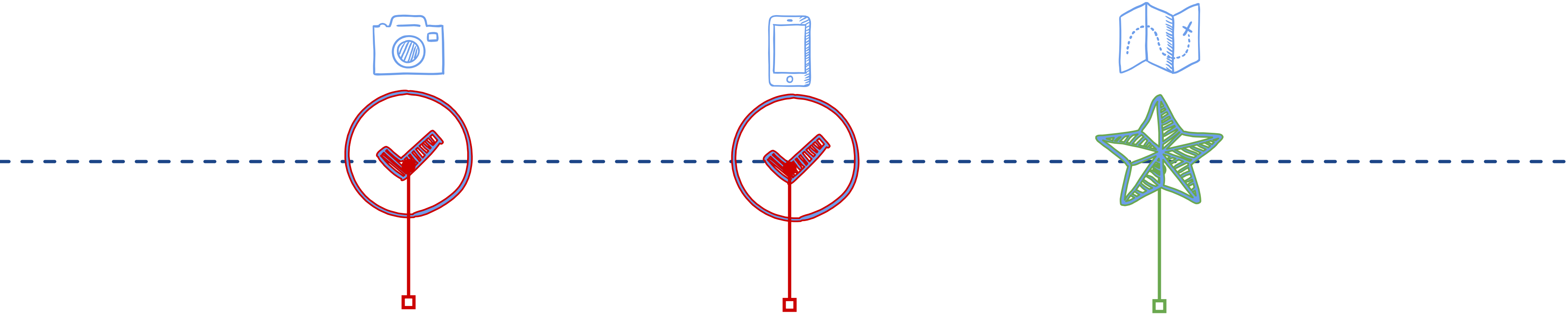
EECS16A
Acoustic Positioning System

Last Lab! :)

Insert names here

Where Are We Now?

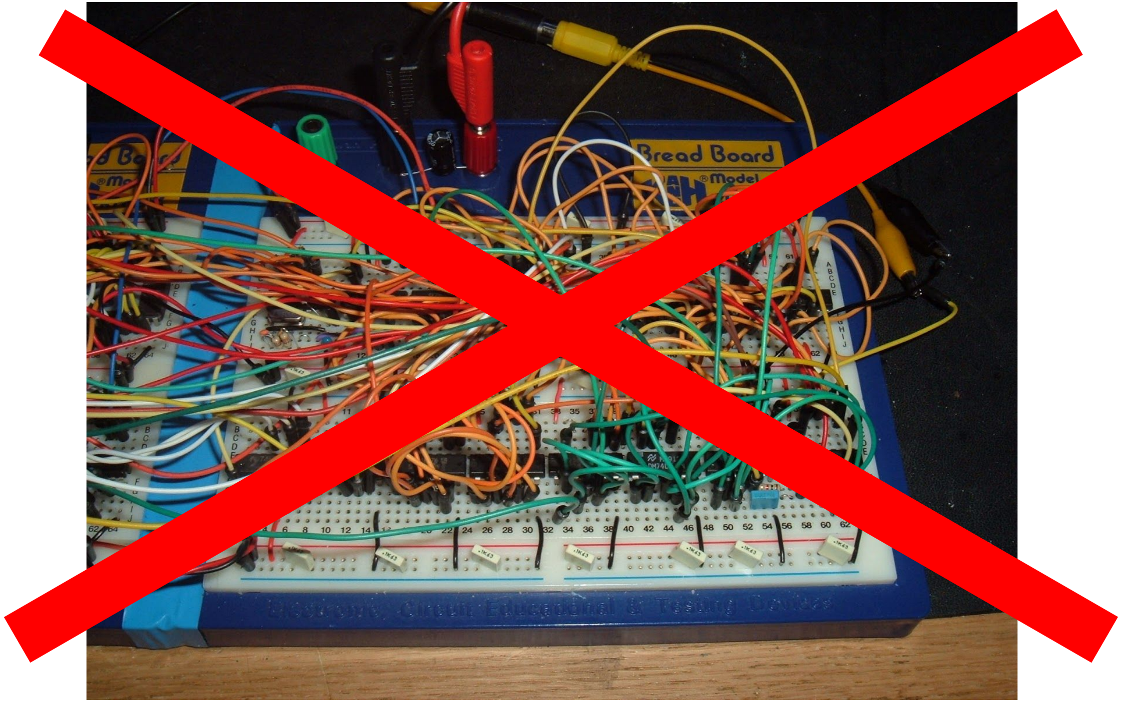


Announcements!

- This is the **last lab!**
- Course evaluations: https://course-evaluations.berkeley.edu/
- Good luck on the final!

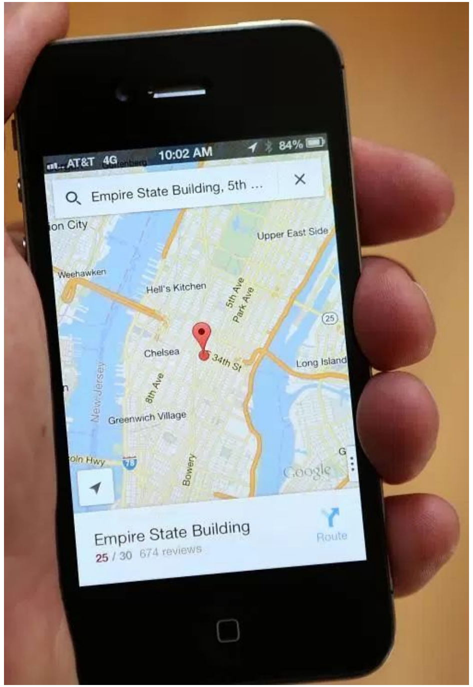
Announcements

- All software

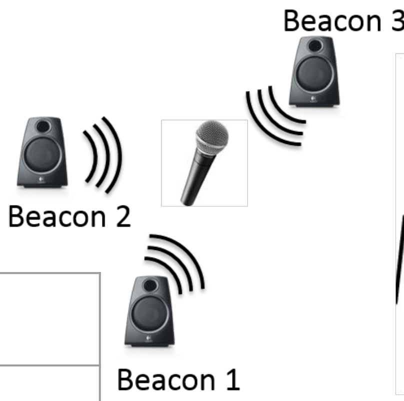


Today’s Lab: Acoustic Positioning System

- Global Positioning System (GPS)
  - Uses radio waves instead of sound waves
- Understand mathematical tools used for shifting and detecting signals
  - Think about cross correlation!



