Software signal processing

Joshua Smith
Intel Research Seattle

Software Signal Processing

- Use software to make sensitive measurements
- Case study: electric field sensing
- You will build an electric field sensor in lab 3
  - Non-contact hand measurement (like magic!)
  - Software (de)-modulation for very sensitive measurements
  - Same basic measurement technique used in accelerometer
  - We will get signal-to-noise gain by software operations

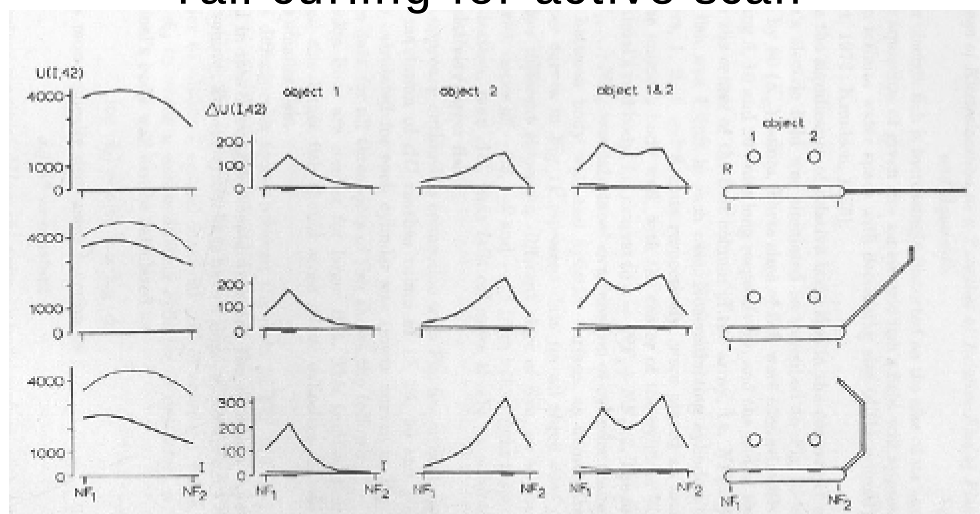
- We will need
  - some basic electronics
  - some math facts
  - some signal processing

Electrosensory Fish

- Weakly electric fish generate and sense electric fields
- Measure conductivity “images”
- Frequency range .1Hz – 10KHz

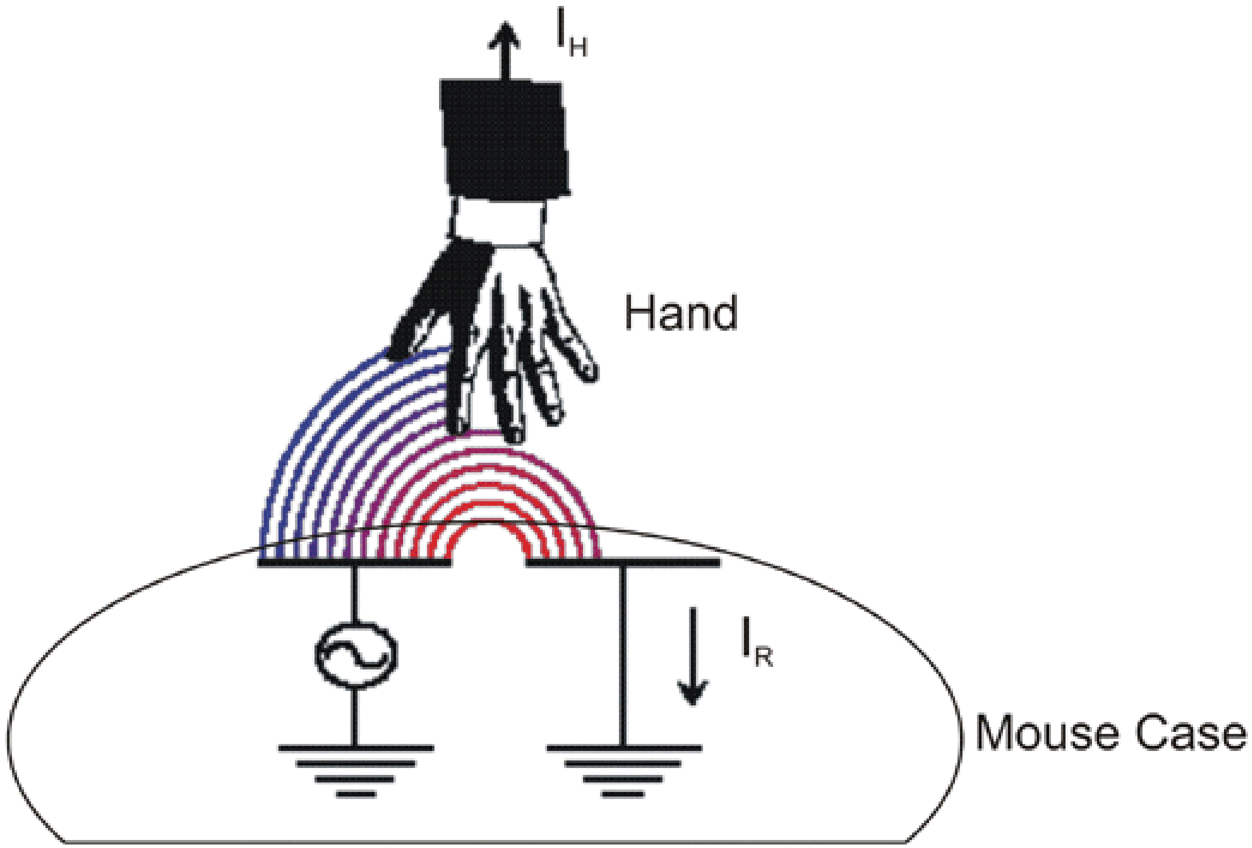
Black ghost knife fish (*Apteronotus albifrons*)
Continuous wave, 1KHz

Tail curling for active scan

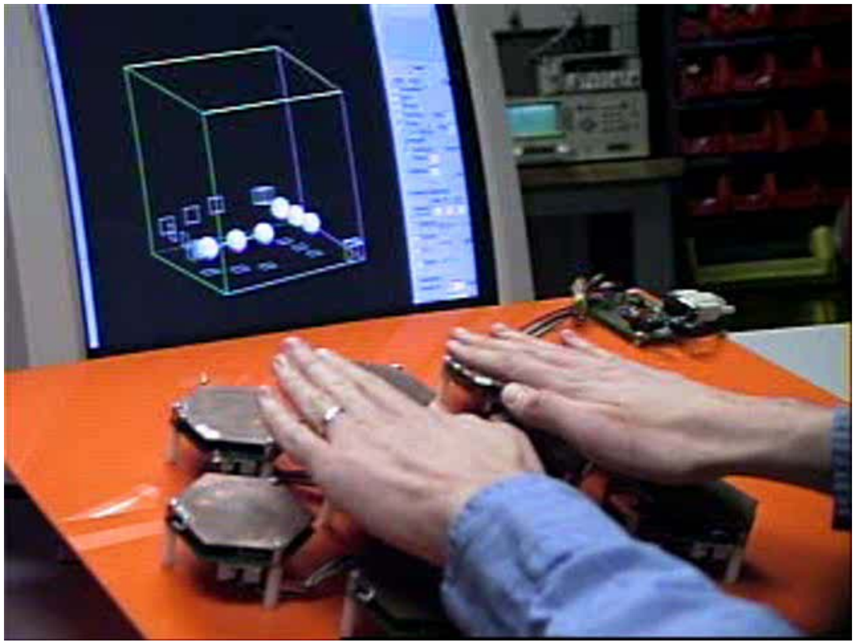


W. Heiligenberg. Studies of Brain Function, Vol. 1: Principles of Electrolocation and Jamming Avoidance Springer-Verlag, New York, 1977.

