MIT 6.808 Mobile and Sensor Computing (The IoT Class)
Spring 2022

Hari Balakrishnan

Notes for Lecture 2: Wireless Localization & Positioning 2/2/22

Plan for the these three lectures: Principles, GPS, Wi-Fi loc, ultrasonic loc, Wi-Fi sensing

Goal: Fundamentals of wireless localization and positioning and systems to achieve localization

Today, we have three goals:

1. Motivating applications: why do we care about location, localization, positioning?
2. Key approaches: a) sensing modalities for location, b) principles behind the methods
3. Key ideas in GPS

Next two lectures: wireless LAN (Wi-Fi) localization, Bluetooth Low Energy (BLE) ranging, ultrasonic/acoustic (+RF) localization, wireless (Wi-Fi) sensing.

Motivating applications

Numerous location-based apps have burgeoned in the last fifteen years; have dominated the consumer app landscape. Can you imagine a smartphone without location services/apps?

A non-exhaustive list (not just mobile but broader) includes:

- Navigation and route-finding (outdoors but also indoors)
- Traffic and road data
- Emergency assistance (911, E911, crash assistance)
- Gaming
- Tracking people and items: hospitals, factory floors, supply chains, packages
- Tracking livestock and animals
- Finding lost items, finding things, tracking things
- Delivery systems for goods
- Ride-hailing and taxis: people mobility
- Robotics: factory floors, homes, etc.
- Security: access based on location
- Fitness and activity
- Contact tracing
- Autonomous vehicles
- *Nefarious uses are possible:* sale of information for advertising purposes or worse

Sensing modalities: what low-level signals can be used to sense for data useful for loc?

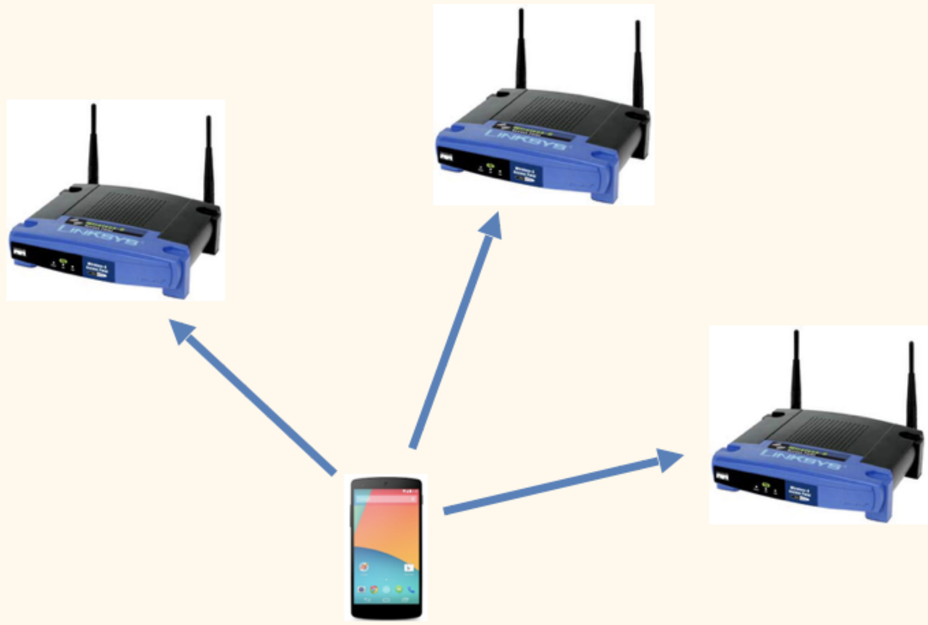
1. Radio: GPS, cellular loc, Wi-Fi loc, BLE
2. Ultrasound/sound
3. Inertial sensors: accelerometer and gyroscope – what do these do?
4. Cameras and LIDAR
5. Magnetometer: provides directional info relative to magnetic north
6. Barometer: provides altitude info

System architecture choices

Where does the localization computation run? On the asset being localized or in the infrastructure? What are the pros and cons of each architecture?

