CMPE58C: Sp. Tp. Mobile Location Tracking & Motion Sensing

Global Positioning System (GPS)

Can Tunca

-

Fall 2022

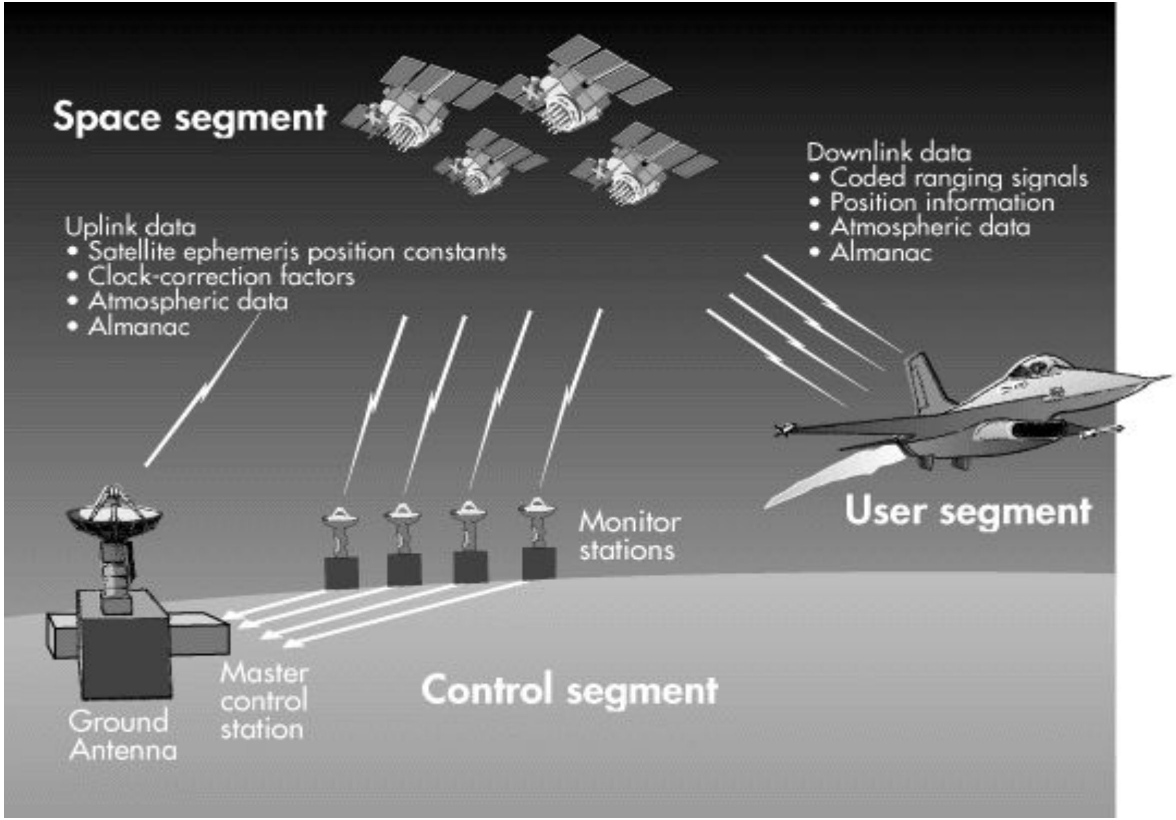
GPS vs. GNSS

- GNSS: Global Navigation Satellite System
  - The general name of satellite-based navigation systems
- GPS: GNSS of USA
- Other GNSS's
  - Galileo: EU
  - Beidou: China
  - GLONASS: Russia
- Mostly compatible with each other
  - A typical "GPS receiver" will be able to operate with most
- We will focus on GPS, but the principles are the same...


GPS Technology Evolution
First portable GPS, 25 lb (11 kg)
1978
First handheld GPS $1,000
1989
In-car GPS
1998
First GPS smartphone
2005
GPS chip, 2 mm
2013

Segments

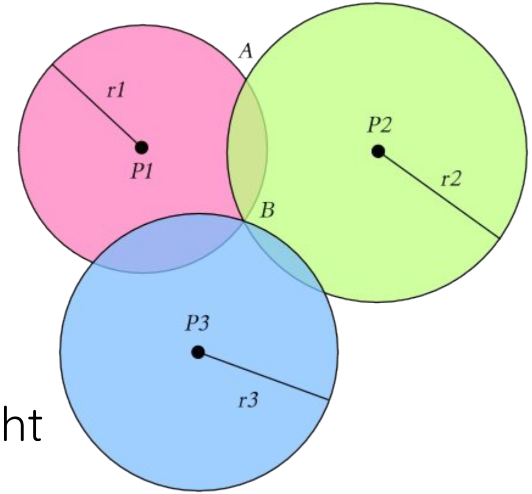
- Space segment
- Control segment
- User segment



Space Segment

- Satellites deployed at Medium Earth Orbit (MEO): ~20200 kilometers
- Currently ~31 satellites
- Minimum 24 satellites needed for optimal operation (rest are for redundancy)
- ~30 satellites retired
    - Have reached end-of-life
    - Disposed to another orbit and turned off
- NOT Geostationary! (we will see why in a minute...)

Positioning Approach



- Approach: Trilateration
- One way communications
  - Satellites transmit, devices receive and locate themselves
- Theory: Estimate distances to satellites via time-of-flight
  - We have to know the transmission and reception times
  - Distance = $\Delta t \times c$
- Reception times are easier
  - The receiver can trust relative measurements from its own clock
- Transmission times are tricky…
  - What if the clocks of the satellites are different?
  - We'll get to that…

Ranging: How to estimate distance?