Electric Field Sensing for input devices



Cool stuff you can do with E-Field sensing



Basic electronics

- Voltage sources, current sources, and Ohm’s law
- AC signals
- Resistance, capacitance, inductance, impedance
- Op amps
  - Comparator
  - Current (“transimpedance”) amplifier
  - Inverting amplifier
  - Differentiator
  - Integrator
  - Follower

Voltage & Current sources

- "Voltage source"
  - Example: microcontroller output pin
  - Provides *defined* voltage (e.g. 5V)
  - Provides current too, but current depends on load (resistance)
  - Imagine a control system that adjusts current to keep voltage fixed
- "Current source"
  - Example: some transducers
  - Provides *defined* current
  - Voltage depends on load

- Ohm's law (V=IR) relates voltage, current, and load (resistance)

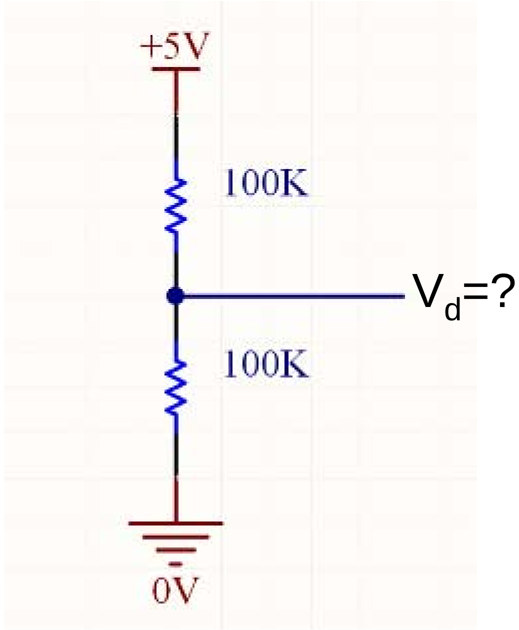
Ohm's law and voltage divider

Need 3 physics facts:

- 1. Ohm's law: V=IR   (I=V/R)
  - Microcontroller output pin at 5V, 100K load ➔ I=5V/100K = 50μA
  - Microcontroller output pin at 5V, 200K load ➔ I=5V/200K = 25μA
  - Microcontroller output pin at 5V,   1K load ➔ I=5V/1K = 5mA
- 2. Resistors in series add
- 3. Current is conserved ("Kirchoff's current law")

Voltage divider

- Lump 2 series resistors together (200K)
- Find current through both: I=5V/200K=25μA
- Now plug this I into V_d=IR for 2nd resistor
- V_d=25μA * 100K = $25*10^{-6} * 10^5$ = 2.5V
- General voltage divider formula: $V_d=VR_2/(R_1+R_2)$



Using complex numbers to represent AC signals

Math facts

- "AC signals": time varying (vs. steady "DC signals")
- DC signal has magnitude only
- AC signal has magnitude and phase
  - Complex numbers good for representing magnitude and phase
- Math facts:
  - e is Euler's const, 2.718…
  - j*j = -1 ("unit imaginary")
  - $e^{x+y} = e^x e^y$
  - $d\{e^{ct}\}/dt = ce^{ct}$
  - Can write complex numbers as
    - Real & imaginary parts: x+jy, or
    - Polar (magnitude & phase): $re^{j\theta}$
  - $e^{j\theta}=\cos(\theta)+j\sin(\theta)$ ("Euler's equation")

Using complex numbers to represent AC signals

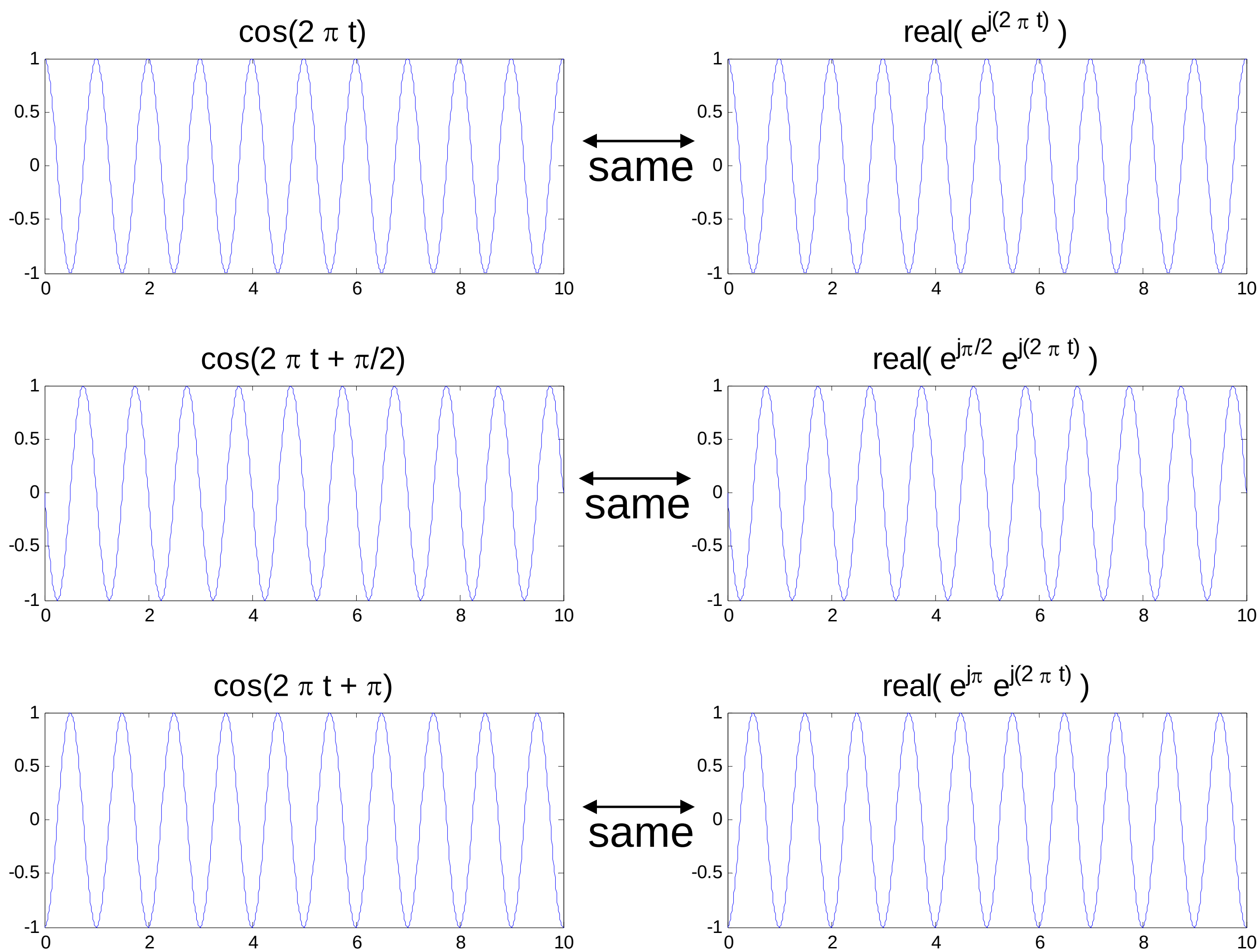
How to do it

- Pretend signals are complex during calculations
- Take the real part at the end to find out what really happens
- Multiply signal by real number ←→ magnitude change
- Multiply by complex number ←→ phase *and* magnitude
  - Example: S'pose we want to represent $\cos(2\pi ft+\Delta)$ (phase shift Δ)
  - In complex exponentials, it's $e^{j(2\pi ft+\Delta)}=e^{j\Delta}\, e^{j(2\pi ft)}$
  - Passive components (inductors & capacitors) affect phase and mag
  - We will model the effect of passives with a single complex number
    - "Complex impedance"
  - Bonus: taking derivatives is easy with this representation

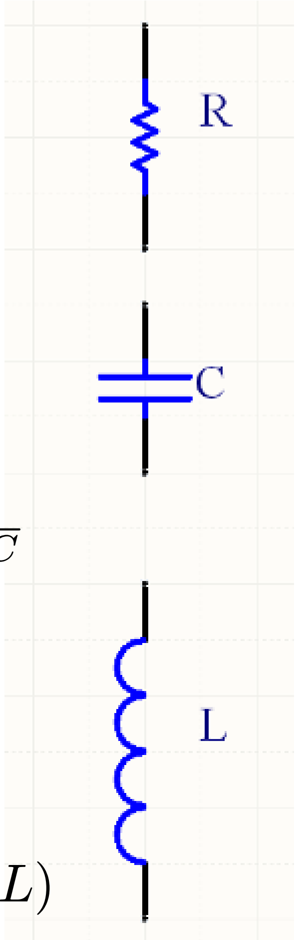
Using complex numbers to represent AC signals

Examples

Cosine | Complex exponential



R,C,L



- R: resistor
  - Non-perfect conductor
  - Turns electrical energy into heat
  - V=IR
- C: capacitor
  - Two conductive plates, not in contact
  - Stores energy in electric field

$$Q = CV$$
$$\frac{d}{dt}Q = \frac{d}{dt}CV \implies I = C\frac{dV}{dt}$$
$$\text{Let } V = V_0 e^{j2\pi f t} \implies I = Cj2\pi f V_0 e^{j2\pi f t} \implies V = I\frac{-j}{2\pi f C}$$