Set-up

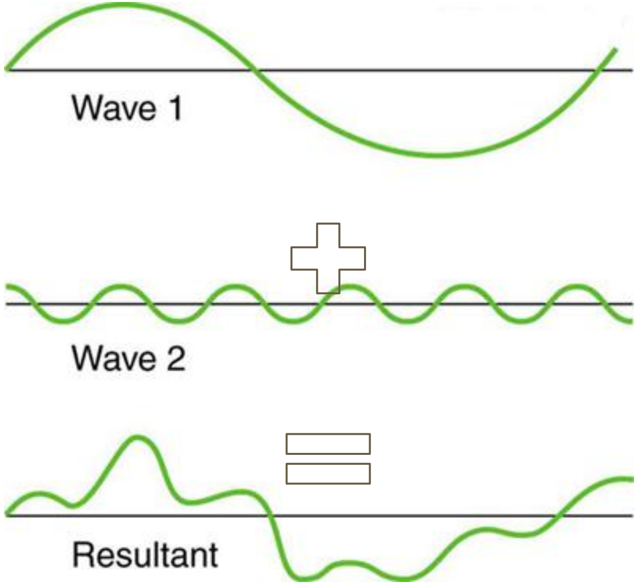


General	Lab Specific
receiver	microphone
Satellites repeatedly transmitting specific beacon signals	Speakers repeatedly playing specific tones (beacon signals)

- Known: Location of each satellite and what beacon signal each satellite is playing
- Unknown: Location of receiver ← what we want to figure out!

Set-up

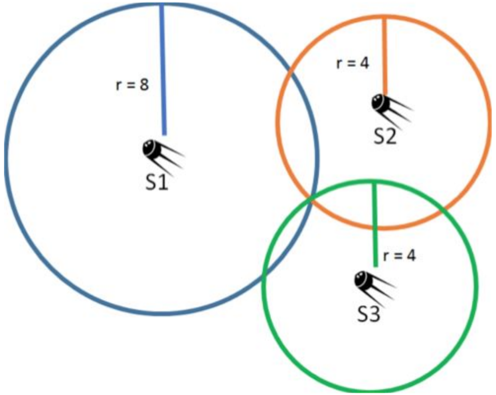
- Satellite:
  - Known, periodic waveforms
  - Know satellite location
- Receiver:
  - Will record the waveform
    - Sum of all shifted beacons
  - Waveform will be shifted from known satellite waveform based on how far it is from satellite (sound takes time to travel)



Let’s go backwards

Assume we know the **distance** between the receiver and every satellite

- Use **lateration** and the satellites’ locations to locate the receiver!
- How many satellites do we need in a 2D world?



How do we get those distances?

Assume we know the **time-delay** (in secs) of every beacon

- Use the **speed of sound** through air to get exactly how far our receiver is from every satellite
    - $d = v_s \cdot t$
    - $v_s \approx 343$ m/s

How do we get those time-delays?

Assume we know how many **samples** it takes for each beacon signal to arrive at the receiver

- Use the **sampling frequency** of receiver to get the **time-delay**
  - Sampling frequency [samples/sec] - rate at which microphone records samples

$$\frac{\text{samples}}{f_s} = \frac{\text{samples}}{\frac{samples}{second}} = \text{seconds}$$

Poll Time!

Let the sampling frequency be 1000 Hz and the speed of sound be 343 m/s. If I detect a signal with a delay of 100 samples, what is the distance between the speaker and the mic?

- 3430 m
- 34.3 m
- 343 m
- 3.43 m

Poll Time!

Let the sampling frequency be 1000 Hz and the speed of sound be 343 m/s. If I detect a signal with a delay of 100 samples, what is the distance between the speaker and the mic?

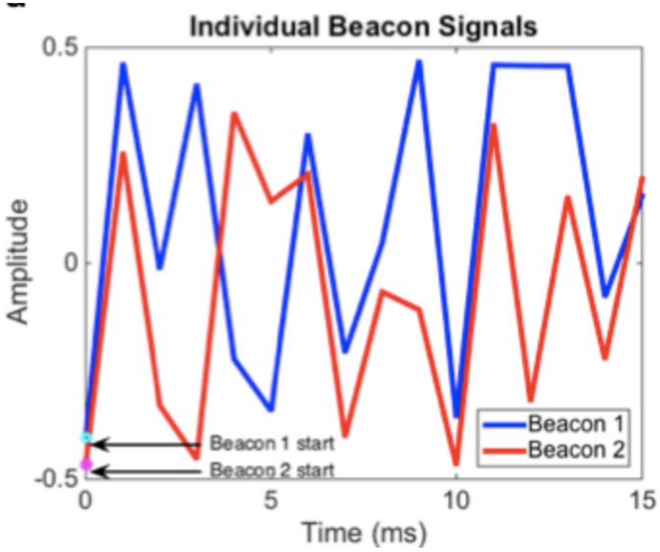
- 3430 m
- **34.3 m →**
- 343 m
- 3.43 m

$$\text{time delay} = \frac{\text{samples}}{\text{sampling frequency}} = \frac{100 \text{ samples}}{1000 \text{ Hz}} = 0.1 \text{ s}$$