1. Device- or terminal-based: computation runs on the asset being localized

   a. Pros: better privacy, no reliance on a data network (can work when disconnected)
   b. Cons: consumes cycles, drains energy, could be computationally intensive, sometimes might not be as accurate (e.g., when we get one constraint at a time on a moving object, can't leverage extra information (e.g., relative positions of other objects, other things in the environment)
   c. Examples: GPS, location services on smartphones
2. Infrastructure- or network-based: computation runs in the infrastructure



a. Pros: can run more computationally intensive methods in the cloud, easier to track in indoor settings, antenna arrays in infrastructure possible, can localize or track without anything on the asset (or with less on the asset)
b. Cons: fundamentally a tracking system so less privacy (e.g., cellular carriers maintain location info of users), often requires uplink network connectivity (though Radar is an exception)
c. Examples: Radar, E911 services

Note that some research systems combine, e.g., offload expensive GPS computations from a more impoverished device to the cloud.

Key approaches

1. Proximity or identity-based: if asset A is near an *anchor* with known location L, then asset A is near L.
2. Lateration: obtain distances to or from known things (beacons, transmitters, receivers) and solve for unknown asset position.
3. Angulation: obtain angles to or from known things and solve for unknown asset position.
4. Fingerprinting: obtain signal strengths or other signal indicators and use pattern matching algorithms to infer asset location.
5. Dead reckoning: starting from a known position, use knowledge of velocity (speed+heading/direction), and/or inertial measurements (acceleration, rotation) to “fill in the gaps” until the next time we obtain a more authoritative position.

Proximity / Identity-based localization

Idea: if I am near something whose location is known, then I am near the known location.

1. When I open a door with my ID card, MIT knows where I am at that time (I think they delete it after some time).
2. If I connect to a campus Wi-Fi AP, the location of my laptop (specifically, the unique MAC address of my laptop) is known
3. War-driving databases to map Wi-Fi MAC addresses
4. Bluetooth ranging (e.g., smartphone maps now work in several of the Boston tunnels)

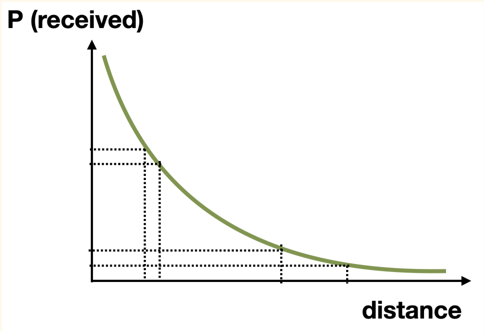
Lateration

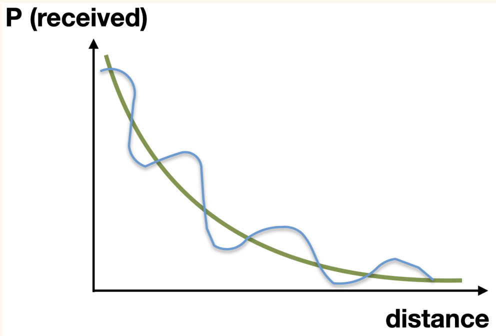
Idea: obtain distances to known things and solve equations to localize. The most successful example is GPS, which we will talk about shortly.

But let's first talk about how to estimate distances. There are several possibilities here, most of which use properties of wireless waveforms (whether radio or ultrasonic/acoustic). In addition, one can also use LIDAR, which uses laser reflections to measure time of flight.