  - Blocks DC…passes AC
- L: inductor
  - Coil of wire
  - Stores energy in magnetic field

$$V = L\frac{dI}{dt}$$
$$\text{Let } I = I_0 e^{j2\pi f t} \implies V = Lj2\pi f I_0 e^{j2\pi f t} \implies V = I(j2\pi f L)$$

  - Passes DC…blocks AC

Z

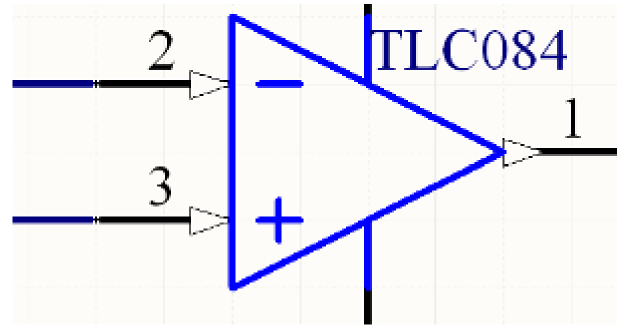
- Z: Impedance
  - AC generalization of resistance
  - Models what passive components do to AC signals
  - Frequency dependent, unlike resistance
- Resistor: “real impedance” = R
- Capacitor: “negative imaginary impedance” = $-j/C2\pi f = -j/\omega C$
  - $\omega=2\pi f$
- Inductor: “positive imaginary impedance” = $j2\pi fL = j\omega L$

- You can lump a network of resistors, capacitors, and inductors together into a single complex impedance with real and imaginary components
- Capacitive and inductive parts of impedance can cancel each other out
  - when they do, it’s called resonance

Operational amplifiers

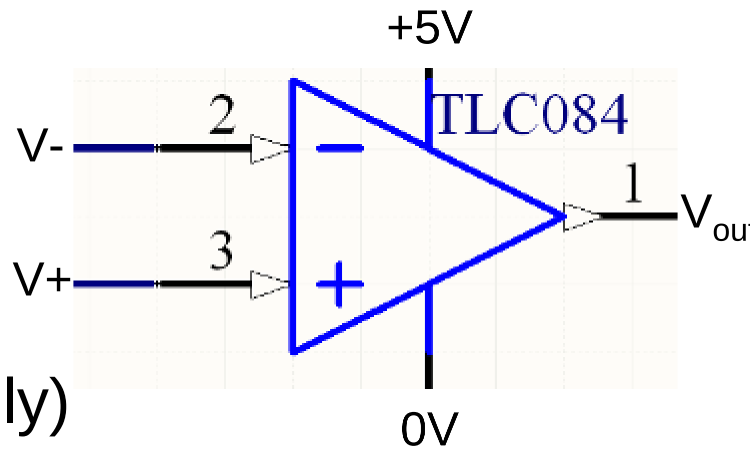
- Amplify voltages (increase voltage)
- Turn weak ("high impedance") signal into robust ("low impedance") signal
- Perform mathematical *operations* on signals (in analog)
  - E.g. sum, difference, differentiation, integration, etc
- History
  - Originally computers were text only; signal processing meant analog
  - Next DSPs moved some signal processing functions to digital
  - Now microcontrollers becoming powerful enough to do DSP functions
  - "Software defined radio"
  - Computation can happen in software; still need opamps for amplification
    - But, some kinds of amplification can even happen in software:
      - "processing gain," "coding gain"
  - Signal processing is historically EE; becoming embedded software topic

Op Amps



- Op amps come 1,2,4 to a package (we will use quad)
- Op amp has two inputs, +ve & -ve.
  - Rule 1: Inputs are “sense only”…no current goes into the inputs
- It amplifies the difference between these inputs
- With a feedback network in place, it tries to ensure:
  - Rule 2: Voltage on inputs is equal
    - as if inputs are shorted together…“virtual short”
    - more common term is “virtual ground,” but this is less accurate
- Using rules 1 and 2 we can understand what op amps do

Comparator

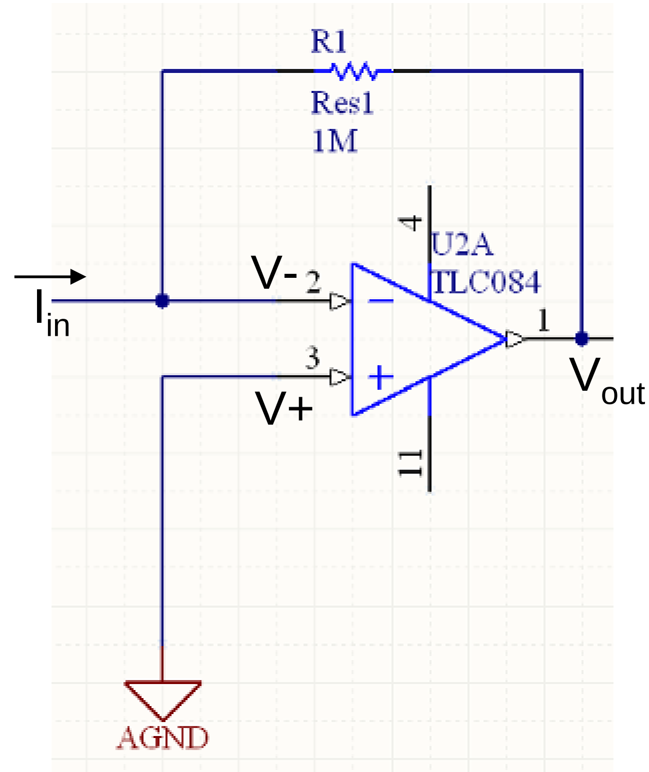


- Used in earlier ADC examples
- No feedback (so Rule 2 won't apply)
- V_{out} = T{g*(V+ - V-)} [g big, say 10^6]
  - T{ } means threshold s.t. V_{out} doesn't exceed rails
- In practice
  - V+ > V- ➔ V_{out} = +5
  - V+ < V- ➔ V_{out} = 0

Transimpedance amplifier

- Produces output voltage proportional to input current
- AGND = V+ = 0V
- By 2, V- = V+, so V- = 0V
- Suppose I_{in} = 1μA
- By 1, no current enters inverting input
- All current must go through R1
- V_{out}-V- = -1μA * 10^6 Ω
- ➔V_{out} = -1V

- Generally, V_{out} = - I_{in} * R1



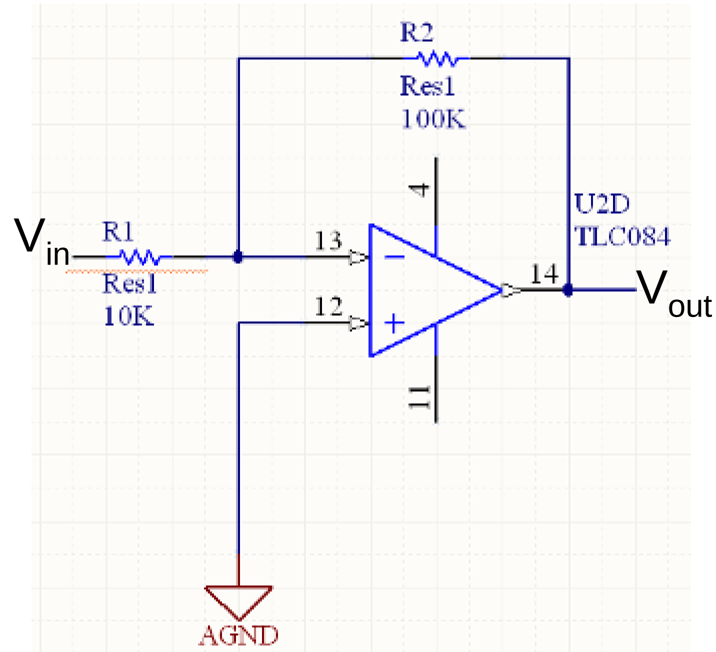
1. No current into inputs
2. V- = V+

Inverting (voltage) amplifier

- S'pose V_{in}=100mV
- Then I_{in}=100mV/10K = 10μA
- By rule 1, that current goes through R2
- By rule 2, V- = 0
- V_{out}-V- = V_{out} = -10μA*100K = -1V