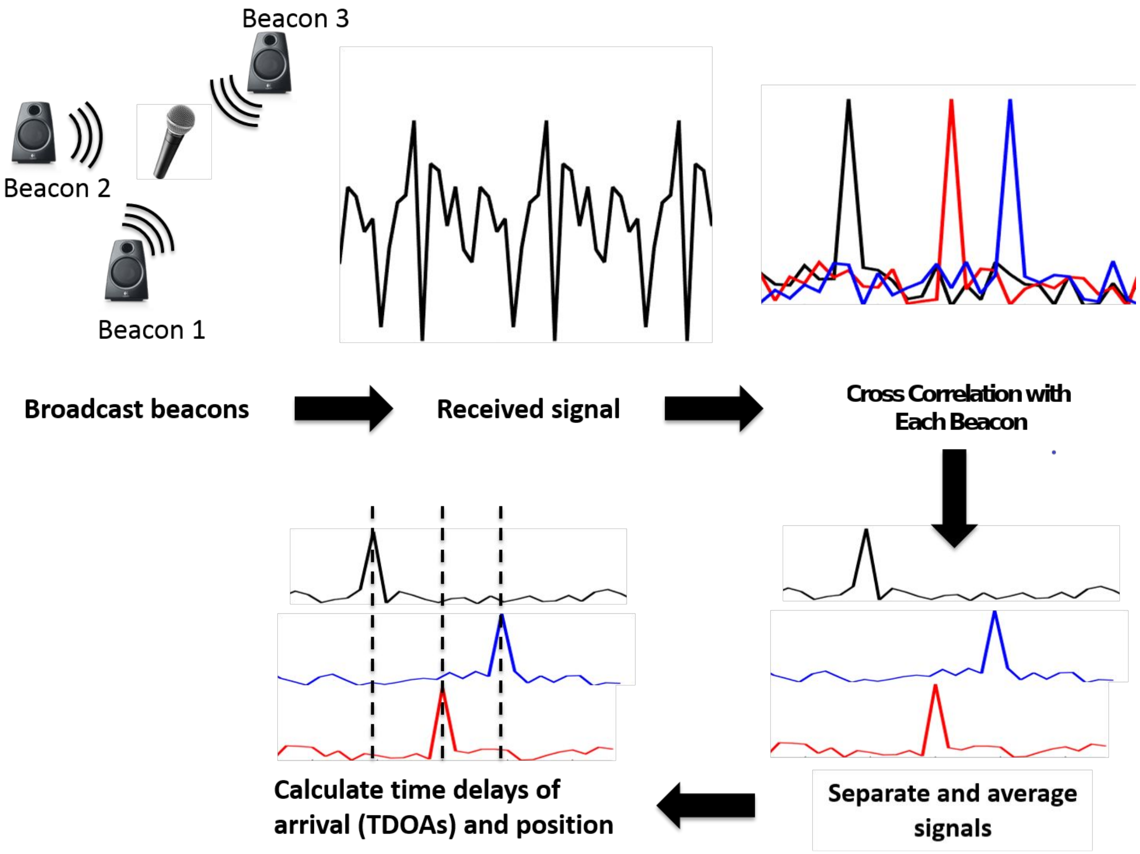
$$\text{d} = \text{v} \cdot \text{t} = 343 \text{ m/s} \cdot 0.1 \text{ s} = \boxed{34.3 \text{ m}}$$

How do we get sample delays?

- Receiver’s recorded signal is the sum of all the beacon signals
- Need to separate the recorded signal into the individual beacon signals to see how many samples each signal is delayed by



Overview


Beacon 3
Beacon 2
Beacon 1
Broadcast beacons
Received signal
Cross Correlation with Each Beacon
Separate and average signals
Calculate time delays of arrival (TDOAs) and position

Recall: Inner (Dot) product

- Computes how similar two vectors are

$$
\begin{aligned}
\langle \vec{x}, \vec{y} \rangle &\equiv \vec{x} \cdot \vec{y} \equiv \vec{x}^T \vec{y} \\
&= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \\
&= x_1 y_1 + x_2 y_2 + \cdots + x_n y_n \\
&= \sum_{i=1}^{n} x_i y_i
\end{aligned}
$$

Recall: Inner (Dot) product

$$\langle \vec{x}, \vec{y} \rangle = ||x|| \, ||y|| \cos \theta$$

An alternate form of the dot product

- **Given this expression, with ||x|| = ||y||, when is this expression maximized?**
  - θ = 0
  - vectors point in the same direction, so they are the same signal (SAME DIRECTION, SAME SIGNAL)

The bigger the dot product magnitude, the more "similar" the two vectors are

Tool: Cross-correlation

$$corr_r(B_A)[k] = \sum_{i=-\infty}^{\infty} r[i]B_A[i-k] \Leftrightarrow$$

In Python:

`cross_correlation(r, B`$_A$`)[k]`

- Mathematical tool for finding similarities between signals
- **Idea**: Computes dot product between r and signal B_A shifted by k samples

- From the previous slide, the peak of the cross-correlation vector tells us which shift amount makes B_A "most similar" to r

Poll Time!

Given $||x|| = ||y|| = 1$, when is the magnitude of the inner product expression maximized?

- theta = 0
- theta = 90
- theta = 180
- theta = -90

Poll Time!

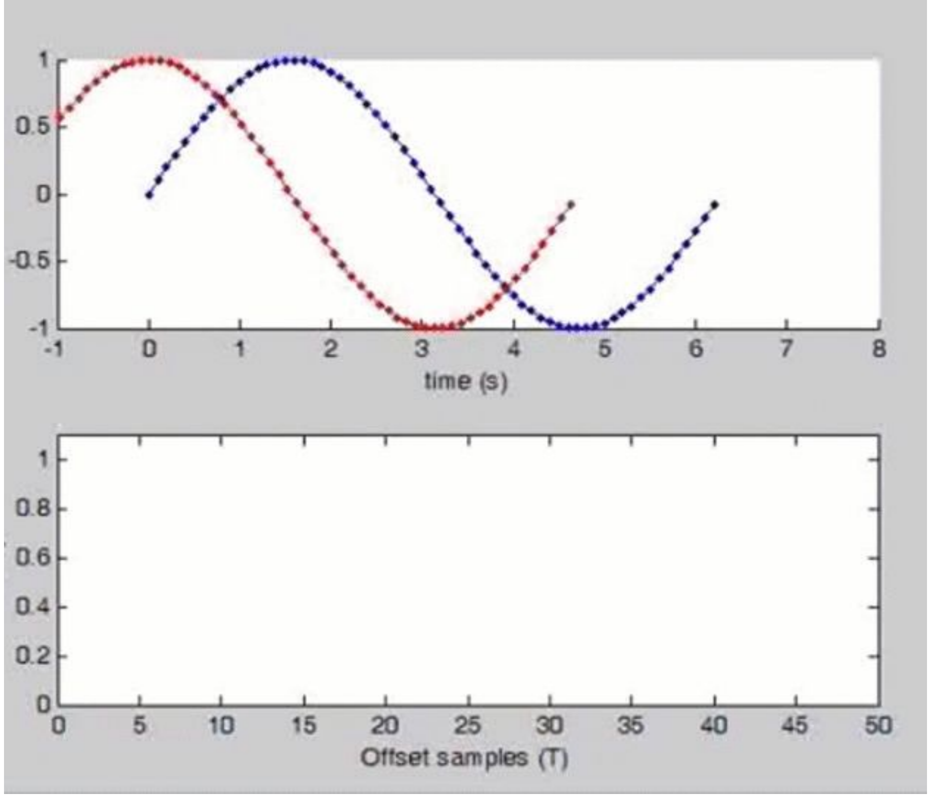
Given ||x|| = ||y|| = 1, when is the magnitude of the inner product expression maximized?

