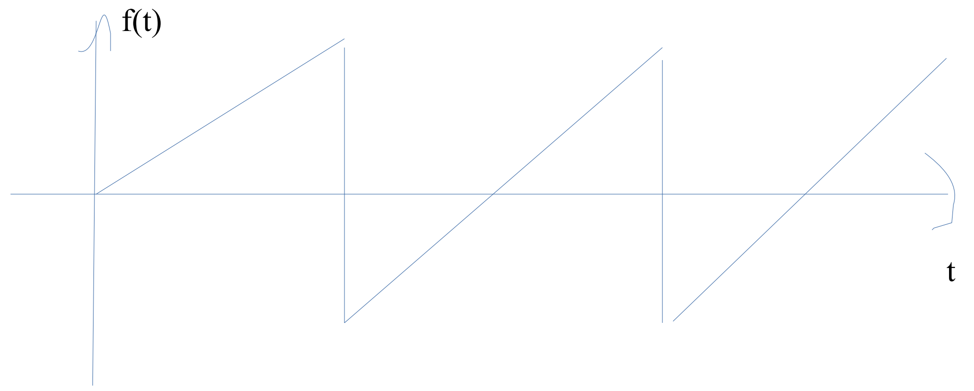
Observe: Speech has a non-uniform pdf, it is usually modeled by a Laplacian distribution or a gausssian mixture model, so it doesn't apply to this case!

An example for a uniform pdf: a **triangular wave**:



How do we obtain its pdf? One can imagine the vertical axis (the function value) covered by small intervals, and each interval is then passed in the same time-span. This means that the resulting pdf is also uniform!

A further example: A **sawtooth wave**:



Again we can make the same argument, each small interval of our function value is covered in the same time-span, hence we obtain a uniform distribution.

We now have seen a few examples which fulfil our assumption of a uniform distribution (realistic examples), and we know: their expectation of their power is $A^2/12$. So what does this then mean for the SNR? The **SNR** is just the ratio

$$SNR = \frac{A^2/12}{\Delta^2/12} = \frac{A^2}{\Delta^2}$$

If we assume our signal is full range, meaning the maximum values of our A/D converter is -A/2 and +A/2 (the signal goes to the maximum), we can compute the step size Δ if we know the **number of bits** of converter, and if we assume uniform quantization step sizes. Assume we have **N bits** in our converter. This means we have 2^N

quantization intervals. We obtain Δ by dividing the full range by this number,

$$\Delta = \frac{A}{2^N}$$

Plug this in the SNR equation, and we obtain:

$$SNR = \frac{A^2}{\Delta^2} = \frac{A^2}{(A/2^N)^2} = 2^{2N}$$

This is now quite a simple result! But usually, the SNR is given in dB, so lets convert it into dB:

$$SNR_{dB} = 10 \cdot \log_{10}(2^{2N}) = 10 \cdot 2N \cdot \log_{10}(2) \approx$$

$$\approx 10 \cdot 2N \cdot 0.301\, dB = N \cdot 6.02\, dB$$

This is now our famous **rule of thumb**, that **each bit** more gives you about **6 dB more SNR**. But observe that the above formula only holds for uniformly distributed full range signals! (the signal is between -A/2 and +A/2, using all possible values of our converter)

What happens if the signal is not full range? What is the SNR if we have a signal with reduced range? Assume our signal has an amplitude of A/c, with a factor c>1.
We can then simply plug this into our equation:

$$SNR = \frac{(A/c)^2}{\Delta^2} = \frac{(A/c)^2}{(A/2^N)^2} = \frac{2^{2N}}{c^2}$$

in dB:

$$SNR_{dB}=10\cdot\log_{10}\left(\frac{2^{2N}}{c^2}\right)=10\cdot 2N\cdot\log_{10}(2)-20\cdot\log_{10}(c)\approx$$

$$\approx 10\cdot 2N\cdot 0.301\,dB-20\cdot\log_{10}(c)=$$

$$=N\cdot 6.02\,dB-20\cdot\log_{10}(c)$$

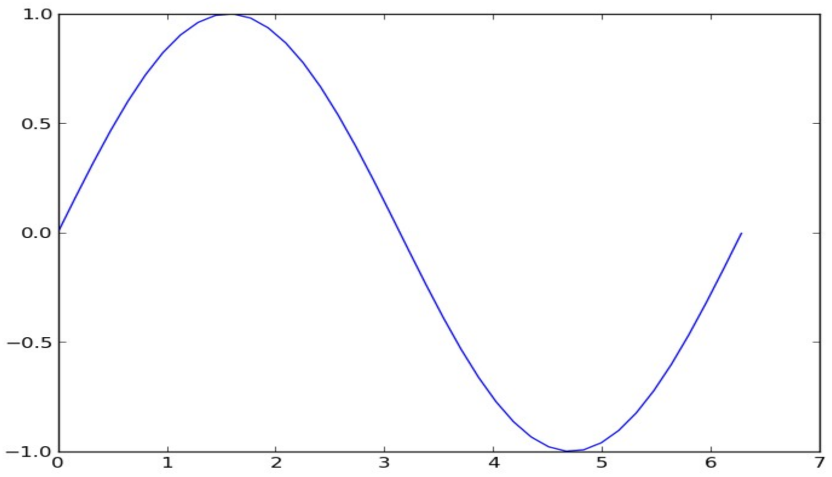
The last term, the $20\cdot\log 10(c)$, is the number of dB which we are below our full range. This means we **reduce our SNR** by this number of **dB** which we are **below full range**!

Example: We have a 16 bit quantiser, then the SNR for uniformly distributed full range signals would be

$$SNR=6.02\cdot 16\,dB=96.32\,dB$$

Now assume we have the same signal, but 20dB below full range (meaning only 1/10th of the full range). Then the resulting SNR would be only 96.32-20=76.32 dB! This is considerably less. This also shows why it is important not to make the safety margin to full range too big! So for instance our sound engineer should keep the signal as big as possible, without ever reaching full range to avoid clipping the signal.

The other assumption we made concerned the type of signal we quantize. What if we don't have a uniformly distributed signal? As we saw, speech signals are best modeled by a Laplacian distribution or a gaussian mixture model, and similar for audio signals. Even a simple sine wave does not fulfill this assumption of a uniform distribution. What is the pdf of a simple sine wave?



Observe: If a sinusoid represents a full range signal, its values are from $-A/2$ to $+A/2$, as in the previous cases.

Remark: This plot was made in python with the following terminal commands:

```
ipython --pylab
t=linspace(0,6.28,100);
s=sin(t);
plot(t,s)
```