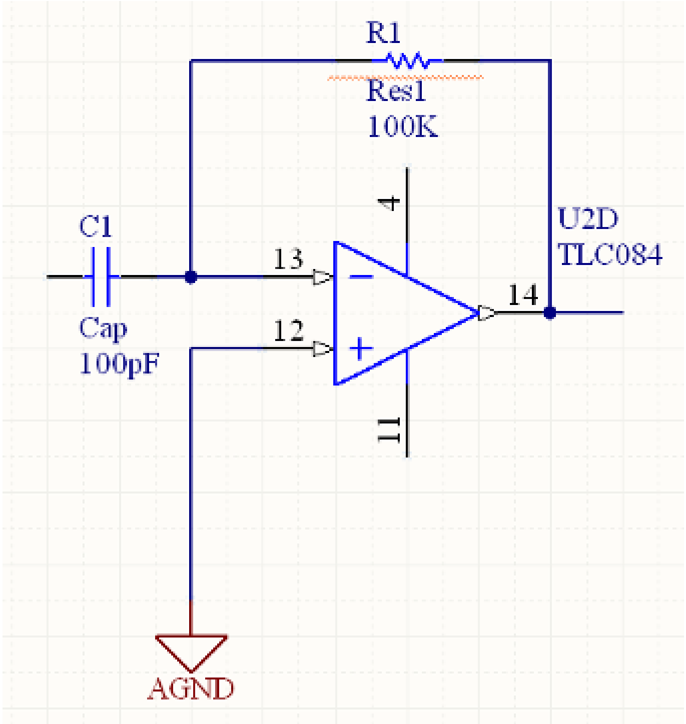
- In general, I_{in}= V_{in} / R1
- V_{out} = -I_{in}R2 = - V_{in} R2 / R1
- ➔ Gain = V_{out} / V_{in} = - R2 / R1

- In this case, gain = 100K / 10K = -10
- -10 * 100mV = -1V. Yep.



1. No current into inputs
2. V- = V+

Differentiator



- Q=CV ➔dQ/dt = C dV/dt ➔ I = C dV/dt
- I_{in}=C dV_{in}/dt
- Now pretend it's a transimpedance amp:
  - V_{out} = - I_{in} * R
  - ➔ V_{out} = - RC dV_{in}/dt
- Output voltage is proportional to derivative of input voltage!

Integrator

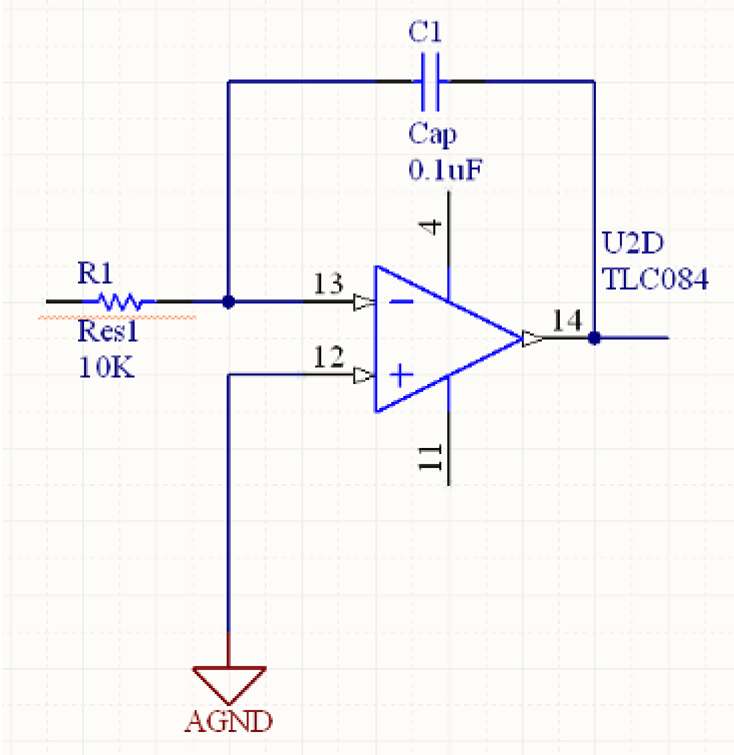
$$I_{in} = V_{in}/R_1$$

$$Q_1 = \int I_{in} dt$$

$$Q_1 = -C_1 V_{out}$$

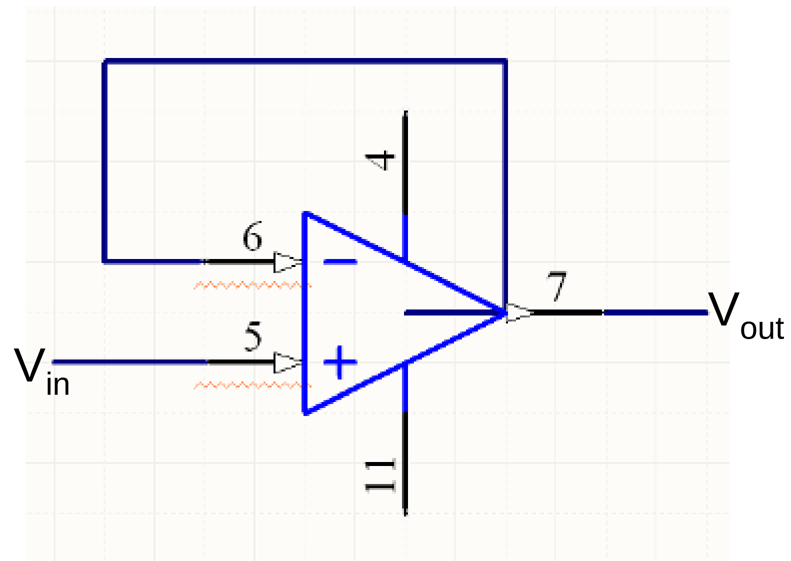
$$\implies V_{out} = -\frac{1}{C_1} \int \frac{V_{in}}{R_1} dt$$

$$\implies V_{out} = -\frac{1}{R_1 C_1} \int V_{in} dt$$



Follower

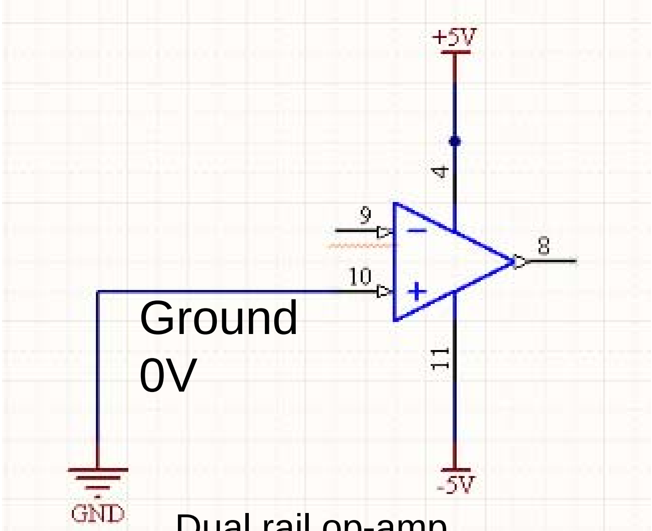
- Because of direct connection, V- = V_{out}
- Rule 2 ➔ V- = V+, so
- $V_{out} = V_{in}$



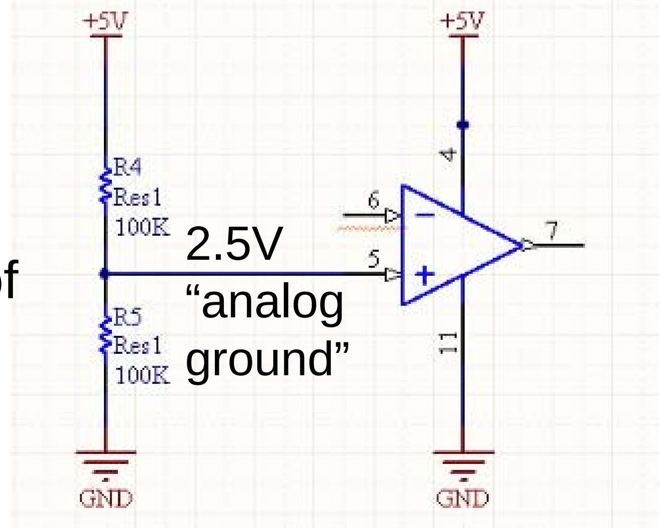
1. No current into inputs
2. V- = V+

Op Amp power supply

- Dual rail: 2 pwr supplies, +ve & -ve
  - Can handle negative voltages
  - “old school”
- Single supply op amps
  - Signal must stay positive
  - Use Vcc/2 as “analog ground”
  - Becoming more common now, esp in battery powered devices
  - Sometimes good idea to buffer output of voltage divider with a follower



Dual rail op-amp



Single supply op-amp

End of basic electronics

Noise

Why modulated sensing?