- **theta = 0**
- theta = 90
- **theta = 180**
- theta = -90

Tool: Cross-correlation

- **At ~ how many offset samples are the signals most similar?**

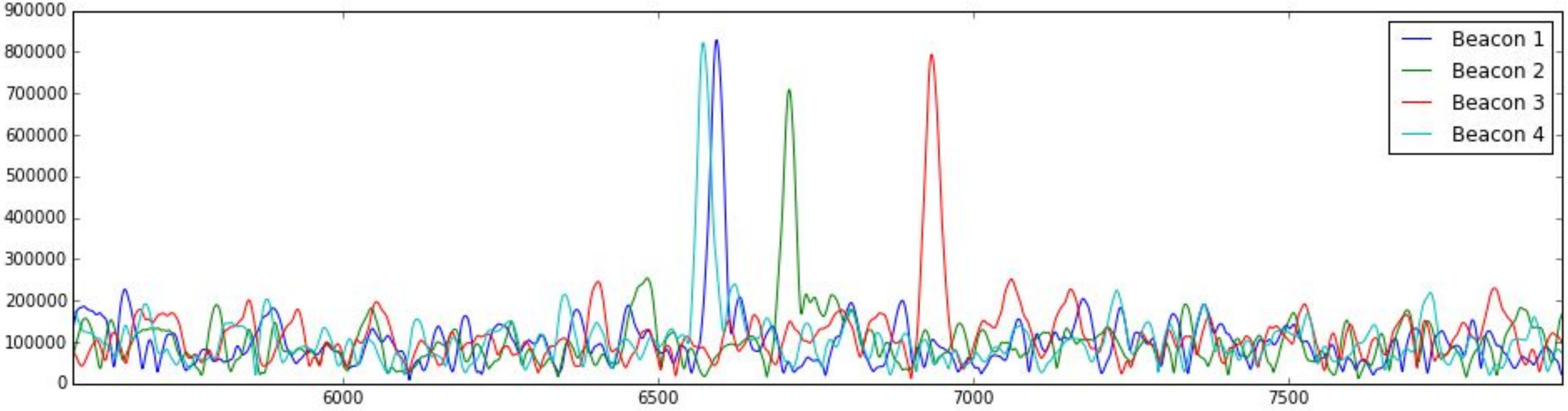
blue = r
red = B_A



Note: zero pad signals to match length

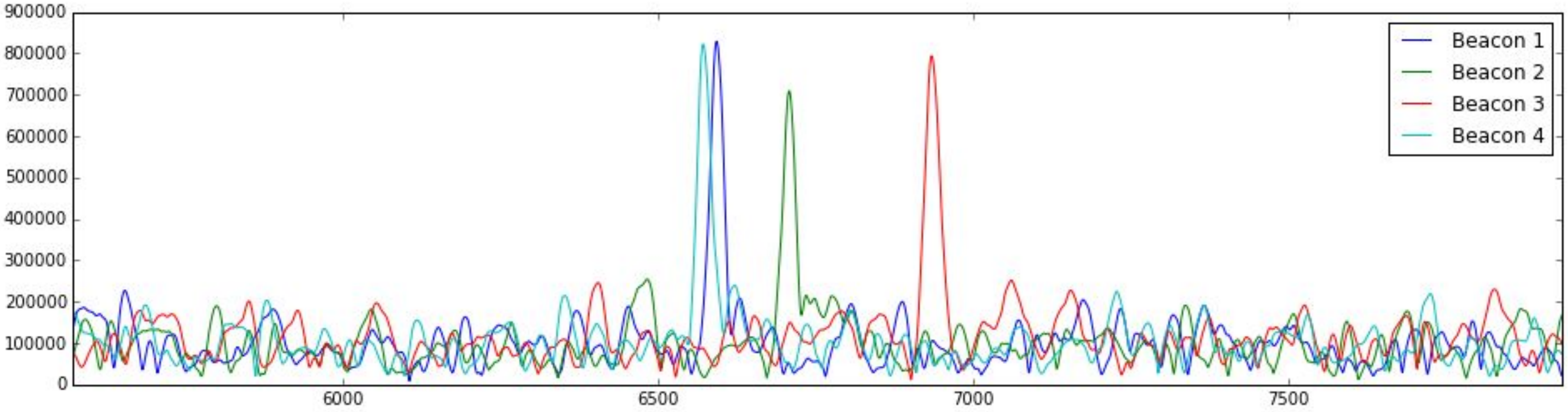
How to use?