Approaches to estimate distances:

1. **Radio signal strength**, aka RSSI (received signal strength indicator). Key property is that radio signals attentuate in amplitude; in free-space the amplitude attenuates as 1/d. The power is the square of amplitude and attenuates as $1/d^2$ in free space. Indoors with obstacles, the attenuation is even larger, $1/d^n$ where n is between 2 and 4 typically. With some calibration, can try to estimate distances or at least know if we're getting closer or further away.
   a. Pros: available on commodity hardware and can be used fairly widely
   b. Cons: (i)Small change in power leads to large distance change as distance grows, (ii) multipath errors (constructive and destructive interference)





2. **Time-of-flight:** There are several ways to measure the time taken for a signal to travel and when we multiply that by the speed of the signal we can estimate a distance.

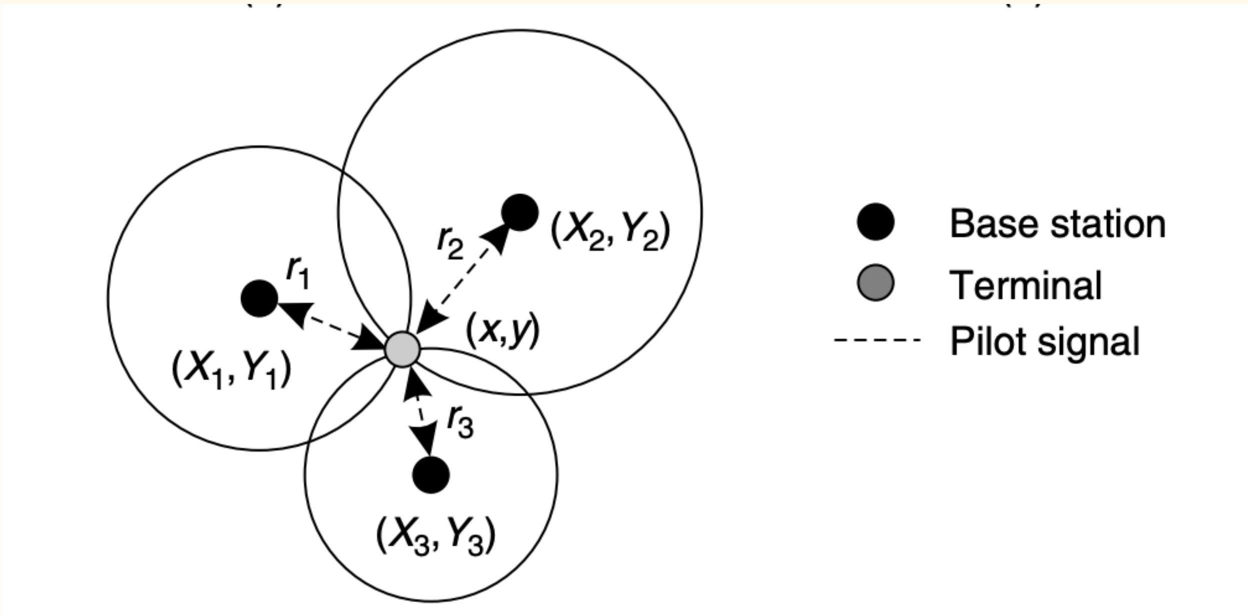
The Global Positioning System (GPS) uses this method. The satellites are tightly time-synchronized. The (mobile) device has an unknown clock offset, but assume for now it's time is also perfectly sync'd. Each satellite sends a signal that arrives with a unique code. Since the satellites are so far away the signals are quite weak, but their bit rate is also quite low (about 50 bits/s of information with significant coding redundancy). Each satellite has a unique code and the device searches for different codes (with assisted GPS, it has an approximate location so doesn't have to search for every satellite). Thanks to the tight time-sync, we can estimate time of flight.

In practice the satellites are tightly time-synced, but the mobile has an unknown offset. The trick is to add another variable capturing this offset and we can solve for four equations (x, y, z, and the time offset) using distances (times of flight) from four satellites.

3. **Time difference of arrival (TDOA):** In some systems we might not have the absolute distances to known locations, but we can estimate the time difference between pairs of known locations.

