2/13/23

Coordinate Systems

• 2 forms of coordinate systems
  1) Inertial / Fixed
  2) Body Frame

ECEF (Earth Centered Earth Fixed) "global"

• cartesian (x, y, z)
• Geocentric Polar (r, θ, λ)
• spheroidal (ellisoidal or geodetic) (ϕ, λ, h)

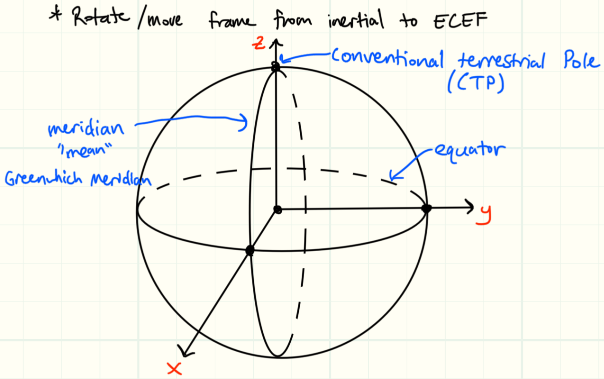
East - North - Up "Local"

• Plane tangent to the center of the earth
• requires a reference point

$\left.\begin{array}{l} \text{>> lla2ecef} \\ \text{>> ecef2lla} \\ \text{>> lla2enu} \\ \text{>> enu2lla} \end{array}\right\}$ textbook appendix

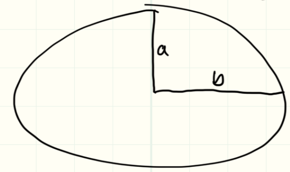
GPS Coordinate System

• needed global coordinate system
  – ECEF → fixed to center of earth
    ↳ EARTH IS NOT RIGID (CHANGES SHAPE)
    ↳ Not elliptical (bulges at equator, flatter at poles)
    ↳ Not uniform
  – Use sea level as an approximate shape
  – Satellites + user to use same coordinate system
    ↳ user frame fixed to earth (not moving from user standpoint)
    ↳ SV in inertial frame (fixed in space)
      * Rotate / move frame from inertial to ECEF



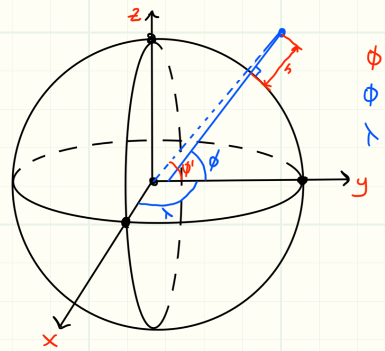
World Geodetic System

• 1984, DoD developed WGS84
  – Not accurate, but globally used
  – Not intuitive and not on earth's surface
• ECEF in Latitude, Longitude, Height / Altitude
  ↳ relative to hypothetical surface
• Geoid: surface consisting of the mean sea level (MSL) and under land if reached by small, frictionless channels
  – geoid is non-even (mass distribution of earth is not equal)
  – difficult to calculate the perfect geoid
  – Approximated (curve-fit) by a spheroid or ellipsoid



$a \approx b + 20\text{km}$

$$\left.\begin{array}{l} \text{- eccentricity} \Rightarrow e^2 = \dfrac{a^2 - b^2}{a^2} \\ \text{- flattening} \Rightarrow f = \dfrac{a - b}{a} \end{array}\right\} e^2 = 2f - f^2$$

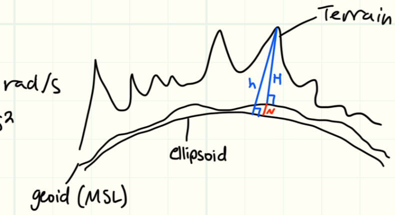


ϕ' Geocentric Latitude
ϕ Geodetic Latitude
λ Longitude (from greenwich meridian)

WGS84

* Not inertial frame
* Datum is ellipsoid to model geoid
* origin is earth's center of mass
  $a = 6{,}378{,}137$ m
  $b = 6{,}356{,}752$ m
* Different datums produce different coordinates for same point
* All points of constant ϕ are circular
* All points of constant λ are elliptical
* 1° of λ is different distance depending on ϕ
  ~ 110 km at equator
  ~ 80 km at 45° ϕ
* Go between ECEF and ellipsoid with model of ellipsoid (i.e. WGS84)

$a = 6378137.0$ m
$1/f = 298.257223563$
$\omega_{earth} = 7.292115\ (10^{-5})$ rad/s
$g_c = 3986004.418\ m^3/s^2$
$c = 299792458$ m/s

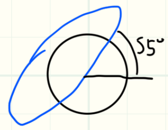


– GPS Provides "h", height above ellipsoid
– WGS84 provides contours of the geoid with respect to ellipsoid (N)

2/15/23

Satellite Orbits

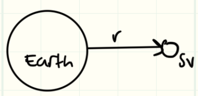
- 24+ SVs (24 minimum spec. for fully functional)
- 6 Orbital Planes
  - 55° inclination angles (less coverage at poles)
    - optimize user coverage
  - approximately circular
  - 12 hr orbits
    - SV position repeats ~ every 23:56 hours
    - 20,162 km from equator (26561 from center of the earth)
    - Travel at ~2.7 $\frac{km}{s}$