- Cross correlating should tell us where each beacon signal arrived in our recorded signal
- Let's cross-correlate each of the known beacon signals with what we recorded and plot the result



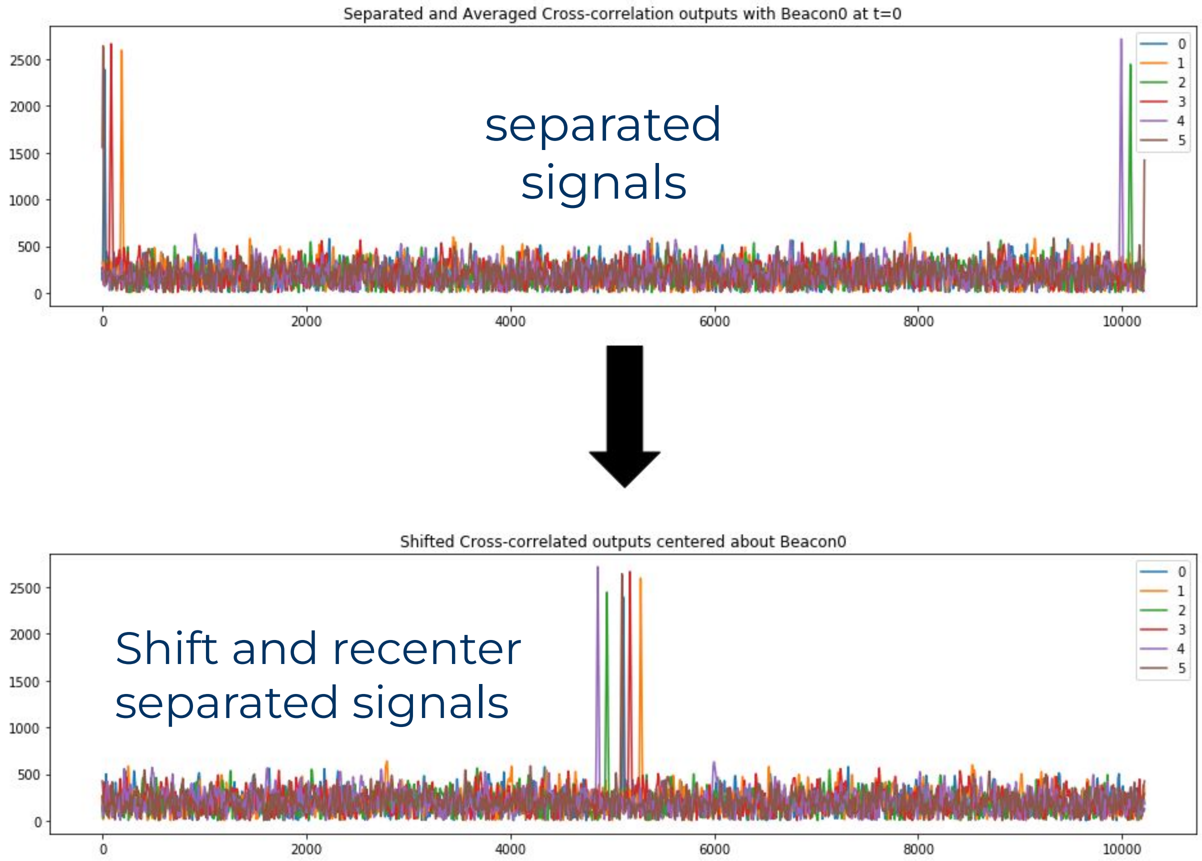
Absolute or relative sample delays?

- We can see peaks where each beacon signal arrived!
- But notice it only gives us **relative** sample delays
  - Still can't tell how many absolute samples each beacon is delayed, we don't know when it was supposed to arrive
- Arbitrarily pick a beacon to be the reference point



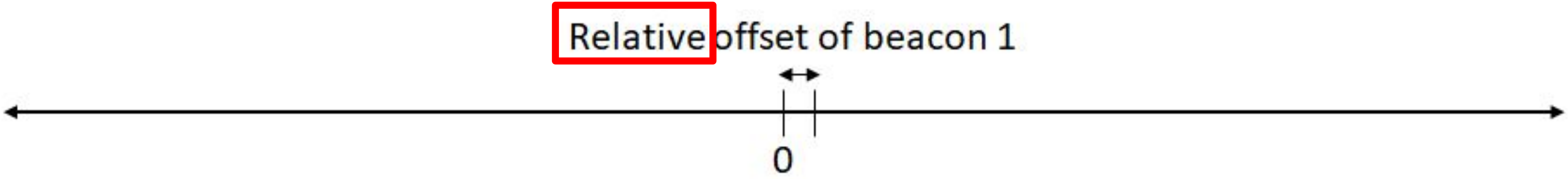
Absolute or relative sample delays?

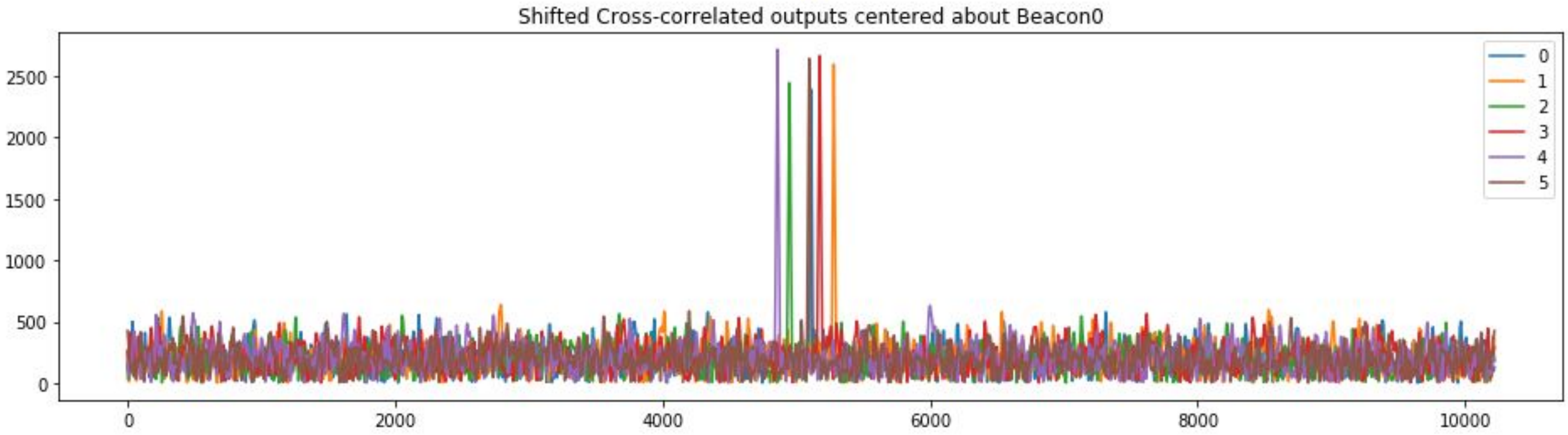
- Let's shift our axis so beacon 0 has a delay of 0
- We could pick any beacon to be the center
  - 0 is arbitrary



Absolute or relative sample delays?

