Key Question: How do we "learn" models from data, and make predictions?

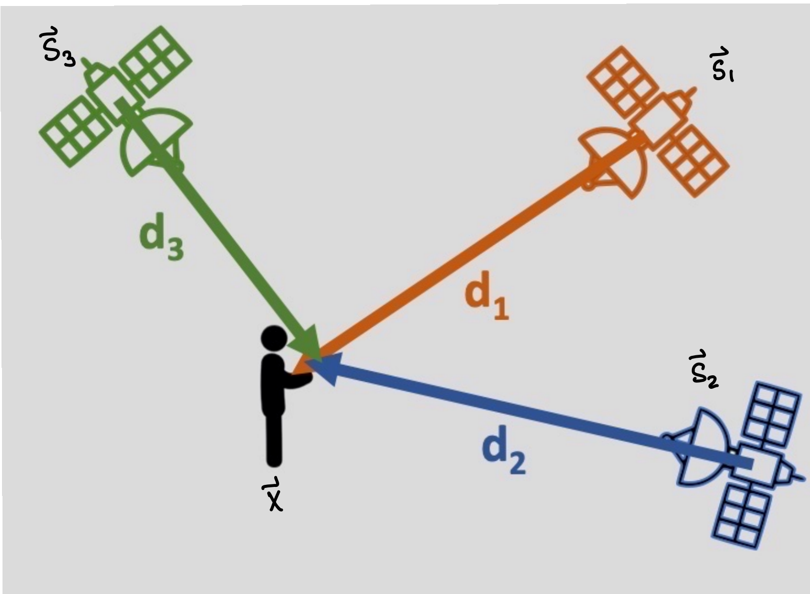
Agenda

- Trilateration
- Signals
- Classification of Satellites
- Cross-Correlation

Trilateration

- Finding position from distances
  - Known: positions of satellites, distances to satellites
  - Unknown: your position

This is the last step in our GPS system! It is also known as triangulation.



Let $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\vec{s}_1 = \begin{bmatrix} s_{1x} \\ s_{1y} \end{bmatrix}$, $\vec{s}_2 = \begin{bmatrix} s_{2x} \\ s_{2y} \end{bmatrix}$, $\vec{s}_3 = \begin{bmatrix} s_{3x} \\ s_{3y} \end{bmatrix}$ represent positions.

How do we solve for unknown values? Write out a system of equations!

$\|\vec{x} - \vec{s}_1\|^2 = d_1^2$ ① $\quad \|\vec{x}\| = \sqrt{\langle \vec{x}, \vec{x} \rangle}$, so let's square both sides to simplify the math!

$\|\vec{x} - \vec{s}_2\|^2 = d_2^2$ ②

$\|\vec{x} - \vec{s}_3\|^2 = d_3^2$ ③

Expanding these equations:

$$① \quad (\vec{x}-\vec{s}_1)^T(\vec{x}-\vec{s}_1) = d_1^2$$

$$\vec{x}^T\vec{x} - \vec{s}_1^T\vec{x} - \vec{x}^T\vec{s}_1 + \vec{s}_1^T\vec{s}_1 = d_1^2$$

$$\|\vec{x}\|^2 - 2\vec{s}_1^T\vec{x} + \|\vec{s}_1\|^2 = d_1^2$$

$$② \quad \|\vec{x}\|^2 - 2\vec{s}_2^T\vec{x} + \|\vec{s}_2\|^2 = d_2^2$$

$$③ \quad \|\vec{x}\|^2 - 2\vec{s}_3^T\vec{x} + \|\vec{s}_3\|^2 = d_3^2$$

We have three equations and two unknowns, but we have a problem! These equations are <u>not</u> linear! We have $\|\vec{x}\|^2$ terms.