- Johnson noise
  - Broadband thermal noise
- Shot noise
  - Individual electrons…not usually a problem
- "1/f" "flicker" "pink" noise
  - Worse at lower frequencies
  - ➔ do better if we can move to higher frequencies
- 60Hz pickup

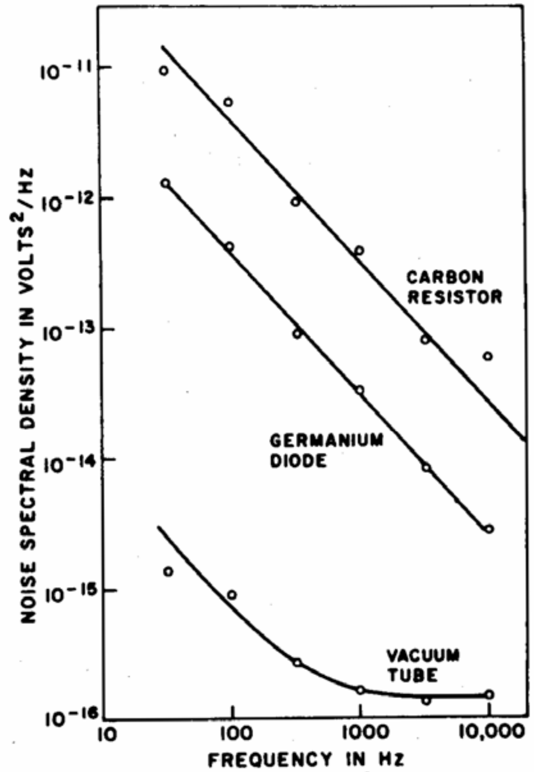


FIGURE 5 Typical electrical noise spectra for some current-carrying devices: 50 KΩ carbon resistor, 2N2000 germanium diode-connected transistor, and 12AX7 vacuum tube. (Reproduced from Brophy).[4]

From W.H. Press, "Flicker noises in astronomy and elsewhere," Comments on astrophysics 7: 103-119. 1978.

Modulation

- What is it?
  - In music, changing key
  - In old time radio, shifting a signal from one frequency to another
  - Ex: voice (10kHz “baseband” sig.) modulated up to 560kHz at radio station
  - Baseband voice signal is recovered when radio receiver demodulates
  - More generally, modulation schemes allow us to use analog channels to communicate either analog or digital information
    - Amplitude Modulation (AM), Frequency Modulation (FM), Frequency hopping spread spectrum (FHSS), direct sequence spread spectrum (DSSS), etc
- What is it good for?
  - Sensitive measurements
    - Sensed signal more effectively shares channel with noise ➔ better SNR
  - Channel sharing: multiple users can communicate at once
    - Without modulation, there could be only one radio station in a given area
    - One radio can chose one of many channels to tune in (demodulate)
  - Faster communication
    - Multiple bits share the channel simultaneously ➔ more bits per sec
    - “Modem” == “Modulator-demodulator”

Just a little more math

- Convolution theorem:
  - Multiplication in time domain ←→ convolution in frequency domain
- What is convolution?
  - Takes two functions a(t), b(t), produces a 3rd: c(τ)
    - Flip one function (invert time axis)
    - slide it along to offset of τ
    - Integrate product of these fns over all t
    - Each offset τ gives a value of c(τ)

$$c(\tau) = \int a(t)b(\tau - t)dt$$

-t → b is flipped wrt time

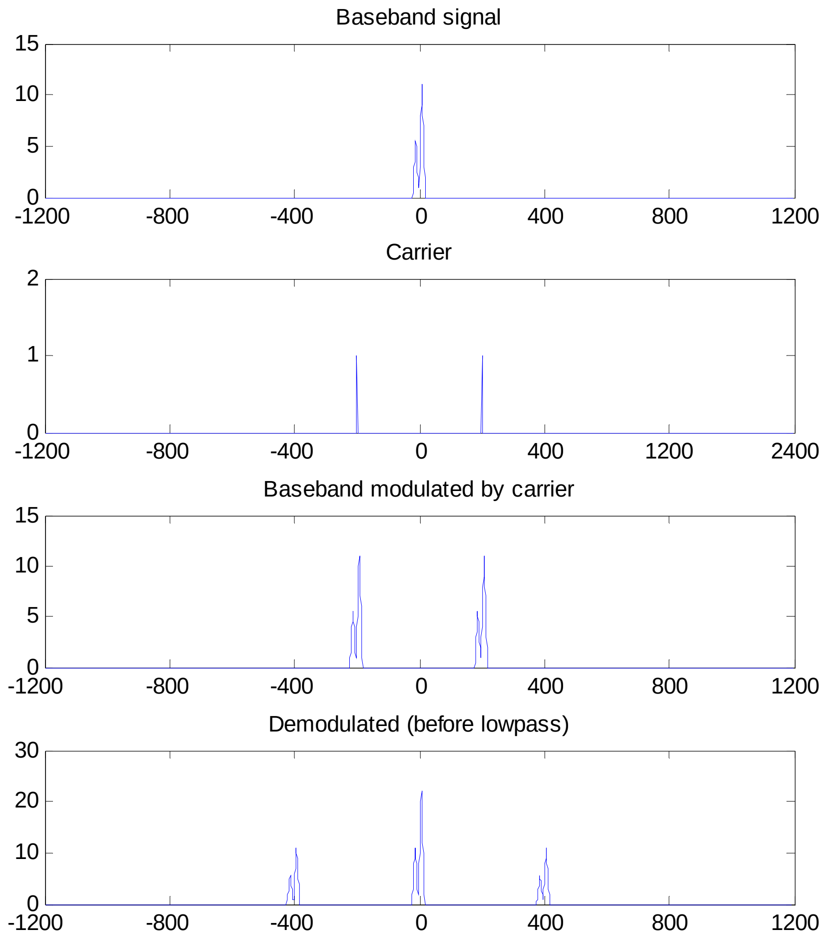
Each τ is a different overlapping of a(t) and (time-inverted) b(-t)

Amplitude modulation

Frequency domain view

In time domain, modulation is multiplication: (baseband x carrier)

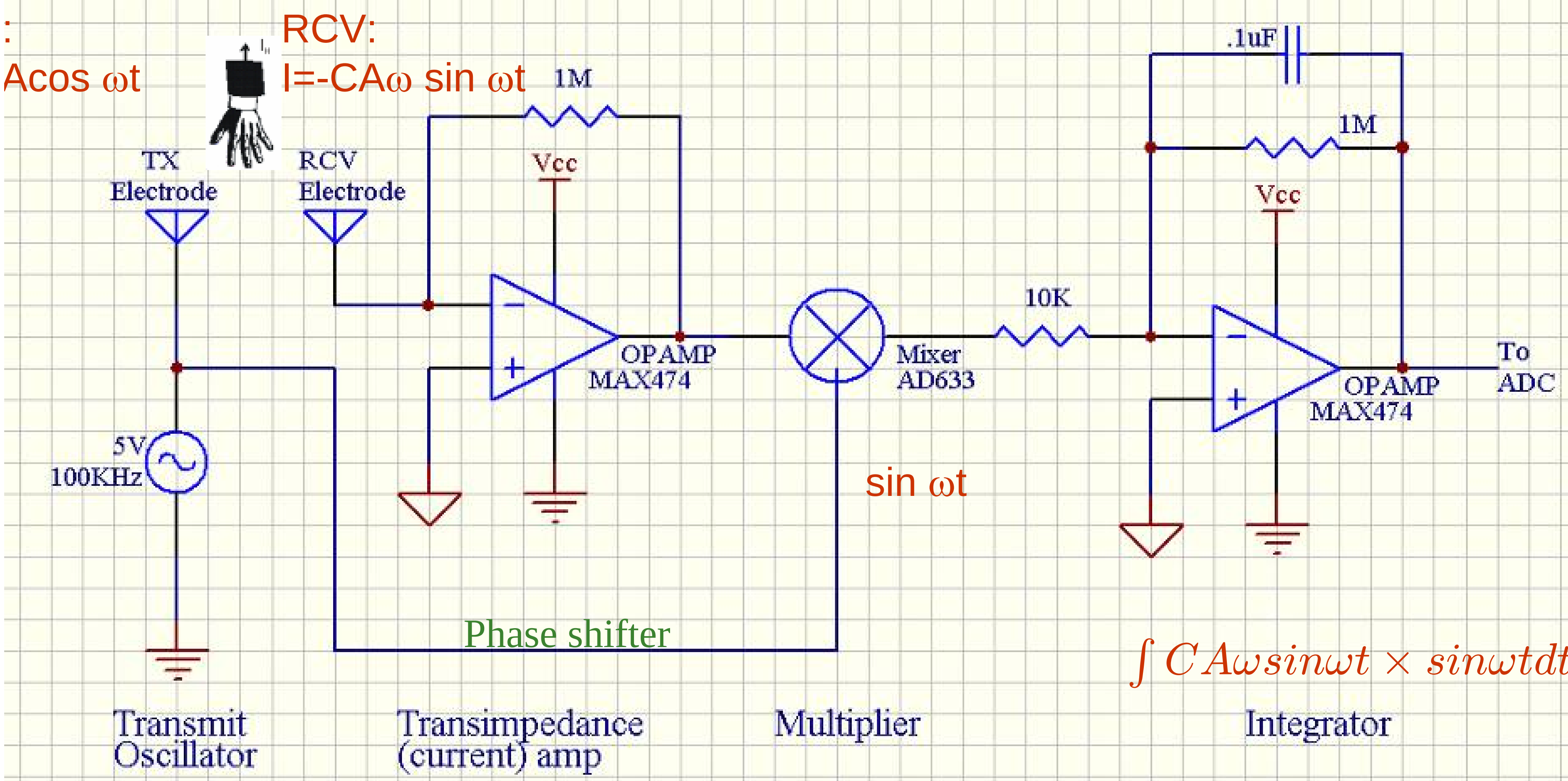
In freq domain (shown here) modulation is convolution of baseband w/ carrier