Now beacon 0 is at our new "origin" and all computations are relative to the new "0" – but how do we find T0?

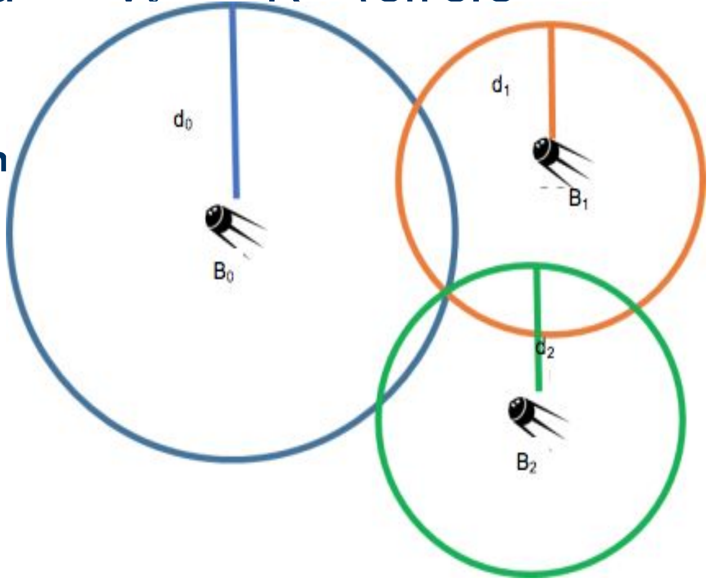




3 Beacon Example

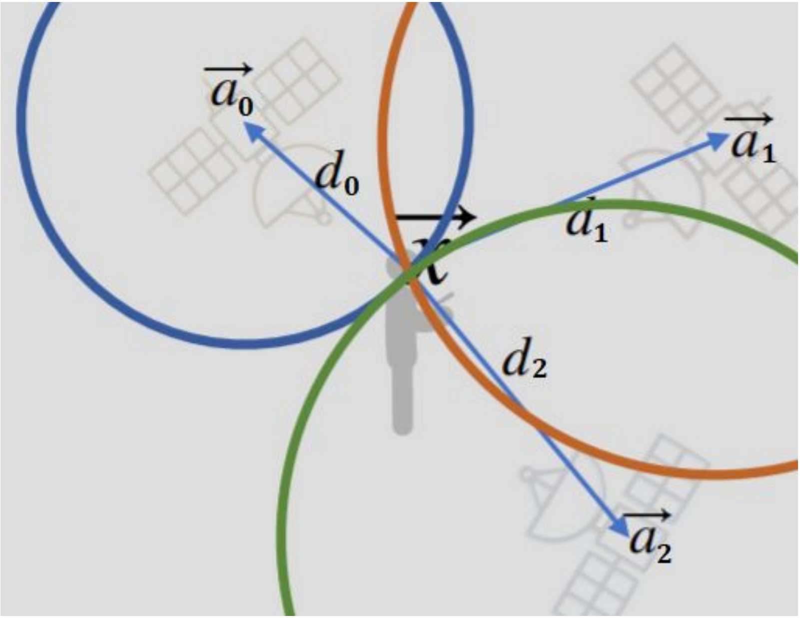
- To answer the T0 question, we must formally set up our system. Let beacon centers be: (x_0, y_0), (x_1, y_1) and (x_2, y_2)
- Time of arrivals: τ_0, τ_1, τ_2
- Distance of beacon m (m = 0, 1, 2) is $d \quad = v\tau \quad = R \quad$ (circle radii)

Circle equations: $(x - x_m)^2 + (y - y_m)^2 = d^2_m$



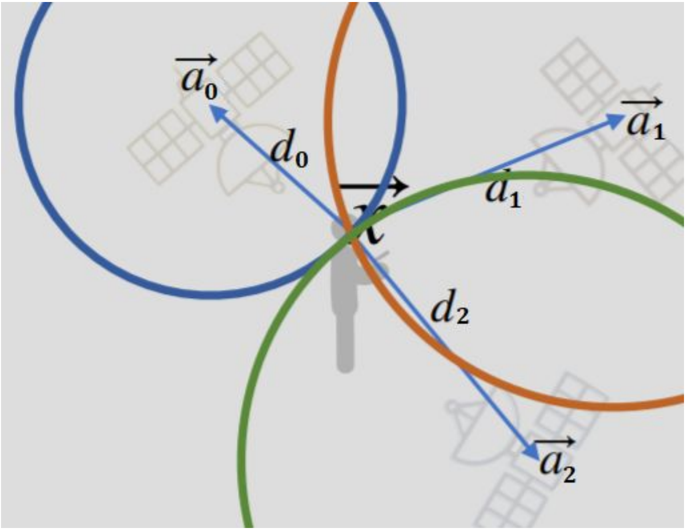
Trilateration

$$||\vec{r} - \vec{a_0}||^2 = d_0^2$$
$$||\vec{r} - \vec{a_1}||^2 = d_1^2$$
$$||\vec{r} - \vec{a_2}||^2 = d_2^2$$



$$d_i = v_s \tau_i$$

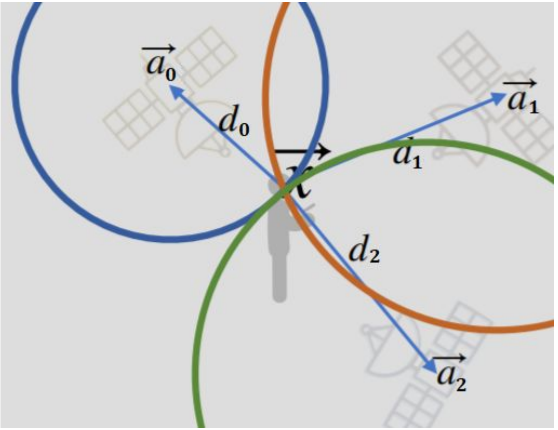
Trilateration



$$||\vec{r}||^2 - 2\vec{a_0}^T\vec{r} + ||\vec{a_0}||^2 = v_s^2\tau_0^2$$
$$||\vec{r}||^2 - 2\vec{a_1}^T\vec{r} + ||\vec{a_1}||^2 = v_s^2\tau_1^2$$
$$||\vec{r}||^2 - 2\vec{a_2}^T\vec{r} + ||\vec{a_2}||^2 = v_s^2\tau_2^2$$