What do we do? A trick! What if we just... got rid of those terms?

$$② - ① \quad \cancel{\|\vec{x}\|^2} - 2\vec{s}_2^T\vec{x} + \|\vec{s}_2\|^2 - \cancel{\|\vec{x}\|^2} + 2\vec{s}_1^T\vec{x} - \|\vec{s}_1\|^2 = d_2^2 - d_1^2$$

$$③ - ① \quad \cancel{\|\vec{x}\|^2} - 2\vec{s}_3^T\vec{x} + \|\vec{s}_3\|^2 - \cancel{\|\vec{x}\|^2} + 2\vec{s}_1^T\vec{x} - \|\vec{s}_1\|^2 = d_3^2 - d_1^2$$

Rearranging these equations:

$$2(\vec{s}_1-\vec{s}_2)^T\vec{x} = \|\vec{s}_1\|^2 - \|\vec{s}_2\|^2 + d_2^2 - d_1^2$$

$$2(\vec{s}_1-\vec{s}_3)^T\vec{x} = \|\vec{s}_1\|^2 - \|\vec{s}_3\|^2 + d_3^2 - d_1^2$$

$$(\vec{s}_1-\vec{s}_2)^T = \left(\begin{bmatrix} s_{11} \\ s_{12} \end{bmatrix} - \begin{bmatrix} s_{21} \\ s_{22} \end{bmatrix}\right)^T = \begin{bmatrix} s_{11}-s_{21} & s_{12}-s_{22} \end{bmatrix}$$

These are linear!!!

$$2\begin{bmatrix} s_{11}-s_{21} & s_{12}-s_{22} \\ s_{11}-s_{31} & s_{12}-s_{32} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \|\vec{s}_1\|^2 - \|\vec{s}_2\|^2 + d_2^2 - d_1^2 \\ \|\vec{s}_1\|^2 - \|\vec{s}_3\|^2 + d_3^2 - d_1^2 \end{bmatrix}$$

Solve using Gaussian Elimination!

Yay! Now, we just have to find out how we even get those distances in the first place.

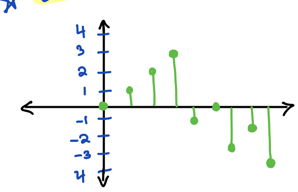
How do we find out how far we are from the satellites?

We need some form of communication to get information from the satellites.

Signals

- A message that contains information as a function of time.
- Two types:

★ Discrete Time Signals (DTS) are defined at specific points in time.

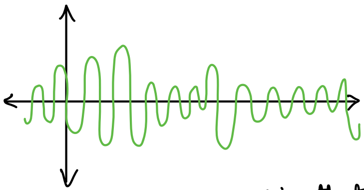


$$\vec{s} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ -1 \\ 0 \\ -3 \\ -2 \\ -4 \end{bmatrix}$$

$s[k]$ is the k^{th} element of $\vec{s}$.

We can represent these signals as vectors, where each element is the value at a single instance in time. For example, $s[0] = 0$ and $s[4] = -1$ above

- Continuous Time Signals (CTS) are defined over all time.

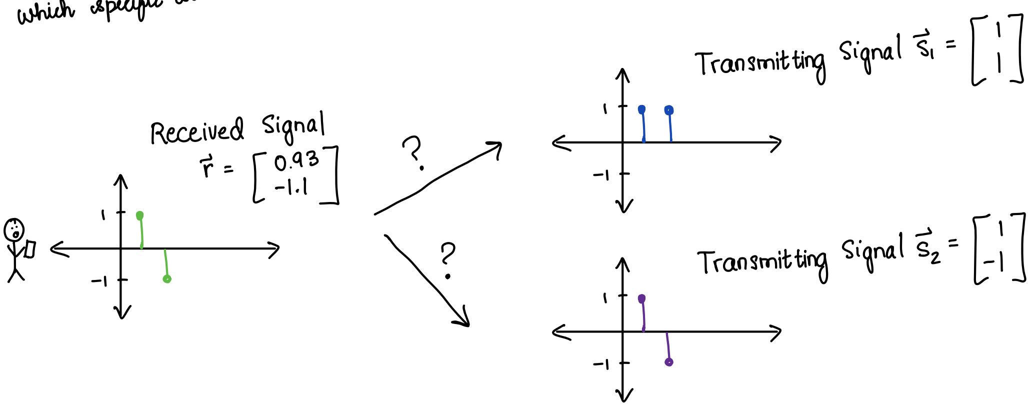


These are usually functions that can be sampled at regular intervals to generate a discrete time signal.