Horizontal axes: frequency (in arbitrary units)

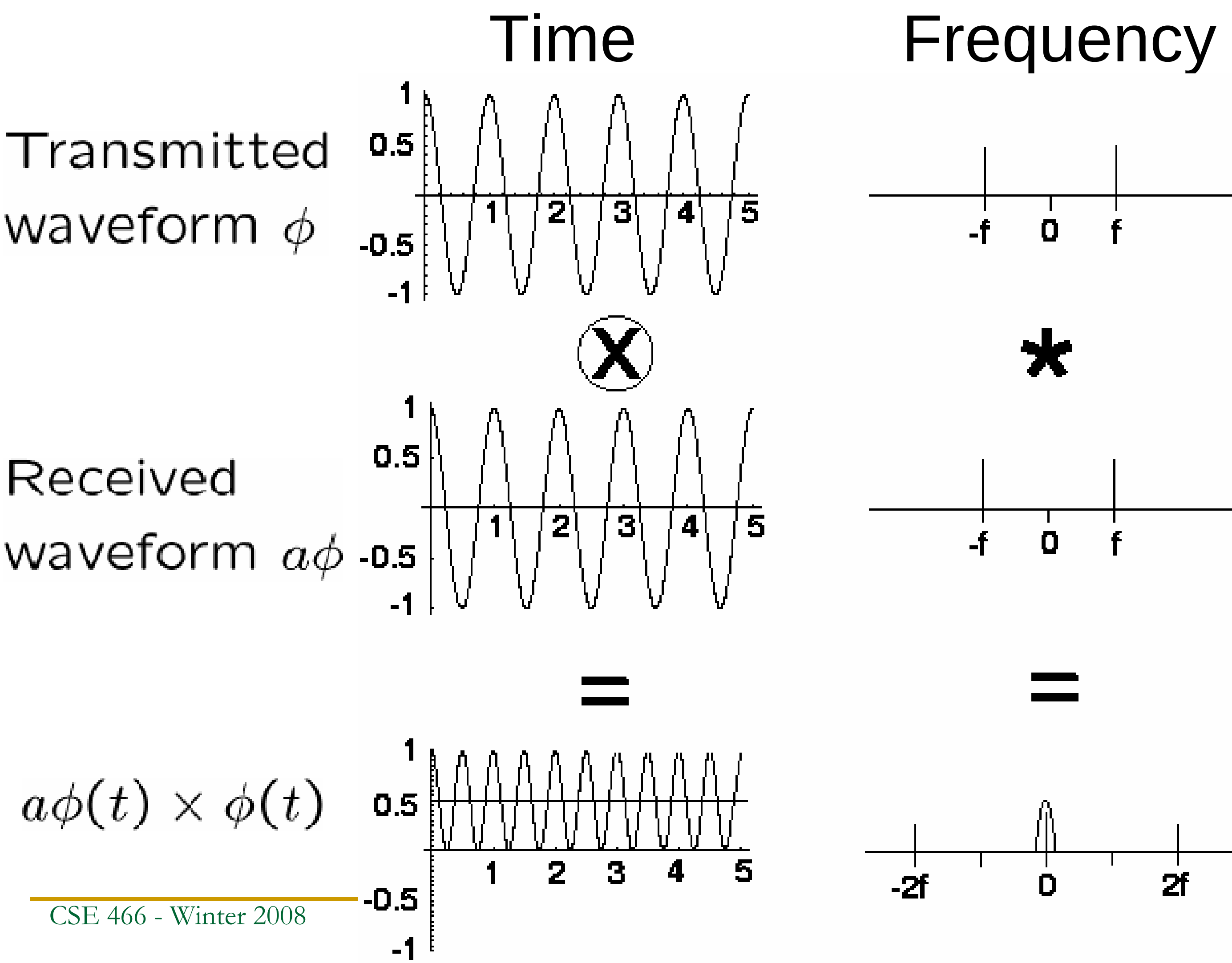
Vertical axes: amplitude (arbitrary units)

Basic Electric Field Sensing Circuit (Analog)



Synchronous Demodulation

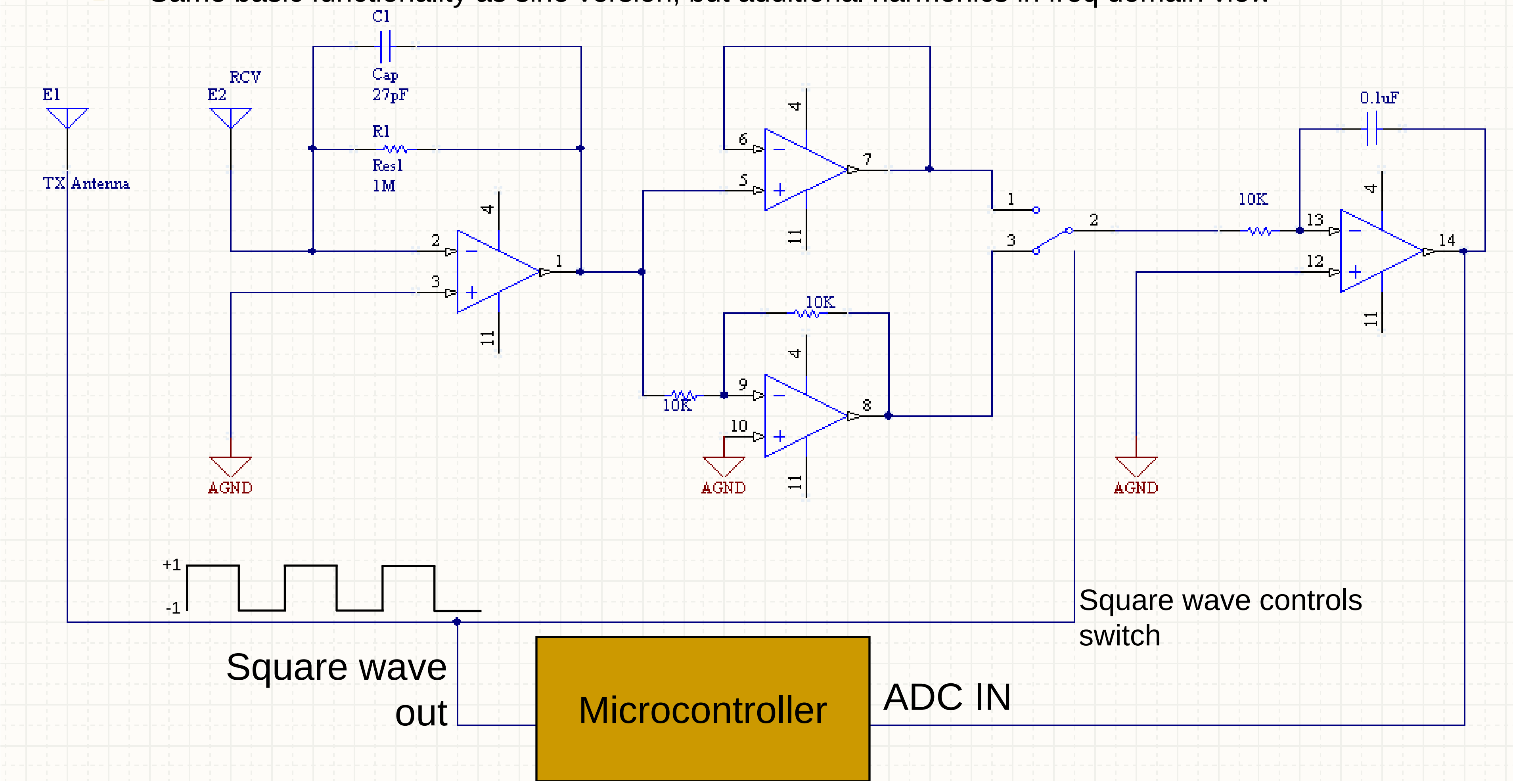
Time and frequency domain view



Electric Field Sensing circuit

Variant 2 (no analog multiplier)