Trilateration

$$||\vec{r}||^2 - 2\vec{a_0}^T\vec{r} + ||\vec{a_0}||^2 = v_s^2\tau_0^2$$
$$||\vec{r}||^2 - 2\vec{a_1}^T\vec{r} + ||\vec{a_1}||^2 = v_s^2\tau_1^2$$
$$||\vec{r}||^2 - 2\vec{a_2}^T\vec{r} + ||\vec{a_2}||^2 = v_s^2\tau_2^2$$



Subtracting the first equation yields:

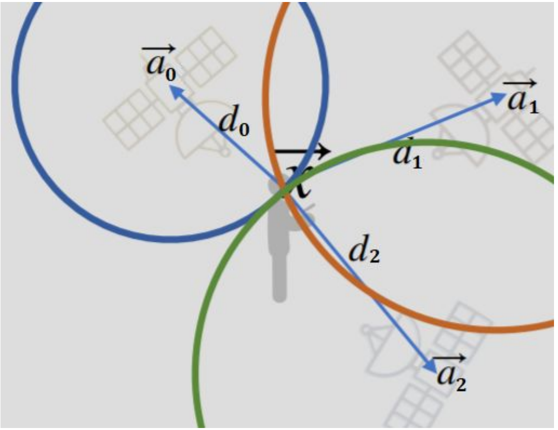
$$-2\vec{a_1}^T\vec{r} + 2\vec{a_0}^T\vec{r} + ||\vec{a_1}||^2 - ||\vec{a_0}||^2 = v_s^2(\tau_1^2 - \tau_0^2)$$
$$\implies 2(\vec{a_0} - \vec{a_1})^T\vec{r} = ||\vec{a_0}||^2 - ||\vec{a_1}||^2 + v_s^2(\tau_1^2 - \tau_0^2)$$

and,

$$2(\vec{a_0} - \vec{a_2})^T\vec{r} = ||\vec{a_0}||^2 - ||\vec{a_2}||^2 + v_s^2(\tau_2^2 - \tau_0^2)$$

Trilateration

$$2(\vec{a_0} - \vec{a_1})^T \vec{r} = ||\vec{a_0}||^2 - ||\vec{a_1}||^2 + v_s^2(\tau_1^2 - \tau_0^2)$$
$$2(\vec{a_0} - \vec{a_2})^T \vec{r} = ||\vec{a_0}||^2 - ||\vec{a_2}||^2 + v_s^2(\tau_2^2 - \tau_0^2)$$



We want to write this in terms of TDOAs and unknowns!

$$(\tau_i^2 - \tau_0^2) = (\tau_i - \tau_0)(\tau_i + \tau_0) = (\tau_i - \tau_0)(\tau_i - \tau_0 + 2\tau_0) = \Delta\tau_i(\Delta\tau_i + 2\tau_0)$$

$$\implies$$

$$2(\vec{a_0} - \vec{a_1})^T \vec{r} - 2(v_s^2 \Delta\tau_1)\tau_0 = ||\vec{a_0}||^2 - ||\vec{a_1}||^2 + v_s^2 \Delta\tau_1^2$$
$$2(\vec{a_0} - \vec{a_2})^T \vec{r} - 2(v_s^2 \Delta\tau_2)\tau_0 = ||\vec{a_0}||^2 - ||\vec{a_2}||^2 + v_s^2 \Delta\tau_2^2$$

Trilateration

We can expand our equations by writing our vectors in component form!

$$\vec{r} = \begin{bmatrix} r_x \\ r_y \end{bmatrix} \qquad \vec{a_i} = \begin{bmatrix} a_{i,x} \\ a_{i,y} \end{bmatrix} \qquad \vec{a_0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\implies$$

$$2a_{1,x}r_x + 2a_{1,y}r_y + 2v_s^2\Delta\tau_1\tau_0 = a_{1,x}^2 + a_{1,y}^2 - v_s^2\Delta\tau_1^2$$
$$2a_{2,x}r_x + 2a_{2,y}r_y + 2v_s^2\Delta\tau_2\tau_0 = a_{2,x}^2 + a_{2,y}^2 - v_s^2\Delta\tau_2^2$$

Trilateration

$$2a_{1,x}r_x + 2a_{1,y}r_y + 2v_s^2\Delta\tau_1\tau_0 = a_{1,x}^2 + a_{1,y}^2 - v_s^2\Delta\tau_1^2$$
$$2a_{2,x}r_x + 2a_{2,y}r_y + 2v_s^2\Delta\tau_2\tau_0 = a_{2,x}^2 + a_{2,y}^2 - v_s^2\Delta\tau_2^2$$

What are our unknowns in this system?