What can we do with these distances (or in the case of TDOA, distance differences)? The answer is different types of *lateration*.

The principle is shown below (Figures 6.4 and 6.5 of the assigned reading):



(c)

This is called circular lateration in 2D and spherical in 3D. Technically in 3D we need four spheres, not three, because there are two points where three spheres meet, but oftentimes the other point solved for is impossible (it's on the "other side" of the three satellites in GPS, for example!).

Hyperbolic lateration solves similar equations using the properties of the hyperbola, $x^2 - y^2 = C$.

Angulation

In angulation, e.g., triangulation, we use a set of *angles* to localize; no distances are involved. For example:

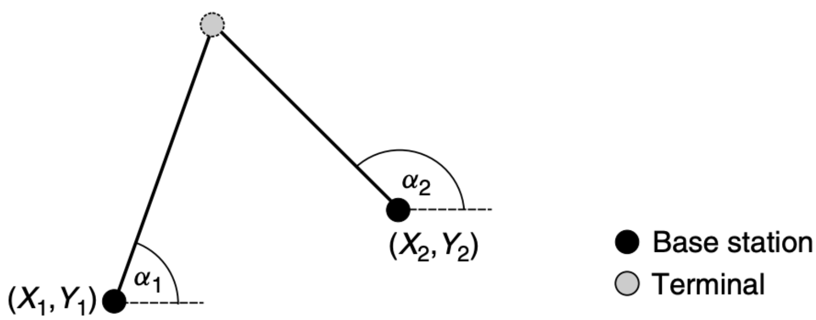


Figure 6.9 Angulation.

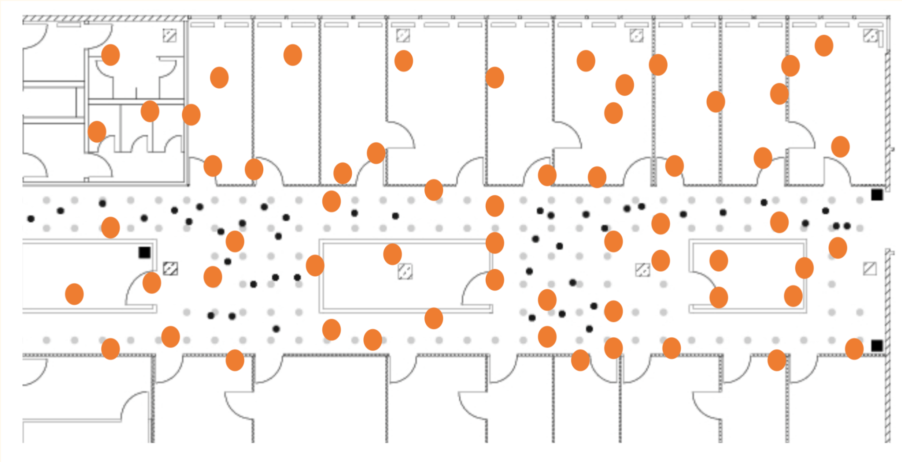
So how do we obtain angles? There are a few possibilities:

1. Use directional or sectorized antennas
2. Use multiple antennas and calculate angles using the phase difference of received signals (trigonometry): two antennas lead to errors, need several (array)

Fingerprinting, aka pattern matching

Produce a training database of RSSI samples at several locations, e.g., (AccessPoint1 RSSI 7, AP2 RSSI 14, AP3 RSSI 24, AP6 RSSI 8) for location L. Repeat at several locations.

Then, when presented with a location we want to identify, do a nearest-neighbor search. There are several metrics possible to obtain the closest *signature*. We will study a couple of these next lecture.



Dead reckoning

Short for “deduced reckoning”.

Idea: Starting from a known position, use knowledge of velocity (speed+heading/direction), and/or inertial measurements (acceleration, rotation) to “fill in the gaps” until the next time we obtain a more authoritative position.