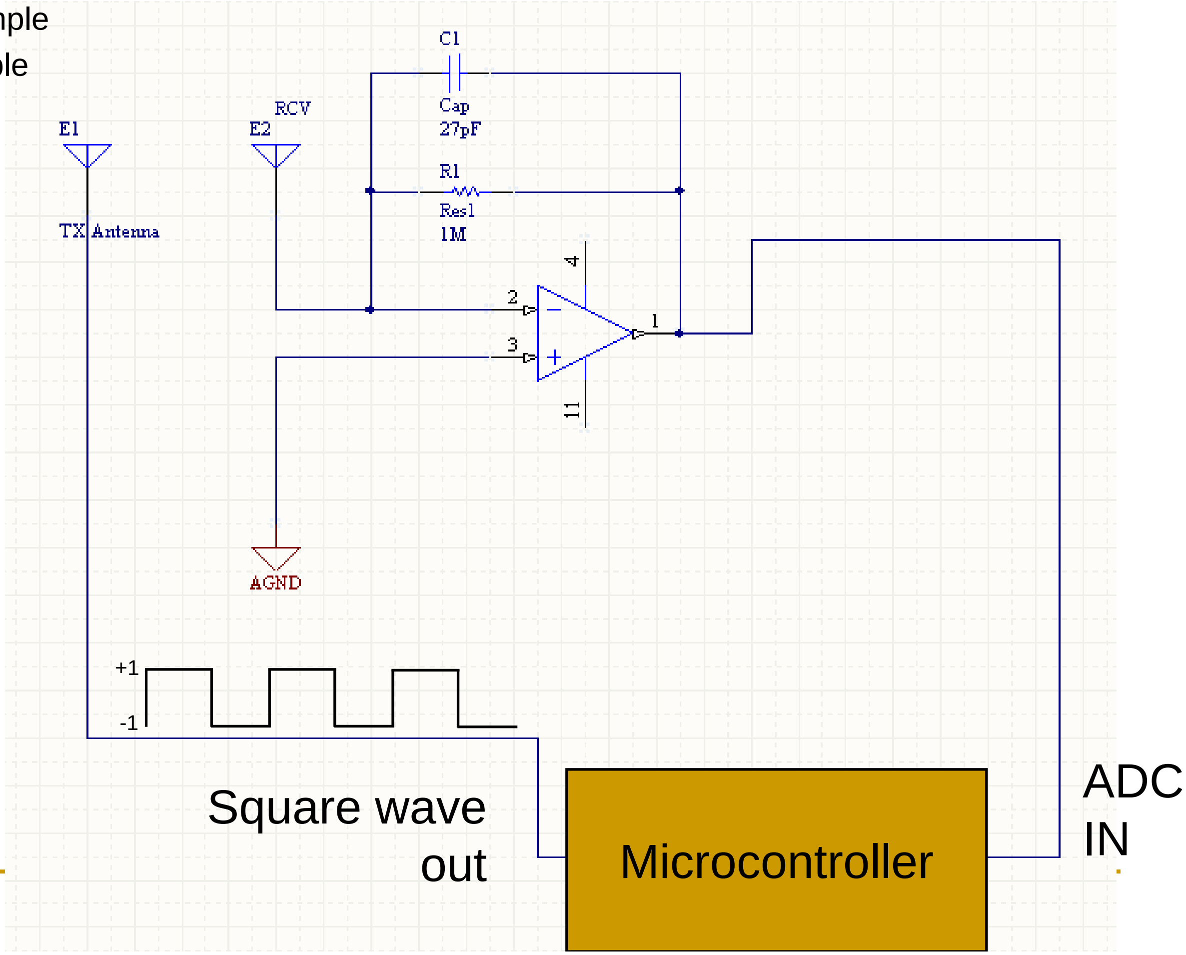
- Replace sine wave TX with square wave (+1, -1)
- Multiply using just an inverter & switch (+1: do not invert; -1: invert)
- End with Low Pass Filter or integrator as before
- Same basic functionality as sine version, but additional harmonics in freq domain view



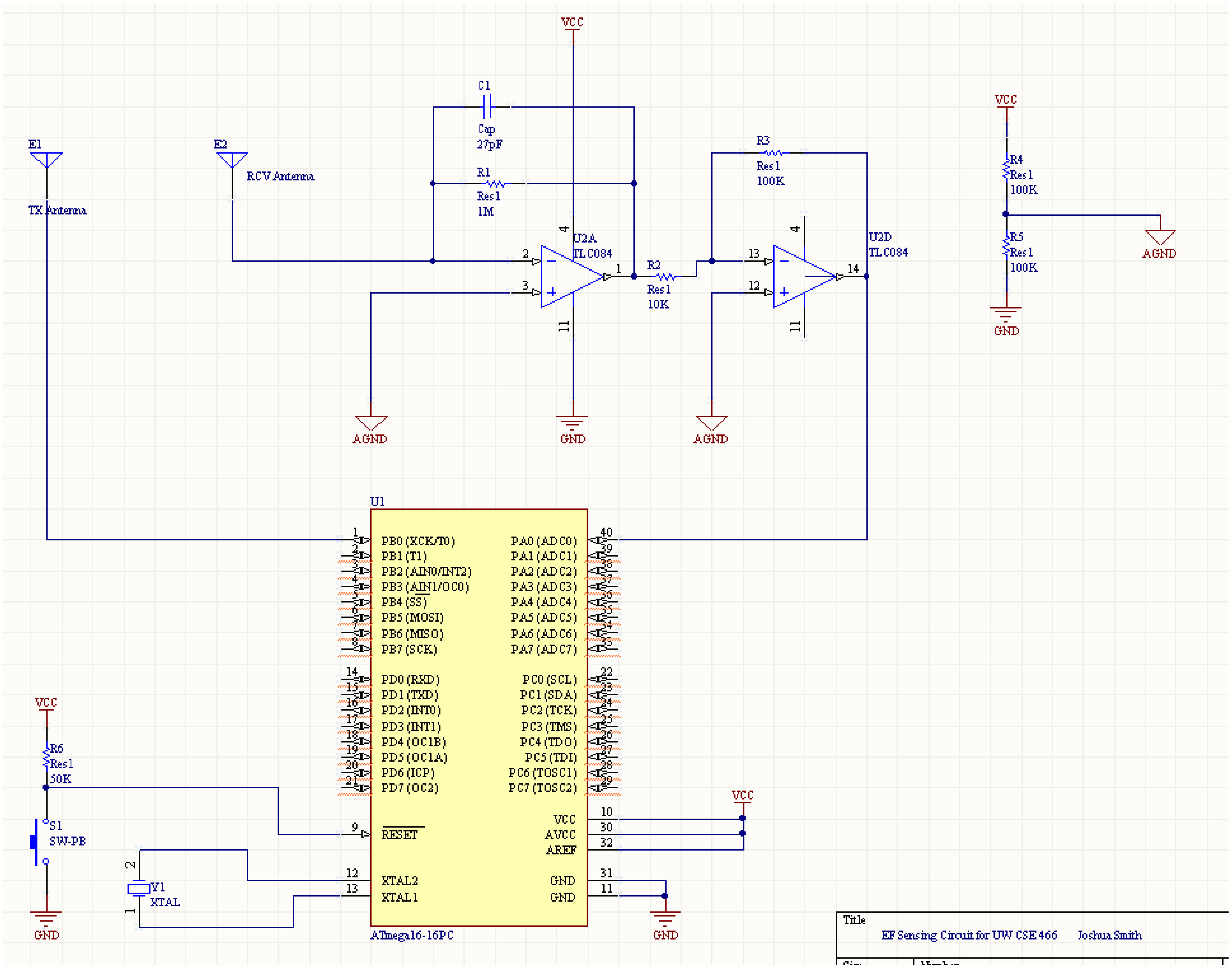
Electric Field Sensing circuit

Variant 3 (implement demodulation in software)