- Satellite and mobile device clocks may be different, but let's pretend they are the same:

$$d_{pseudo,S1} = c\,(t_m - t_{S1})$$

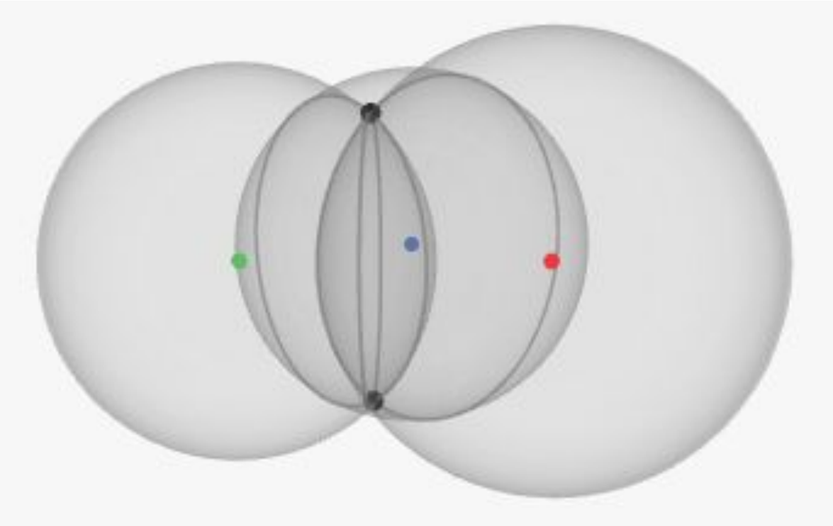
- Called **pseudo-range**, since it is not the true distance…

$$d_{true,S1} = d_{pseudo,S1} + ct_{offset,S1}$$

- Real distance has to accommodate for the clock offset between satellite and mobile device

Assuming pseudo-range is true…

- We only need 3 satellites to build a geometrical model
- Why not 4 satellites?
  - Red, blue, green are satellites
  - Black dots are possible positions
  - Can we eliminate one?



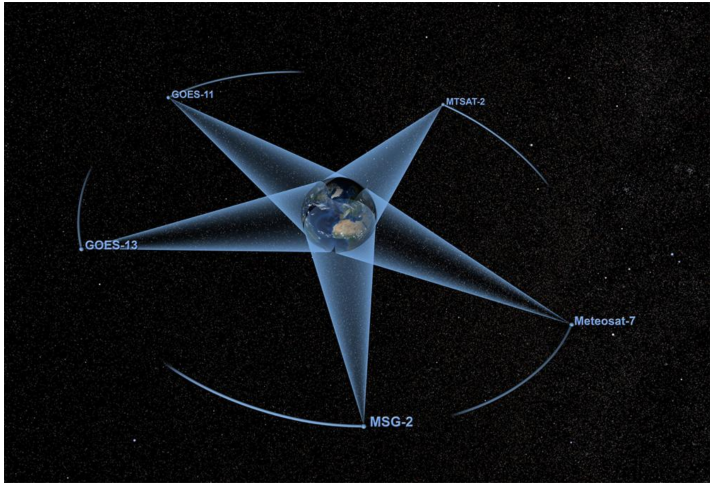
(Spoiler: We'll eventually need the 4th satellite for something else!)

Revisiting geostationary orbits...

- Orbital period equal to Earth's rotational period
- Fixed position in the sky

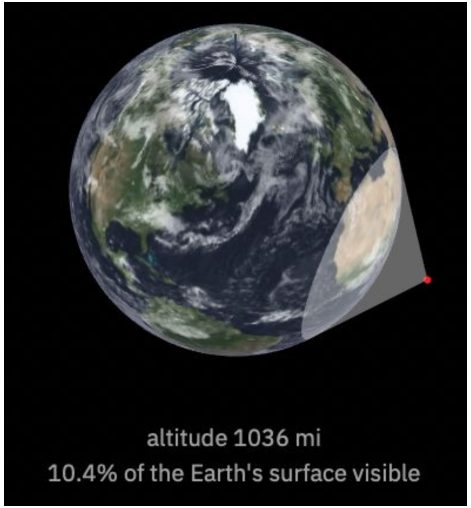
  (faces the same point on Earth always)
  - To achieve this the satellite has to be above the equator
- ~35700 km altitude (High Earth Orbit)
- Examples:
  - Communications satellites (including TV satellites)
  - Meteorological satellites...
- Geosynchronous:

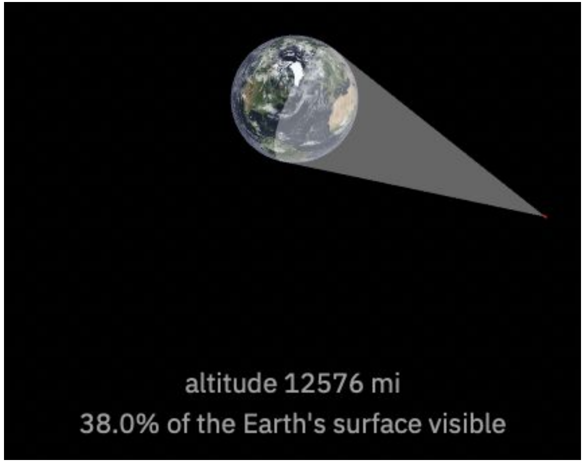
  Not necessarily fixed point on Earth

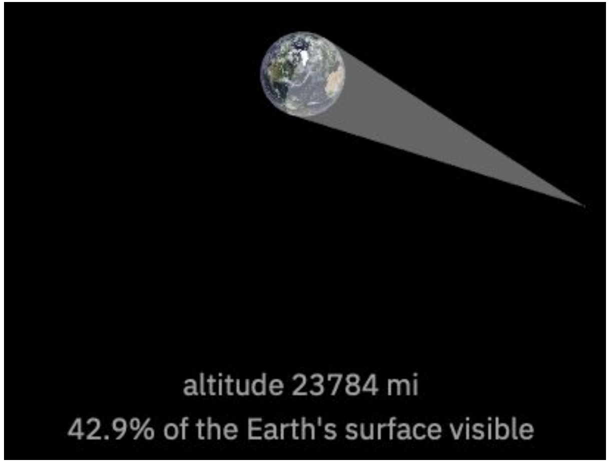


Geostationary advantages

- Satellite position easier to deduce (always fixed relative to Earth)
- More visibility
  - Means fewer satellites
  - After a point the gains become incremental though...

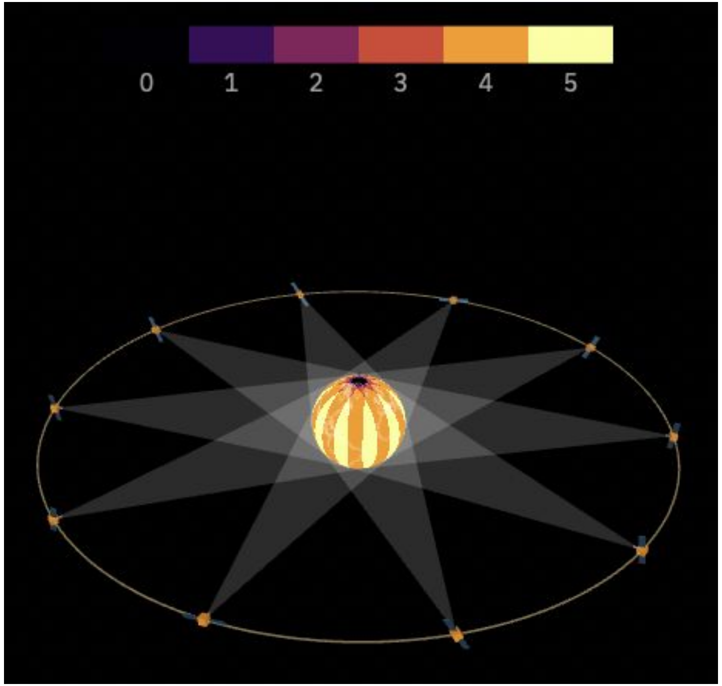




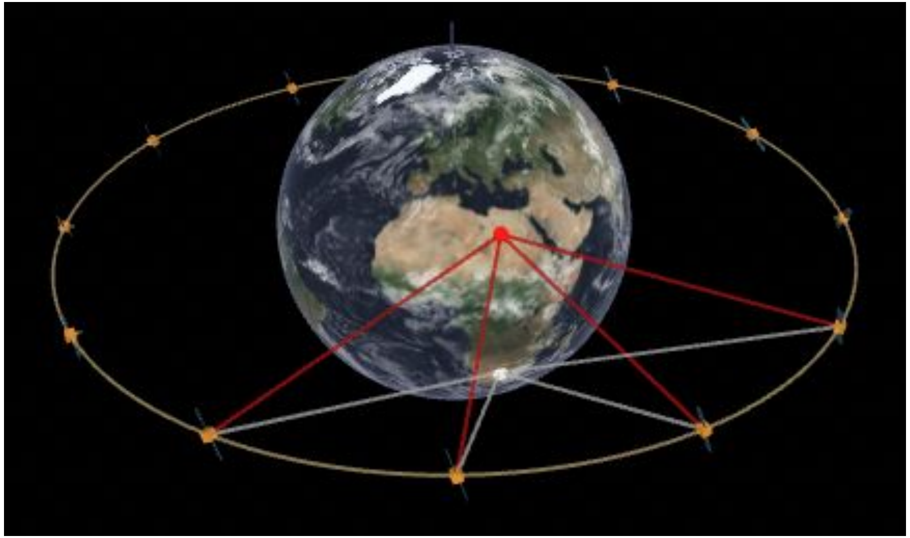


Why not geostationary?

Impossible to cover the poles
No matter how many satellites



All satellites on the same orbital plane
Can't eliminate the second possible position
It also is on Earth!



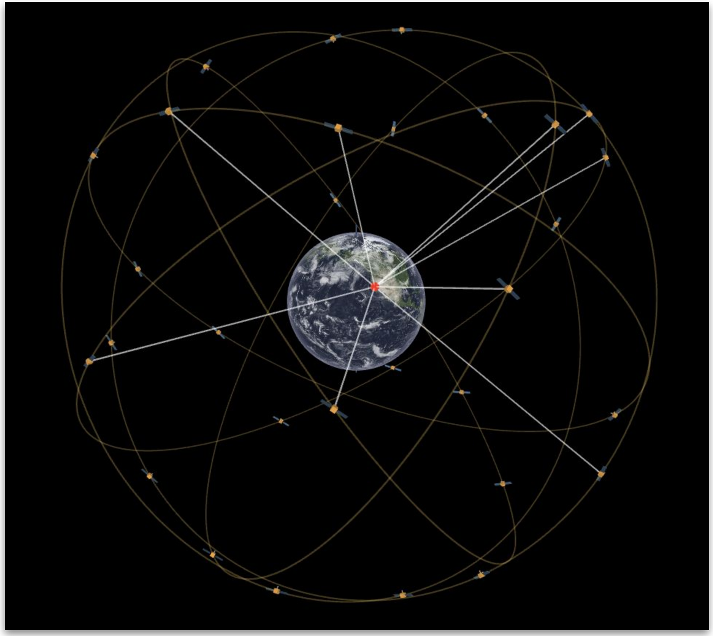
(Note: Satellites drawn closer than they actually are for better demonstration)

Why not geosynchronous?