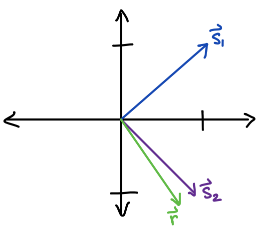
Classification of Satellites

Our satellites are continuously broadcasting unique signals, or messages, and to receive and digitize these messages, we need some kind of receiver, like a phone.

For simplicity, let's say we can only receive one signal at a time. How do we know which satellite sent this signal to us? We care about which specific satellite it was, because we need to know the location of the beacon, or satellite. We know each satellite has a unique broadcast.



One idea: let's find the distance between each of the transmitted signal vectors and the received signal, or the length of the error vector.



We want to find:

$$i^* = \underset{i \in \{1,2\}}{\operatorname{argmin}} \left\| \vec{r} - \vec{s}_i \right\|$$

↑ which vector, $\vec{s}_1$ or $\vec{s}_2$, minimizes the length of the difference?

Let's expand this:

$$i^* = \underset{i \in \{1,2\}}{\operatorname{argmin}} \left\| \vec{r} - \vec{s}_i \right\|^2$$

↙ doesn't change argmin value (even if it changes min value)

$$\begin{aligned}
\left\| \vec{r} - \vec{s}_i \right\|^2 &= \langle \vec{r} - \vec{s}_i , \vec{r} - \vec{s}_i \rangle \\
&= \langle \vec{r}, \vec{r} - \vec{s}_i \rangle - \langle \vec{s}_i, \vec{r} - \vec{s}_i \rangle \\
&= \langle \vec{r}, \vec{r} \rangle - \langle \vec{r}, \vec{s}_i \rangle - \langle \vec{s}_i, \vec{r} \rangle + \langle \vec{s}_i, \vec{s}_i \rangle \\
&= \underbrace{\left\| \vec{r} \right\|^2 + \left\| \vec{s}_i \right\|^2}_{\text{fixed values do not affect argmin}} - 2 \underbrace{\langle \vec{r}, \vec{s}_i \rangle}_{\text{This term is important}}
\end{aligned}$$

because $\vec{s}_i$'s all have the same length.

(scaling of 2 will not impact the argmin)

If $\langle \vec{r}, \vec{s}_i \rangle$ is maximized, then $\left\| \vec{r} - \vec{s}_i \right\|^2$ is minimized.

This is why we talked about inner products yesterday! Given two signals, we can figure out how similar they are!

To summarize, our procedure to classify the received signal was

for $i \in \{1, 2\}$:

  compute $\langle \vec{r}, \vec{s}_i \rangle$

return i for which $\langle \vec{r}, \vec{s}_i \rangle$ was the maximum.

In this case,

$$\langle \vec{r}, \vec{s}_1 \rangle = 0.93 \cdot 1 + (-1.1) \cdot 1 = -0.17$$

$$\langle \vec{r}, \vec{s}_2 \rangle = 0.93 \cdot 1 + (-1.1) \cdot (-1) = 2.03 \leftarrow$$ This is bigger! So, $i^* = 2$, and our signal was transmitted from $\vec{s}_2$.

Realistically, we will probably have more than one transmitted signal in our received signal, as well as some noise, causing interference.
Consider the case of just two satellites, which can be generalized.

Possibilities

① No satellite signals are in our received signal.
$\vec{r} = \vec{n}$

② Only $\vec{s}_1$ is in our received signal
$\vec{r} = \vec{s}_1 + \vec{n}$

③ Only $\vec{s}_2$ is in our received signal
$\vec{r} = \vec{s}_2 + \vec{n}$

④ Both $\vec{s}_1$ and $\vec{s}_2$ are in our received signal.
$\vec{r} = \vec{s}_1 + \vec{s}_2 + \vec{n}$.

