

January 28, 2005

➤ No Class: Monday, Jan 31st

➤ **Office Hours™**

⊕ 1:10 - 2:10 pm, Monday, A223 Cook

⊕ Not next week (no class Monday Jan 31st!)

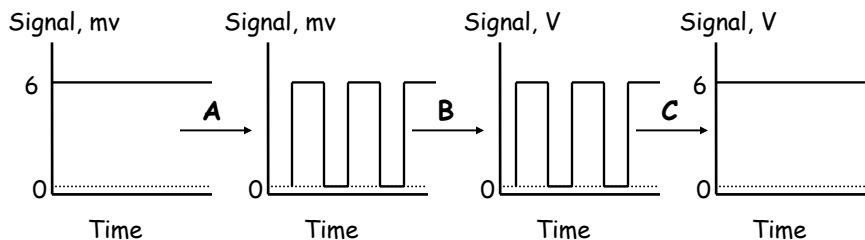
⊕ 1-2 pm, Thursday, 300 Waterman

⊕ 3-4 pm, Friday, 300 Waterman

1

Tuned Amplification: *Chopper Amplifiers*

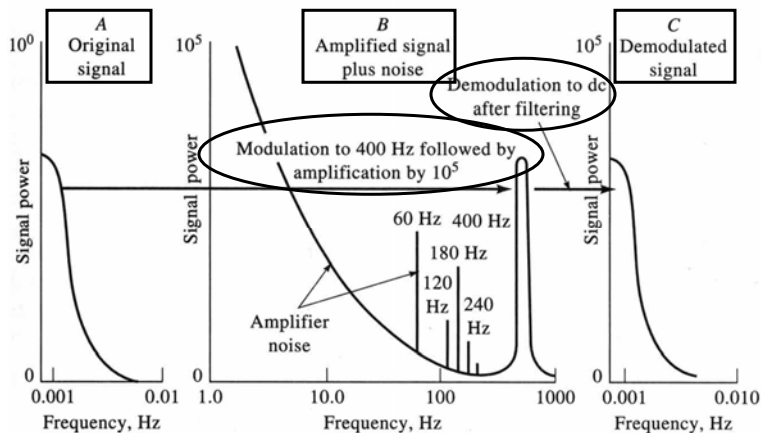
- If we *amplify* the signal at the modulation frequency, then we can also *increase the signal* while we *reduce the noise*:



A: modulate **B:** amplify **C:** demodulate

2

Chopper Amplifiers: In the Frequency Domain



3

Lock-In Amplification

- Tuned (chopper) amplification is limited by stability of f_0 reference
 - this limits the effective Δf to about 1 Hz
- Lock-in amplification tunes the amplifier to both f_0 as well as the phase of the signal
 - results in an effective reduction of Δf to about 0.01 Hz

4

Signal Averaging

- Need a *repeatable* signal (time domain method)
- Sum digitally-stored replicate signals
- How does this improve S/N?

Signal: increases *linearly* with number (n) of replicate signals

Noise: increases as $(n)^{1/2}$

So, **S/N** increases as $(n)^{1/2}$

5

Signal Averaging: Example

- Suppose we wish to *mass* a **10-mg** object on an analytical balance ($\sigma = 0.1 \text{ mg}$)

For a single (n = 1) measurement:

$$S = 10. \text{ mg}, N = 0.1 \text{ mg} \rightarrow \mathbf{S/N = 100}$$

For n = 4:

$$S = n \times 10. \text{ mg} = 4 \times 10. \text{ mg} = 40. \text{ mg}$$

$$N = \sigma_T = (n(\sigma)^2)^{1/2} = (4(0.1)^2)^{1/2} = 2(0.1) = 0.2 \text{ mg}$$

$$\mathbf{S/N = 40./0.2 = 200}$$

For n = 16: **S/N = 400**

6

Signal Averaging: General

- In general, we can say:

$$(S/N)_n = (n)^{1/2} (S/N)_{n=1}$$

- Diminishing returns: *Suppose, it takes 1 second per measurement?*

<u>S/N</u>	<u>n</u>	<u>Time</u>
3	1	1 sec
6	4	4 sec
30	100	100 sec
300	10,000	2.8 <i>hours</i>
3000	1,000,000	11.6 <i>days</i>

7

More Signal Averaging

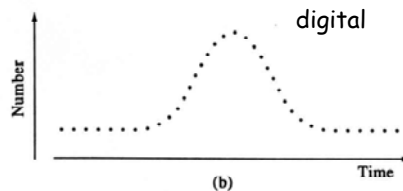
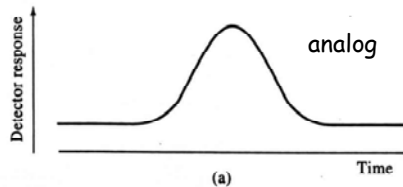
- **Signal must be digitized .**
... *How?*

- **Digitization frequency?**
- *Nyquist Theorem*

- **Precision of digitization?**
- *how many bits?*

8-bit = $2^8 = 256$ (0.4 %)

12-bit = $2^{12} = 4096$ (0.02 %)



8