- Geosynchronous may solve these problems because the satellites don't need to be on the same plane...
- But, the satellites are still too far
  - Signal attenuation, more power requirement
- Instead GPS orbits are semi-synchronous
  - 20200 km altitude, much closer to Earth
  - Passes the same location on Earth twice a day (deterministic coverage!)
  - Earth visibility: 38% vs 42% (geo), still quite good

GPS Orbits

- 6 orbits with ~5 satellites each
- Everywhere on Earth covered by at least 4 satellites always (Usually even more!)
- Semi-synchronous: Periodical visibility could be relied on



Back to Satellite Clocks...

- Satellites must share synchronized clocks: Atomic clocks!
- This means clock offset from a device to all satellites are the same:

$$t_{offset,S1} = t_{offset,S2} = t_{offset,S3} = \ldots = t_{offset}$$

- We'll also need to solve for t_{offset}
- 4 variables: x, y, z, t_{offset} => 4 satellites needed!
- In short, 4th satellite is needed to compute the clock offset between the satellites and the mobile device

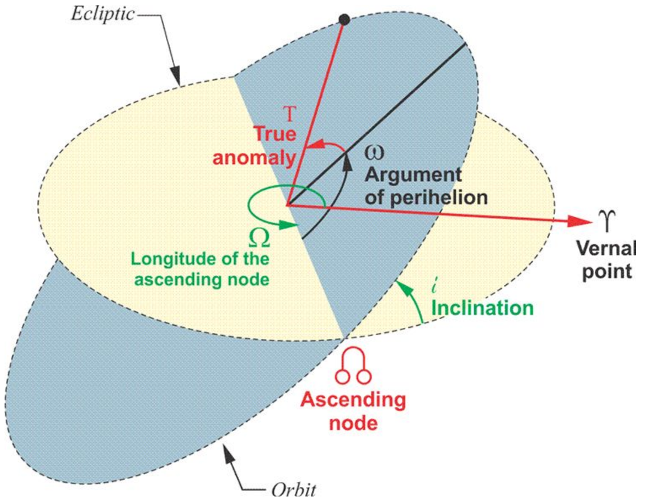
GPS is a TDoA system

- In essence GPS uses time-difference-of-arrival
  - We do not need to know the actual time-of-flight!
- More than 4 satellites can be used to improve accuracy
  - Constellation redundancy helps
  - Not all visible satellites may be at an ideal location (we’ll revisit this)
- We haven’t yet talked about how the device knows the satellite locations!
  - Required for TDoA model
  - Not as simple as fixed infrastructure, the satellites move!

Satellite Positions

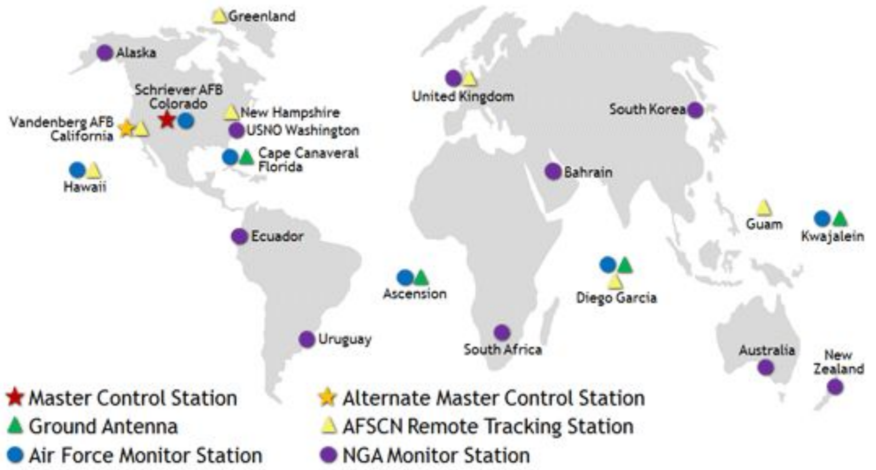
- **Ephemeris:** The orbital trajectory each satellite follows
- Known in advance and transmitted in the packages (we'll see how)
- Pretty accurate, but small errors add up...
- So a series of ground stations to monitor the satellites:

Control Segment



Control Segment

- Monitor the health of the satellites (functionality + orbit)
- Activate spares if needed
- Compute/upload ephemeris corrections
- Compute/upload clock corrections
- Issue orbital corrections if needed
  (satellites have engines!)
- Discard EOL satellites to other orbits



User Segment

- Mobile devices: where the actual positioning is done
- GPS payload also includes **almanac**
    - List of <u>other</u> satellites and their approximate orbital information
- Mobile device fetches ephemeris and almanac
- Calculates a fix using pseudo-ranges if it can find 4+ satellites

GPS Payload

- **25 frames** with **5 subframes** with **10 words** with **30 bits**
- Every subframe has TLM and HOW
- **TLM:** Telemetry word
    - A fixed preamble
    - Bits for integrity check
- **HOW:** Handover word
    - Contains the timestamp
- **Clock corrections**
    - GPS week number
      (used to compute fine time together with HOW)
    - + time corrections (we'll get to that)
- **Health bit**
    - To temporarily disable a satellite (during corrections)
- **Almanac**
    - A frame contains only 1/25 pages!

