CSM: EECS 16A **(Designing Information Devices and Systems I)**

Worksheet #13

Term: **Fall 2019** *Name:*

Problem 3: Revisitng the Acoustic Positioning System

Learning Goal: Understand how to apply properties of inner products, cross correlations, trilateration, least squares, and OMP to build an acoustic positioning system. Learn how to identify the pros/cons when applying different techniques in building the system.

In this question, we will revisit the **Acoustic Positioning System** (APS) and learn how to build it from the ground up using what we know about cross correlation, trilateration, least squares, and Orthogonal Matching Pursuit (OMP).

Recall that in an APS, we have a number of satellites (let's say there are m) transmitting gold codes, and you are a person standing at a location with the coordinate $\vec{x}$, with your phone as the receiver of the signals.

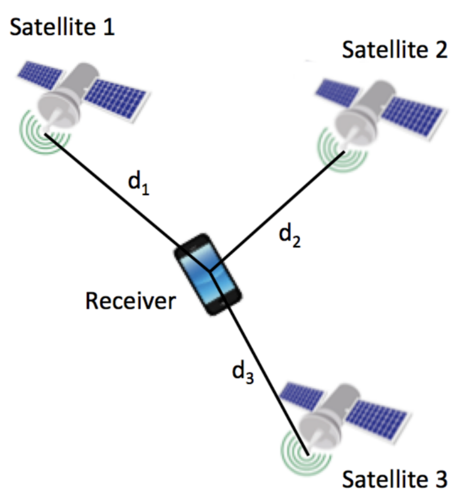
You receive a linear combination of these transmitted signals:

$$\vec{r} = \alpha_1 \vec{s}_1^{(\tau_1)} + \alpha_2 \vec{s}_2^{(\tau_2)} + \ldots + \alpha_m \vec{s}_m^{(\tau_m)}$$

As shown in the expression above, each signal is scaled by a constant which is a *"message"* the satellite encodes into its signal while transmitting.

To solve for our current position, we can set up a system of equations based on our current position $\vec{x}$, the position of each satellite $\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_m$, and the distance from our current position to each satellite d_i.

Here's an illustration of the APS (given 3 satellites):



1. Based on the provided information above, which of the following variables are known? Which are unknown (the ones we are trying to solve for)?

- The position of each satellite $\vec{p}_i$
- Our current position $\vec{x}$
- The transmitted signals $\vec{s}_i^{(\tau_i)}$
- The distances from our current position to each satellite d_i

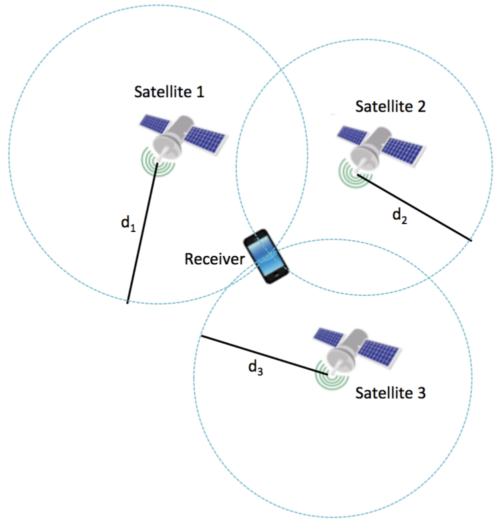
2. How can we express d_i in terms of $\vec{p}_i$ and $\vec{x}$? How many such equations can we set up in total?

3. Recall that an APS can help us determine where we are. Which variable given in the question corresponds to our current location?

4. Based on the system of equations we set up in part 2, how can we solve for our current position?

5. If our current position can be represented by an n-dimensional vector, how many satellites do we need to be able to solve for our position?

6. As shown below geometrically, we can represent the area of coverage by each satellite as a circle with a radius of d_i. Explain why the radius of each circle is d_i, and how finding our current position is equivalent to finding the point of intersection among the circumferences of the circles.



7. Now that we have figured out where we are, it is time for us to decode the message! Recall that what our phone receives is a linear combination of these transmitted signals:

$$\vec{r} = \alpha_1 \vec{s}_1^{(\tau_1)} + \alpha_2 \vec{s}_2^{(\tau_2)} + \ldots + \alpha_m \vec{s}_m^{(\tau_m)}$$

From the expression above, which of the following variables are we trying to solve for?

- The scaling (attenuating) constant α_i
- The original signal sent by the satellite $\vec{s}_i$
- The delay in the transmission of the signal τ_i

8. To solve for the unknown variables from the previous part, we can use the **least squares** method. How can we reformulate the given expression and information above as a **least squares** problem? In other words, if we are to rewrite the problem in the form:
$$A\vec{v} \approx \vec{b},$$
what would A, $\vec{v}$, and $\vec{b}$ be equal to respectively?

9. What is the solution using the **least squares** method?

10. Does there exist a case where least squares would not work? Write down a sufficient condition where we have to resort to some other techniques to recover the original message.

11. If we are to solve this problem using Orthogonal Matching Pursuit (OMP) instead of least squares, what is one assumption made by this technique?

12. Describe what the stopping conditions for OMP are. In other words, how do we know when to stop running the OMP algorithm?