Trilateration

$$2a_{1,x}r_x + 2a_{1,y}r_y + 2v_s^2\Delta\tau_1\tau_0 = a_{1,x}^2 + a_{1,y}^2 - v_s^2\Delta\tau_1^2$$
$$2a_{2,x}r_x + 2a_{2,y}r_y + 2v_s^2\Delta\tau_2\tau_0 = a_{2,x}^2 + a_{2,y}^2 - v_s^2\Delta\tau_2^2$$

What are our unknowns in this system?

$$r_x, r_y, \tau_0$$

Trilateration

$$2a_{1,x}r_x + 2a_{1,y}r_y + 2v_s^2\Delta\tau_1\tau_0 = a_{1,x}^2 + a_{1,y}^2 - v_s^2\Delta\tau_1^2$$
$$2a_{2,x}r_x + 2a_{2,y}r_y + 2v_s^2\Delta\tau_2\tau_0 = a_{2,x}^2 + a_{2,y}^2 - v_s^2\Delta\tau_2^2$$

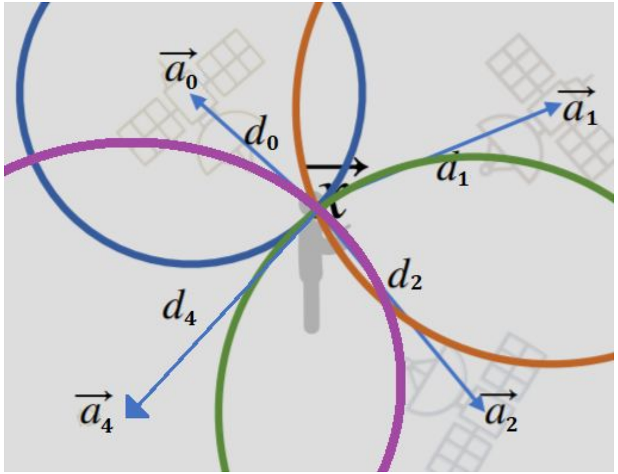
What are our unknowns in this system?

$$r_x, r_y, \tau_0$$

Problem: 3 unknowns and 2 equations!

Solution: add another beacon to produce a third equation!

Trilateration



3 equations and 3 unknowns, so we have a solvable system!

$$2a_{1,x}r_x + 2a_{1,y}r_y + 2v_s^2\Delta\tau_1\tau_0 = a_{1,x}^2 + a_{1,y}^2 - v_s^2\Delta\tau_1^2$$
$$2a_{2,x}r_x + 2a_{2,y}r_y + 2v_s^2\Delta\tau_2\tau_0 = a_{2,x}^2 + a_{2,y}^2 - v_s^2\Delta\tau_2^2$$
$$2a_{3,x}r_x + 2a_{3,y}r_y + 2v_s^2\Delta\tau_3\tau_0 = a_{3,x}^2 + a_{3,y}^2 - v_s^2\Delta\tau_3^2$$

Multilateration

We can produce overdetermined system with M beacons!

$$2\begin{bmatrix} a_{1,x} & a_{1,y} & v_s^2\Delta\tau_1 \\ a_{2,x} & a_{2,y} & v_s^2\Delta\tau_2 \\ & \vdots & \\ a_{M-1,x} & a_{M-1,y} & v_s^2\Delta\tau_{M-1} \end{bmatrix}\begin{bmatrix} r_x \\ r_y \\ \tau_0 \end{bmatrix} = \begin{bmatrix} a_{1,x}^2 + a_{1,y}^2 - v_s^2\Delta\tau_1^2 \\ a_{2,x}^2 + a_{2,y}^2 - v_s^2\Delta\tau_2^2 \\ \vdots \\ a_{M-1,x}^2 + a_{M-1,y}^2 - v_s^2\Delta\tau_{M-1}^2 \end{bmatrix}$$

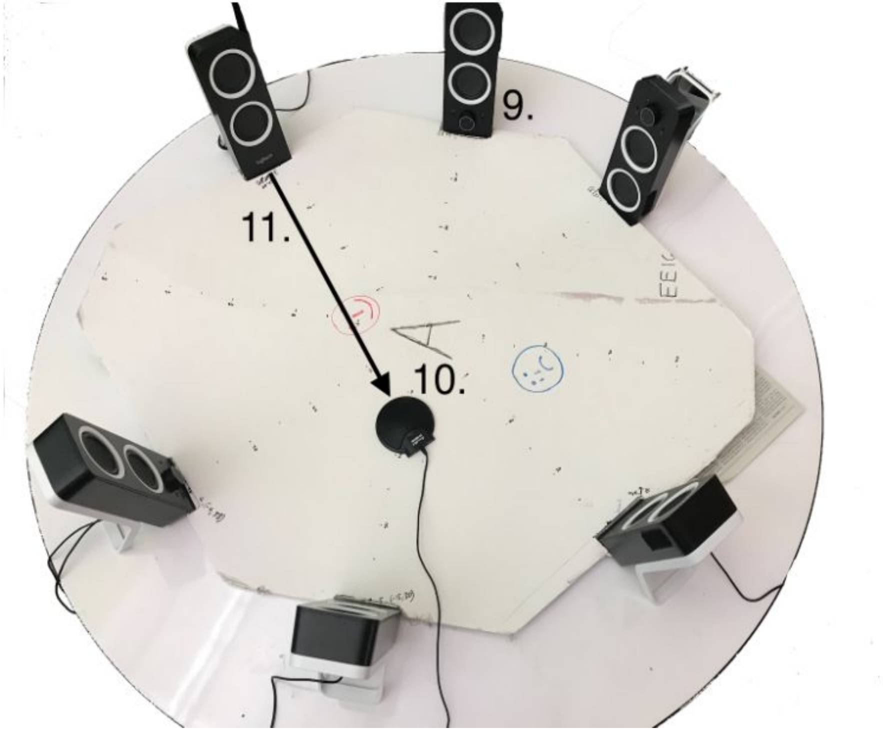
“Solving” an Overdetermined System

- After simplifying, we have more equations than unknowns (x,y)
- Can do least-squares regardless of number of beacons
- Best estimate of location if measurements are inconsistent
- If there is no exact point of intersection because of error or noise

$$Ax = b$$

$$A^T A x = A^T b$$

Setup Looks Like:



Important Notes

FOR TODAY, FOLLOW THESE INSTRUCTIONS

- Download and extract the zip file
- **Use “launch_notebook.bat” or open Anaconda Powershell Prompt and run “jupyter notebook” – do not just use standard command line**
- Navigate to the lab folder and proceed as usual

- Read over the math carefully, we’ll be asking you about it!
- Stay safe and good luck with the rest of the semester!