1 frame

TLM	HOW	Clock corrections, health
TLM	HOW	Ephemeris
TLM	HOW	Ephemeris
TLM	HOW	Almanac/other – page 1
TLM	HOW	Almanac – page 1

TLM	HOW	Clock corrections, health
TLM	HOW	Ephemeris
TLM	HOW	Ephemeris
TLM	HOW	Almanac/other – page 2
TLM	HOW	Almanac – page 2

...

TLM	HOW	Clock corrections, health
TLM	HOW	Ephemeris
TLM	HOW	Ephemeris
TLM	HOW	Almanac/other – page 25
TLM	HOW	Almanac – page 25

GPS Signal

- Three bands
  - **L1:** 1575.42 MHz with a bandwidth of 15.345 MHz <u>(we'll focus on this)</u>
  - **L2:** 1227.6 MHz with a bandwidth of 11 MHz
  - **L5:** 1176.45 MHz with a bandwidth of 12.5 MHz
- Each satellite transmits with a power of ~30W
- Could be received at ~10^{-16} W!

  (Remember, these satellites are pretty far away: ~20000 km)
- This is thousands of times below typical noise levels for other tech!
- The answer is operating at a low data rate on a high bandwidth!

GPS Signal Characteristics

- Shannon's Theorem:

$$C = B \log_2(1 + S/N)$$

Capacity (bits/second)

Bandwidth (Hz)

Signal-to-noise ratio

- With big B and small C, we can get away with a very small S/N!
    - B is 15.345 MHz, which is a high number given 1575.42 MHz frequency
    - S/N is around 0.000002! The actual signal hides in the noise!
    - Results in a bit rate of **50 bps** (pretty low for other comms tech, but enough)

How do we increase bandwidth?

- Signal spreading!
- Each bit we want transmit is **spread** into multiple bits by XORing with a spreading code (i.e. chips)

1	**0**	**0**	**1**	Data to be transmitted, D
0010	**0010**	**0010**	**0010**	Spreading code (0010), C
1101	**0010**	**0010**	**1101**	C XOR D

Signal Spreading

- To decode the original signal do the inverse operation
- Inverse of XOR is XOR!
- i.e. XOR the received bits with spreading code (C)

0010	**0010**	**0010**	**0010**	Spreading code (0010), C
1101	**0010**	**0010**	**1101**	Received (C XOR D)
1111	**0000**	**0000**	**1111**	De-spread signal

What if there are some errors?

- We should expect errors for such low S/N ratios
- An example with two erroneous bits:

0010	**0010**	**0010**	**0010**	Spreading code (0010), C
1100	**0010**	**0010**	**1001**	Received (C XOR D)
1110	**0000**	**0000**	**1011**	De-spread signal

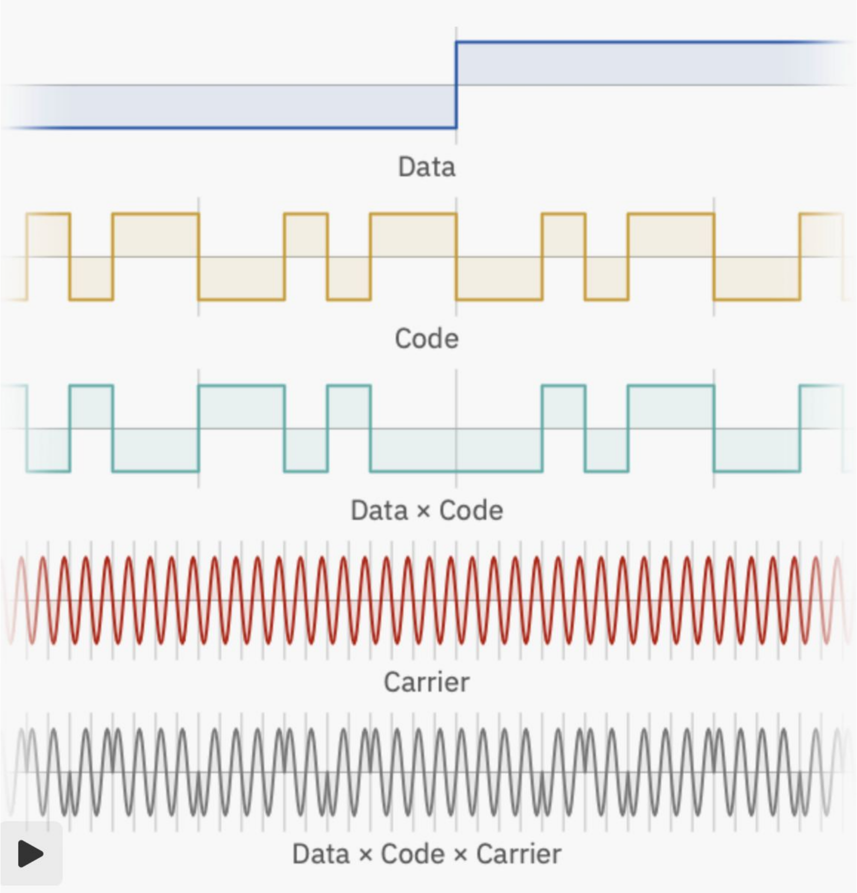
We can still decode!

- **Principle:** The chips change at a much higher rate than source bits

  In reality, the code is much bigger! (1023 bits, 20 repetitions for 1 source bit)

Putting it all together...