- For nsamps desired integration
- Assume square wave TX (+1, -1)
- After signal conditioning, signal goes direct to ADC
- Acc = sum_i T_i * R_i ➔
  - When TX high, acc = acc + sample
  - When TX low, acc = acc - sample



Lab 3 Schematic



Lab 3 pseudo-code

```
// Set PORTB as output
// Set ADC0 as input; configure ADC
NSAMPS = 200; // Try different values of NSAMPS
//Look at SNR/update rate tradeoff
acc = 0; // acc should be a 16 bit variable
For (i=0; i<NSAMPS; i++) {
            SET PORTB HIGH
            acc = acc + ADCVALUE
            SET PORTB LOW
            acc = acc - ADCVALUE
}
Return acc
```

Why is this implementing inner product correlation? Imagine unrolling the loop.
We'll write ADC_1, ADC_2, ADC_3, … for the 1st, 2nd, 3rd, … ADCVALUE

$acc = ADC_1 – ADC_2 + ADC_3 – ADC_4 + ADC_5 – ADC_6 +…$

$acc = +1*ADC_1 + -1*ADC_2 + +1*ADC_3 + -1*ADC_4 +…$

$acc = C_1*ADC_1 + C_2*ADC_2 + C_3*ADC_3 + C_4*ADC_4 + …$

where C_i is the ith sample of the carrier

acc = <C,ADC> Inner product of the carrier vector with the ADC sample vector

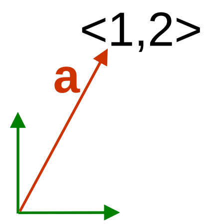
End of intro to E-Field Sensing

Outline

- Demo of EF Sensing circuit
- A completely different way to think about modulation
- Synchronous demodulation vs diode demodulation

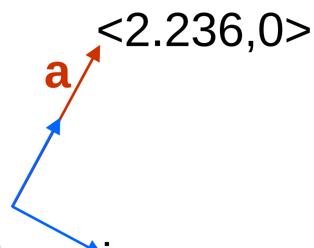
More math facts!

- Think of a signal as a vector of samples
- Vector lives in a vector space, defined by bases
- Same vector can be represented in different bases



Vector **a** in
some basis

Length:
Sqrt(1^2+2^2)=2.236

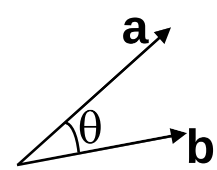


Vector **a** in
another basis

Length:
Sqrt(2.236^2)=2.236

Still more math facts…

- Remember inner (“dot”) product:
  - <1,2,3,4 | 5,6,7,8> = 1*5 + 2*6 + 3*7 + 4*8=70
  - **<a**|**b>** = |**a**|*|**b**| cos θ (“projection of **a** onto **b**”)
  - If **b** is a unit vector, then **<a**|**b> =** |**a**| cos θ
  - Inner product is a good measure of correlation
    - Two identical signals➔ parallel vectors ←➔perfectly correlated
      - <b|b> == 1 (b normalized)
    - …no common component ➔ orthogonal vectors ←~➔ uncorrelated
      - <b|c> == 0 (b and c orthogonal)
    - Used frequently in communication: correlate received signal with various possible transmitted signals; highest correlation wins
    - DSPs (and now micros) have special “multiply-accumulate” instructions for inner product / correlation



Another view of modulation & demodulation

Suppose we're (de)modulating just one bit (time 0 to T). Then to do low pass filter at end of demodulation operation, we can integrate over the whole bit period T (intuition: integration for all time gives DC [0 frequency] component…all higher frequencies contribute nothing to integral)

$$m(t) = b\cos(\omega t)$$

Modulation is multiplication by carrier

$$d = \int_0^T m(t)\cos(\omega t)dt$$

Demodulation is 2nd multiplication by carrier
Low pass filter implemented by integration from 0 to T

Now consider discrete-time:

$$\text{Let } c_t = \cos(\omega t)$$

$$m_t = bc_t$$

Modulation is multiplication by carrier

$$d = \sum_{t=0}^{T} m_t c_t$$

Demodulation is 2nd multiplication by carrier
Hey, that looks like an inner product

$$d = \sum_{t=0}^{T} bc_t c_t = b < c_t | c_t >$$

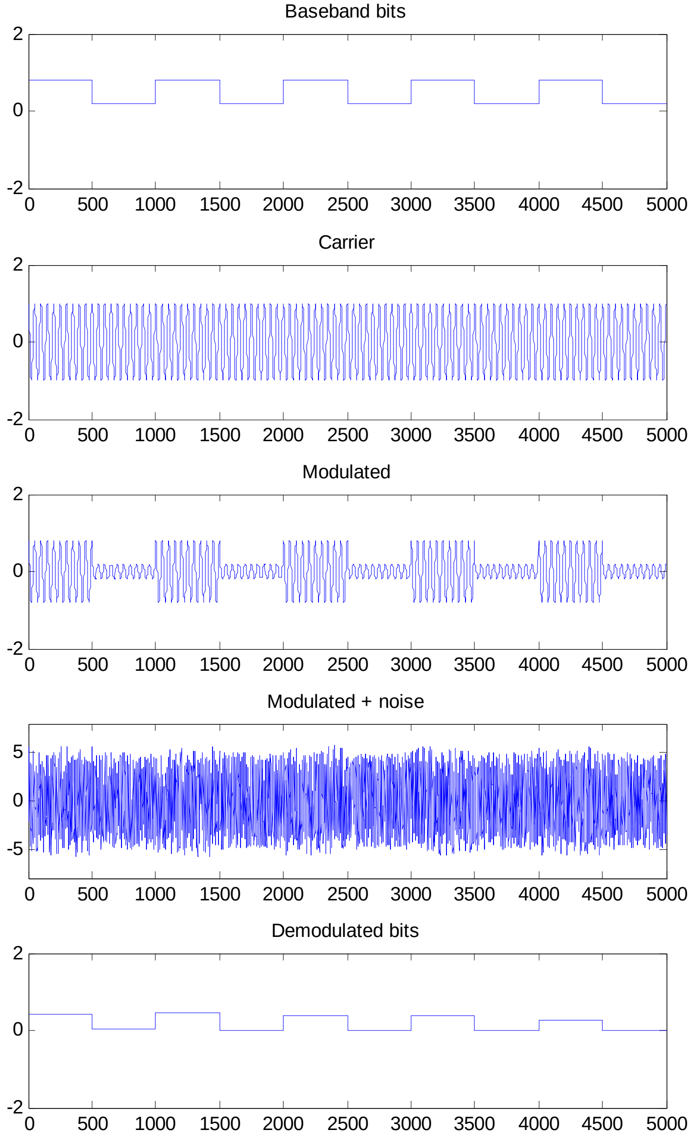
For c_t normalized ➔ $<c_t|c_t>=1$ ➔ d=b

Other observations

- Inner product concept applies in continuous case too…just that vectors are infinite dimensional. Instead of summing as last step of inner product, integrate
- Sines, cosines of different frequencies are orthogonal
  - They form a complete basis for “function space”
- Fourier transform is a change of basis
  - Time domain basis is delta fns (spikes): $f(t) = \int f(t)\delta(t)dt$
  - Project signal onto each frequency component (each basis vector for frequency domain) to get representation in Fourier basis
- Synchronous demodulation is computing one Fourier component
  - Rejects noise at all frequencies further from carrier than final low pass filter bandwidth

Synchronous demodulation example

Horizontal axes: Time



Signal during one bit period: b (a constant)

Carrier during one bit period: $c_t = \cos(\omega t)$

Modulated carrier $m_t = b\, c_t$

Signal apparently buried by noise

Signal + noise: $m_t + n_t = b\, c_t + 10*(\text{rand}-.5)$

Correct bits recovered (threshold this signal to get bits)

$$d = \frac{1}{500}\sum_{t=1}^{500}(m_t + n_t) \times c_t$$
$$d = \frac{1}{500}\sum_{t=1}^{500}(bc_t + n_t) \times c_t$$
$$d = \frac{1}{500}\sum_{t=1}^{500}(bc_t c_t + n_t c_t))$$

Envelope-following demodulation

Horizontal axes:
Time