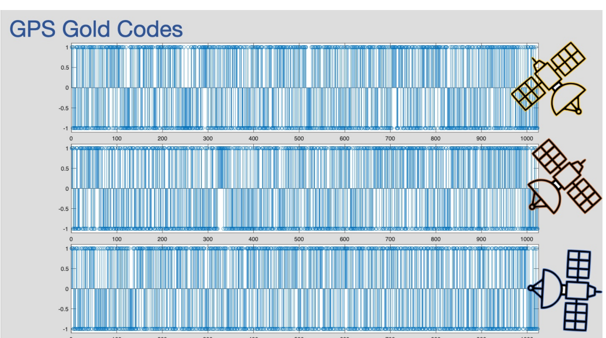
If both $\vec{s}_1$ and $\vec{s}_2$ are in $\vec{r}$, then our inner product for our similarity measure will be:

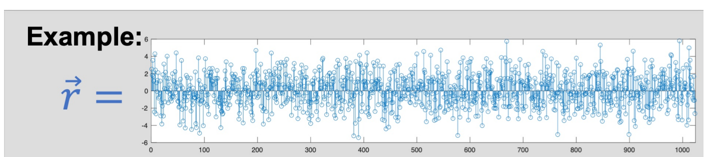
$$\langle \vec{r}, \vec{s}_1 \rangle = \langle \vec{s}_1 + \vec{s}_2 + \vec{n}, \vec{s}_1 \rangle$$

$$= \underbrace{\langle \vec{s}_1, \vec{s}_1 \rangle}_{\text{Desired}} + \underbrace{\langle \vec{s}_2, \vec{s}_1 \rangle}_{\text{Interference}} + \underbrace{\langle \vec{n}, \vec{s}_1 \rangle}_{\text{Small}}$$

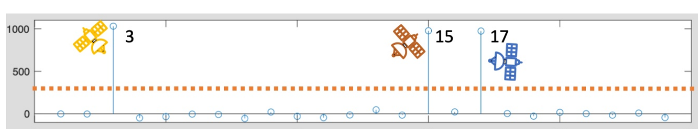
How do we design signals that don't interfere?
Make them orthogonal! $\langle \vec{s}_i, \vec{s}_j \rangle = 0$ for $i \neq j$.

The unique signals that satellites broadcast are called gold codes.





$\langle \vec{r}, \vec{s}_i \rangle = \quad \vec{r}^T \vec{s}_1 \quad \vec{r}^T \vec{s}_2 \dots \vec{r}^T \vec{s}_{24}$



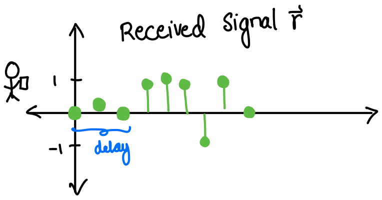
Satellite 3, 15, and 17 are present in the signal

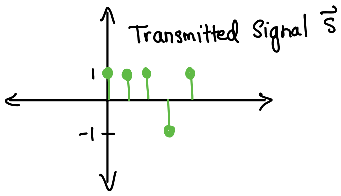
Now, I know which satellites are nearby and in my received signal.

What is my distance from each of them?

I know that each satellite transmits a unique code, and that signal is then received and digitized by a receiver. But, the satellites orbit the Earth at an altitude of approximately 13,000 miles (20,000km), so it takes some time for the signal to reach me. I also know that $d = vt$, where d is the distance covered by something traveling at v velocity for t amount of time. If the velocity of radio waves (or light or sound) is known, I need to find out how long it took the signal to reach me to find my distance from the satellite!

For simplicity, let's assume we are receiving a signal from one satellite which transmits a signal once, and then stops.