- Actual data (bits)
  - 50 bps
- Spreading code
  - 1023 bits x 20 repetitions
  - 1 Mbps
- Data ⊕ Code
- Carrier wave
  - 1575.42 MHz (L1)
- Final signal
  - **BPSK** (Binary Phase Shift Keying)
  - Carrier phase is flipped on bit transitions

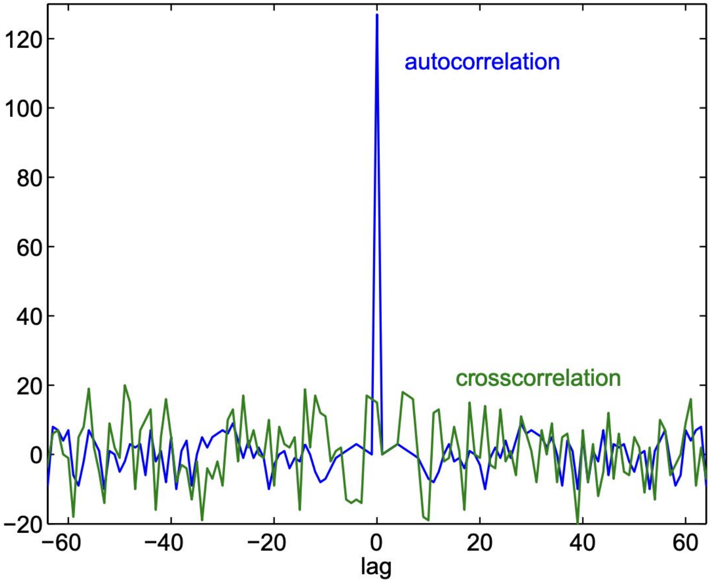


How to deal with interference?

- Spreading code allows us to deal with noise
- But if multiple satellites transmit simultaneously the signals would interfere
- Each satellite has a different code!
- Codes must weak cross-correlation
- A code must have strong autocorrelation

  (only when it is perfectly aligned with itself)
- **Gold codes!**

for 127 bits long gold codes



Gold Codes

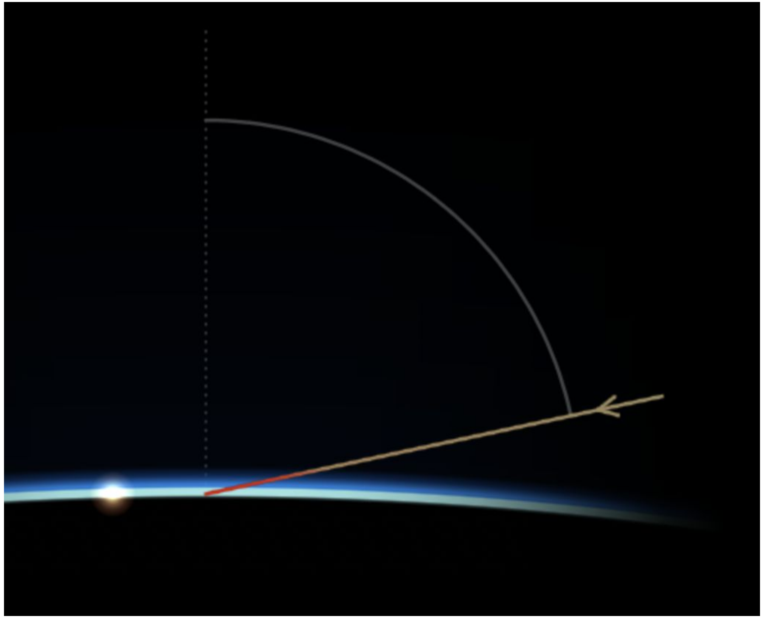
- Code for each satellite is known in advance
- If we can correlate a known code with the received signal, then we have locked on to that satellite!
- Low cross-correlation of gold codes helps as a code cannot accidentally produce a high quality source signal with another satellite’s data

Error Sources

- Ionospheric delays (~5m)
- Satellite selection and signal acquisition (~3m)
- Ephemeris errors (~2.5m)
- Satellite clock errors (~2m)
- Multipath (<1m in LOS scenarios)

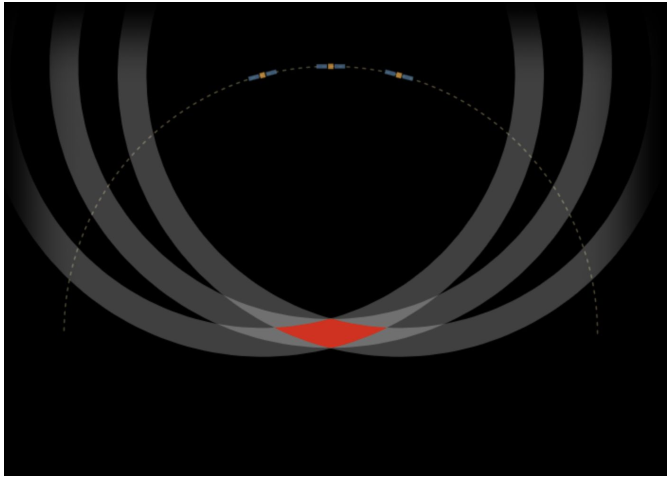
Error Source: Ionospheric delays

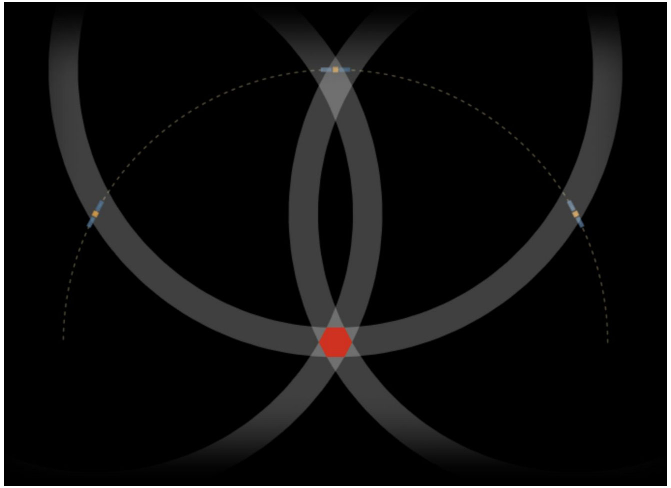
- 50-1000 km above the surface of the Earth
- Signal is delayed and deflected
- Worse for satellites closer to horizon
- Can be modeled to some degree
- Also listening to other bands (L2, L5) may be useful, as different frequencies have different characteristics! (can get error down to 0.5m!)
- Lower atmosphere also affects signal, but not that much (negligible)



Error Source: Satellite Selection

- Better to select a well spread subset
- But, we should not select satellites too close to horizon (ionospheric effects)
- Constellation redundancy helps, but there may be not enough choice due to other factors
- Signal acquisition could also be problematic in some lower tier devices





Error Source: Ephemeris

- It’s a complex task to compute accurate orbital trajectories
- Control segment has an important job here
- But even if the ephemeris transmitted by the satellite is up-to-date and accurate, it may not have reached the device yet…
- Cold start, warm start, hot start…

Turning on the receiver…

- **Cold start:** The device is turned on for the first time (or after considerable time)
  - Have to download the whole almanac
  - Have to search for all satellites in view (based on almanac)
  - Takes ~12.5 minutes! (mostly time to download the whole almanac)
- **Warm start:** Subsequent runs…
  - Receiver caches almanac (valid for ~180 days), ephemeris (valid for ~4 hours)
  - Much quicker to get a fix
- **Hot start:** If you had a location fix within the last 4 hours
  - You can get a fix without needing any satellite data
  - Knowing rough location also helps
  - What if satellite data is transmitted to receiver via some other means? (A-GPS?)

Error Source: Clock

- Special relativity: Fast moving objects experience time dilation
  - i.e. satellite clock runs slower
- General relativity: Clock runs faster in lower gravitational field
  - Effect more than special relativity
- Overall satellite clock runs faster ~38 microseconds per day than the receiver clock
- If not corrected would result in ~11.4 km ranging error!
- Satellites apply correction themselves + control segment issues fine corrections

- less gravity than on Earth ⇒ general relativity: +45μs/day
- move very fast (~ 3km/s) ⇒ special relativity: -7μs/day

Net result: + 38 μs/day + eccentric variations

Relativistic offset is built into the clocks (programmed slower before launch), variations are built into receiver code.

Clocks are really important...

https://www.bbc.com/news/technology-35491962

- January 2016: Clock of 15 satellites were 13 microseconds off!
- Lasted for ~12 hours, GPS was essentially broken!


Sign in | News | Sport | Weather | iPlayer | Sounds

NEWS

Home | UK | World | Business | Politics | Tech | Science | Health | Family & Education

Technology

GPS error caused '12 hours of problems' for companies

By Chris Baraniuk
Technology reporter

4 February 2016

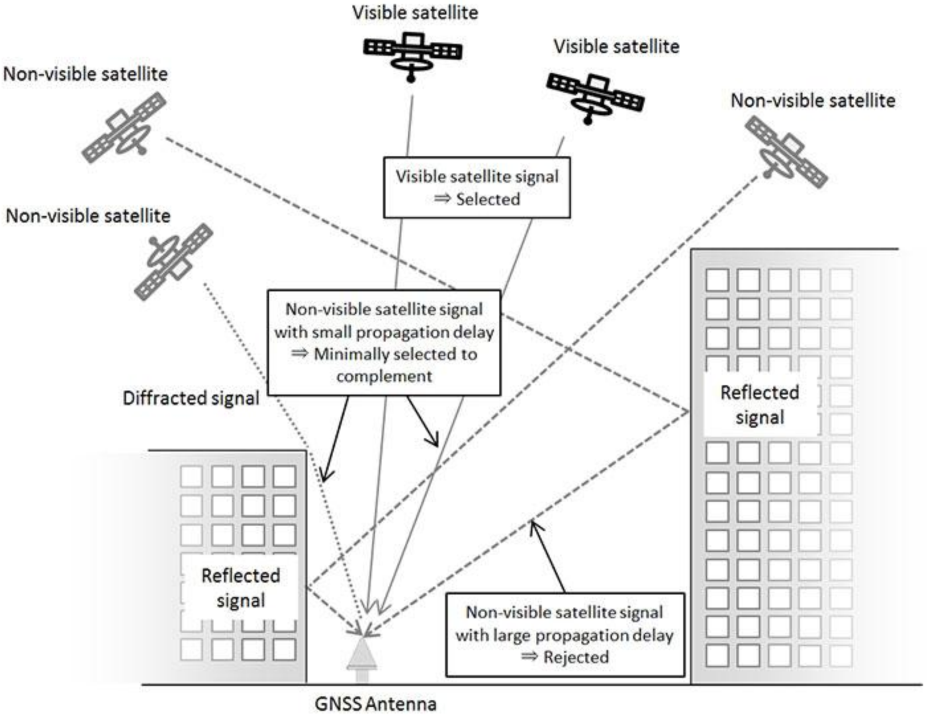
Share

GPS for getting universal time

- Another use case for GPS!
- Calculating time offset and knowing the transmitted satellite time allows us to get accurate global time
- Around 1 ns in theory
- Telecommunications, banking and a lot more systems depend on this!
- The incident on January 2016 also affected these systems...