Idea: We can pattern match the signals

But, the problem is, these signals are not the same length! $\vec{r} \in \mathbb{R}^9$ but $\vec{s} \in \mathbb{R}^5$

Let's define infinite signals by zero-padding these signals!

$$\vec{r} \in \mathbb{R}^9 \longrightarrow r[n] = \begin{cases} r_n & 0 \le n \le 8 \\ 0 & \text{elsewhere} \end{cases}$$

$$\vec{s} \in \mathbb{R}^5 \longrightarrow s[n] = \begin{cases} s_n & 0 \le n \le 4 \\ 0 & \text{elsewhere} \end{cases}$$

We want to find the time delay between $\vec{s}$ and $\vec{r}$. We already used inner products to find how similar two vectors or signals are. What if we checked how similar one signal was to the other at different time shifts?

Cross-Correlation

- A measure of the similarity between two signals $\vec{x}$ and $\vec{y}$ based on the inner product

$$\langle r[n], s[n] \rangle = \sum_{-\infty}^{\infty} r[n]s[n]$$

$$\langle r[n], s[n-1] \rangle = \sum_{-\infty}^{\infty} r[n]\, s[n-1] \quad \text{(shift s to the right by 1)}$$

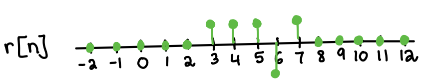
$$\vdots$$

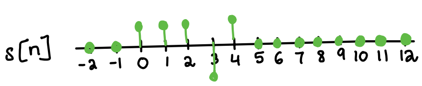
In general,

$$\text{corr}_{\vec{x}}(\vec{y})[k] = \sum_{-\infty}^{\infty} x[n]y[n-k] = \langle x[n], y[n-k] \rangle$$

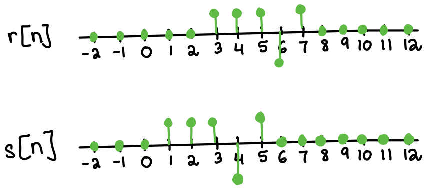
Let's find the cross-correlation between $\vec{r}$ and $\vec{s}$ in the example above! We want to find the cross-correlation at every shift where the signals overlap.

$$\text{corr}_{\vec{r}}(\vec{s})[0] = \sum_{-\infty}^{\infty} r[n]\, s[n] = (0\cdot1)+(0\cdot1)+(0\cdot1)+(1\cdot -1)+(1\cdot1)+(1\cdot0)+(-1\cdot0)+(1\cdot0) = 0$$

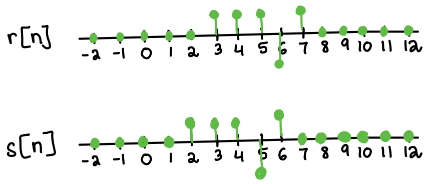




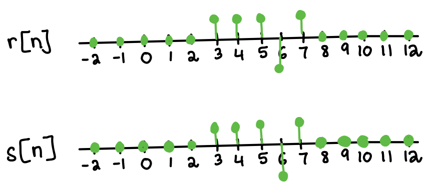
$$\mathrm{corr}_{\vec{r}}\,\vec{s}[1] = \sum_{-\infty}^{\infty} r[n]\, s[n-1] = (0\cdot1)+(0\cdot1)+(1\cdot1)+(1\cdot-1)+(1\cdot1)+(-1\cdot0)+(1\cdot0) = 1$$



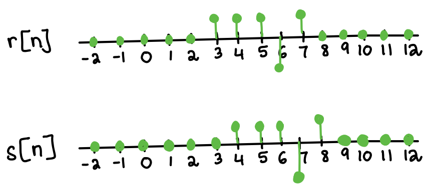
$$\mathrm{corr}_{\vec{r}}\,\vec{s}[2] = \sum_{-\infty}^{\infty} r[n]\, s[n-2] = (0\cdot1)+(1\cdot1)+(1\cdot1)+(1\cdot-1)+(-1\cdot1)+(1\cdot0) = 0$$