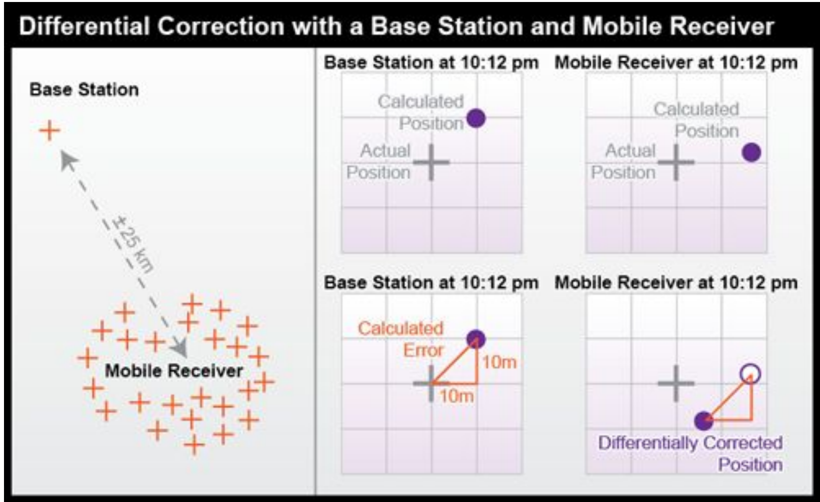
Error Source: Multipath

- Dominant in NLOS cases
- Could be combated with satellite selection, to some degree...



Differential GPS

- A good method to mitigate many of these error sources
- Reference stations at precisely known locations
- Can estimate the error sources and calculate corrections
- Broadcast them to nearby receivers
  - Receivers has to be physically nearby, otherwise the corrections don't apply
  - So, we can't simply issue corrections to devices anywhere on Earth, there must be a station close by
- Can get positioning error down to a few centimeters!

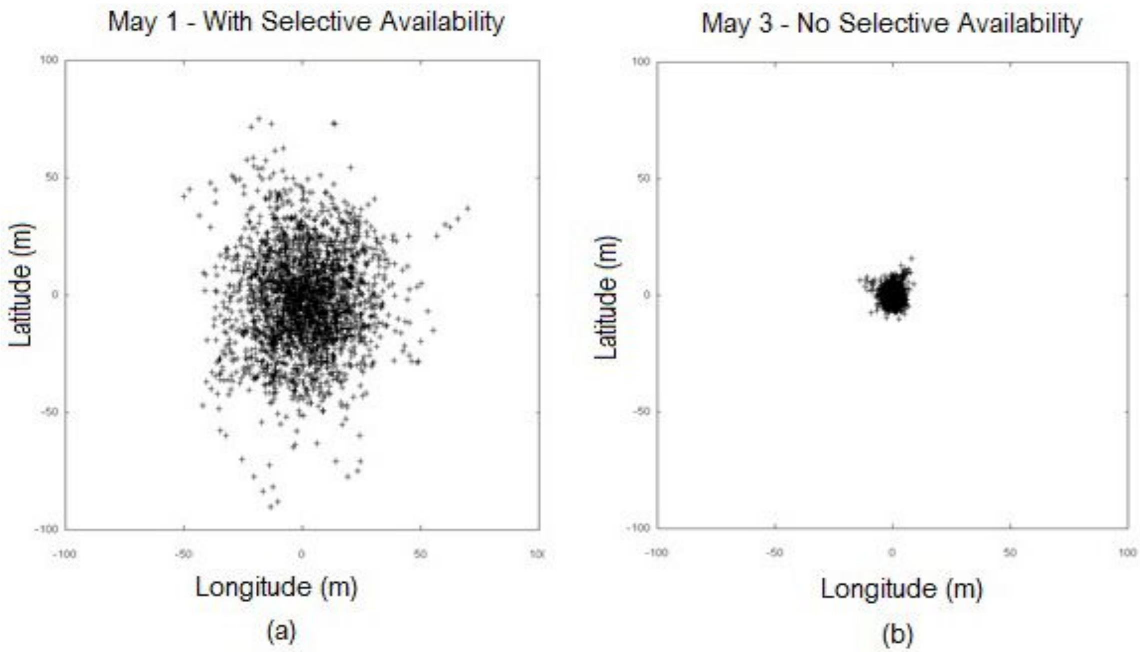


Assisted GPS (A-GPS)

- It's a wide term that includes any assistance really…
  - Could also include Differential GPS
- What if almanac and ephemeris info could be delivered to the device by other means?
  - A cellular network can deliver these info effortlessly
  - Hence we can always do hot start! (How your phones are able to get a quick fix)
- Not solely relying on GPS
  - Cellular networks can also compute position (although not as accurate as GPS)
  - Previously visited WiFi APs could be added to a DB with known locations (kind of like fingerprinting)
  - Results in an even faster fix that gradually gets better!

Selective Availability

- Intentional degradation of public GPS signals
- Turned off in May 2000



A highly recommended resource

https://ciechanow.ski/gps/

- A good explanation of GPS
- Starts very basic, gets slowly to the most complicated bits!
- Very good and interactive visualizations