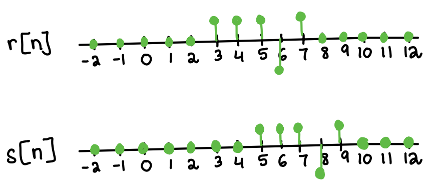
$$\mathrm{corr}_{\vec{r}}\,\vec{s}[3] = \sum_{-\infty}^{\infty} r[n]\, s[n-3] = (1\cdot1)+(1\cdot1)+(1\cdot1)+(-1\cdot-1)+(1\cdot1) = 5$$



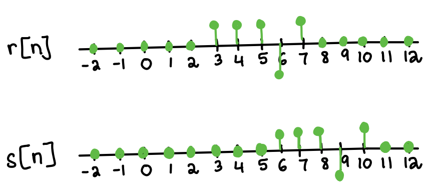
$$\mathrm{corr}_{\vec{r}}\,\vec{s}[4] = \sum_{-\infty}^{\infty} r[n]\, s[n-4] = (1\cdot0)+(1\cdot1)+(1\cdot1)+(-1\cdot1)+(1\cdot-1)+(0\cdot1) = 0$$



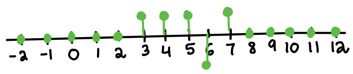
$$\mathrm{corr}_{\vec{r}}\,\vec{s}[5] = \sum_{-\infty}^{\infty} r[n]\, s[n-5] = (1\cdot0)+(1\cdot0)+(1\cdot1)+(-1\cdot1)+(1\cdot1)+(0\cdot-1)+(0\cdot1) = 1$$

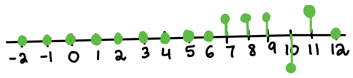


$$\mathrm{corr}_{\vec{r}}\,\vec{s}[6] = \sum_{-\infty}^{\infty} r[n]\, s[n-6] = (1\cdot0)+(1\cdot0)+(1\cdot0)+(-1\cdot1)+(1\cdot1)+(0\cdot1)+(0\cdot-1)+(0\cdot1) = 0$$



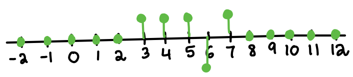
$$\text{corr}_{\vec{r}}\, \vec{s}[7] = \sum_{-\infty}^{\infty} r[n]\, s[n-7] = (1\cdot 0)+(1\cdot 0)+(1\cdot 0)+(-1\cdot 0)+(1\cdot 1)+(0\cdot 1)+(0\cdot 1)+(0\cdot -1)+(0\cdot 1) = 1$$

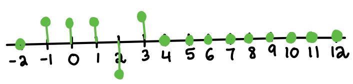
r[n] 

s[n] 

We have one more possible shift where the two signals overlap! A left shift!

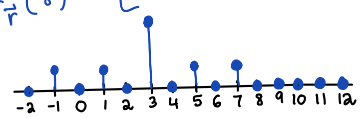
$$\text{corr}_{\vec{r}}(\vec{s})[-1] = \sum_{-\infty}^{\infty} r[n]\, s[n+1] = (0\cdot 1)+(0\cdot 1)+(0\cdot 1)+(0\cdot -1)+(1\cdot 1)+(1\cdot 0)+(1\cdot 0)+(-1\cdot 0)+(1\cdot 0) = 1$$

r[n] 

s[n] 

We can plot the cross-correlation:

$$\text{corr}_{\vec{r}}(\vec{s}) = [\ldots 0\ 1\ 0\ 1\ 0\ 5\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0 \ldots]^T$$



What is the delay?

$$k^* = \underset{k}{\text{argmax}}\ \text{corr}_{\vec{r}}(\vec{s})[k]$$

$k^* = 3$ is the delay!

Cross-Correlation Properties

- $\text{corr}_{\vec{x}}(\vec{y}) \neq \text{corr}_{\vec{y}}(\vec{x})$
- Length of the cross-correlation:
  If $\vec{x} \in \mathbb{R}^N$ and $\vec{y} \in \mathbb{R}^M$, then the length of $\text{corr}_{\vec{x}}(\vec{y})$ is $N+M-1$
- Autocorrelation: $\text{corr}_{\vec{x}}(\vec{x})$
  $k^* = \underset{k}{\text{argmax}}\ \text{corr}_{\vec{x}}(\vec{x})[k] \longrightarrow k^* = 0$

Periodic Signals

We assumed above that the satellite transmits the signal only once and then stops, but it actually repeats the code over and over again.

- Cross-correlation is "periodically expanded" instead of zero-padded.

Choosing Gold Codes

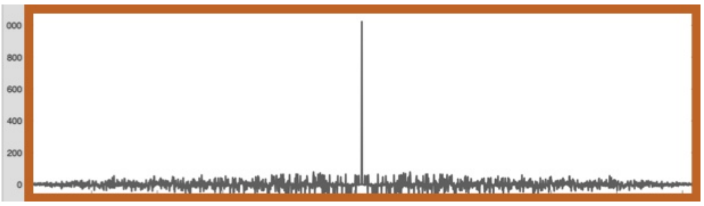
We know the signals our 24 satellites transmit must be orthogonal.
What are other properties they should have?

Our received signal consists of various time-delayed transmitted signals, along with noise.

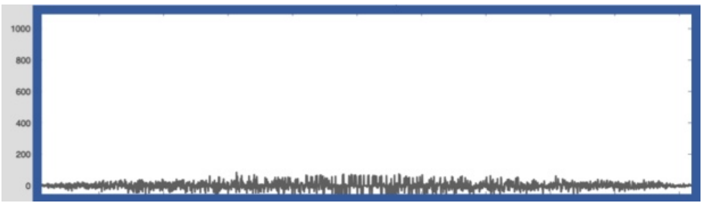
$r[n] = s_1[n-\tau_1] + s_2[n-\tau_2] + \text{noise}[n]$

If we find the correlation of this with a transmitted signal, we have:

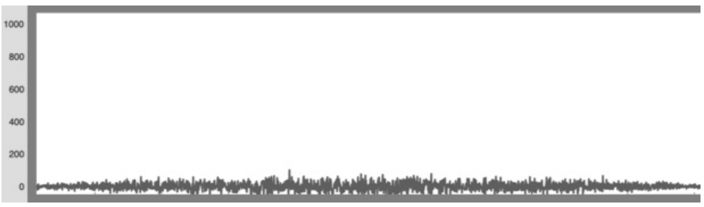
$$\text{corr}_{\vec{r}}(\vec{s}_1)[k] = \langle r[n], s_1[n-k] \rangle$$
$$= \underline{\langle s_1[n-\tau_1], s_1[n-k] \rangle} + \underline{\langle s_2[n-\tau_2], s_1[n-k] \rangle} + \underline{\langle \text{noise}[n], s_1[n-k] \rangle}$$



Autocorrelation should be like an impulse.

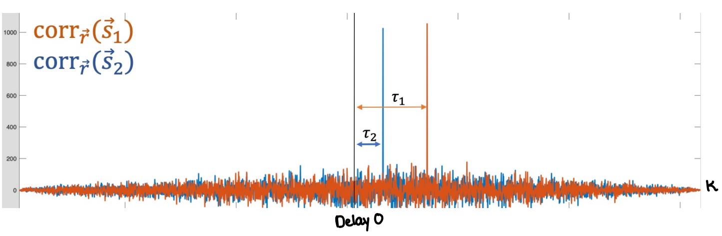


Cross-correlation with other satellites should be small



Cross-correlation with noise should be small
But can this always be true?

What the cross-correlations should look like:



Summary

1. Identify which satellites are in the transmitted signal
2. Find the time delay/shift for each satellite signal
3. Utilize the shifts to find the distance to each satellite
4. Use trilateration to find unknown coordinates.

But what if our receiver clock is not in sync with the satellites? What if we have a lot of noise?
Stay tuned for tomorrow! ∵