Sidereal Time: 24 sidereal hour = 1.002738 * 24 solar hours

Newton vs. Kepler

$\mu_E = 3{,}986{,}004.418 \frac{m^3}{s^2}$
$G = 6.674 \times 10^{-1} \frac{m^3}{kg\, s^2}$



$$F = \frac{G M_E m_{SV}}{r^2} = \frac{\mu_E m_{SV}}{r^2}$$

Using Newton's Laws: $\Sigma F = m\ddot{x}$

- SV: $m_{sv}\ddot{r}_s = \frac{-G M_E m_{SV}}{r^2} \cdot \frac{\vec{r}}{r} = \frac{-G M_E m_{SV}}{r^3}\vec{r}$

- EARTH: $M_E \ddot{r}_E = \frac{G M_E m_{SV}}{r^3}\vec{r}$

Differencing Equation:

$$M_E m_{sv}\ddot{r}_s - M_E m_{sv}\ddot{r}_E = \frac{-G M_E m_{sv}^2}{r^3}\vec{r} - \frac{G M_E^2 m_{sv}}{r^3}\vec{r}$$

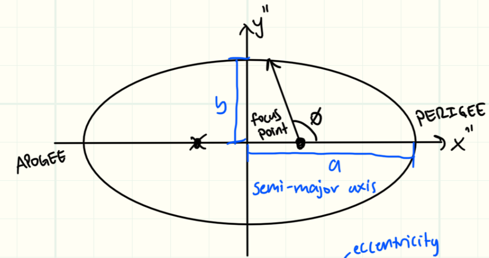
$$M_E m_{sv}\ddot{\vec{r}} = \frac{-G}{r^3}\vec{r}\left(M_E m_{sv}^2 + M_E^2 m_{sv}\right)$$

$$\ddot{\vec{r}} = \frac{-G}{r^3}\vec{r}\left(M_E + m_{sv}\right)$$

$$\ddot{\vec{r}} + \frac{G M_{tot}}{r^3}\vec{r} = 0 \quad \leftarrow$$ 6th order non-linear homogeneous diff. eq.
- requires 6 ICs $\vec{r}(0)$ and $\dot{\vec{r}}(0)$

Solution results in Kepler's 3 Laws of Orbits:
1) Elliptical motion
2) Motion is faster when closer to orbiting body
3) $t_{orbit}^2 = k d_{avg}^3$ ← average orbital distance

Position of SV in Orbital Frame



$x'' = r\cos(\phi)$
$y'' = r\sin(\phi)$
SV Position

$T_p = \frac{2\pi}{\sqrt{\mu_E}} a^{3/2}$ ← orbital period

$e = a^2 - b^2$ ← eccentricity
$b = a\sqrt{1-e^2}$

$r = \frac{a(1-e^2)}{1+e\cos(\phi)}$

$n = \frac{2\pi}{T_p} = \sqrt{\frac{\mu_E}{a^3}} = \sqrt{\frac{GM}{a^3}}$ ← mean angular velocity

$|r| = \frac{a(1-e^2)}{1+e\cos(v)} = a(1-e\cos(E))$ ← eccentric anamoly

$M = n(t - t_{PERIGEE})$
$= E - e\sin(E)$ ← mean anamoly

$v = \tan^{-1}\left(\frac{\sin(E)\sqrt{1-e^2}}{\cos(E)-e}\right)$

$x = |r|\cos(v)$
$y = |r|\sin(v)$ ← true anomoly

* Must solve for M iteratively until $\Delta E < 1e^{-12}$

Taking derivatives:

$\dot{M} = \dot{E}(1-\cos(E)) = n$

$\dot{r}_T = \dot{R}_3(\theta) r_1 + R_3(\theta)\dot{r}_1$

$\dot{r}_x = \frac{-n\,a\sin(E)}{1-e\cos(E)}$ $\quad \dot{r}_y = \frac{n\,a\cos(E)\sqrt{1-e^2}}{1-e\cos(E)}$

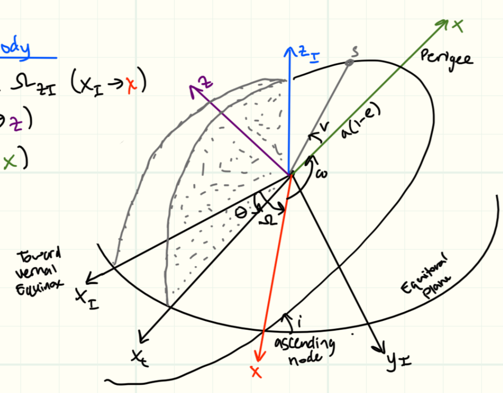
$$\ddot{\vec{r}} = -\frac{GM_{tot}}{r^3}\vec{r}$$

Rotation Matrices

- Orbital Frame → ECEF frame
- Order of rotations is critical!

Inertial to Body

1st: θ_{z_I} and Ω_{z_I} $(x_I \to x)$
2nd: i_x $(z_I \to z)$
3rd: ω_z $(x \to x)$



Position Translation with Rotation Matrices

$$\vec{r} = R_3(\omega)\, R_1(i)\, R_3(\Omega)\, \vec{r}_I$$

$\omega°$ about z-axis; $i°$ about x axis; $\Omega°$ about z-axis

Properties of Rot. Mat.:

1) $R^{-1} = R^T$
   ↳ $R^{-1}(\theta) = R(-\theta) = R^T(\theta)$

$$\vec{r} = R_3(-\Omega)\, R_1(-i)\, R_3(-\omega)\, \vec{r}_T$$

$$\vec{r}_T = R_3(\theta)\, \vec{r}_I$$

about same axis as RAAN, combined in GPS

GPS SV Position Calculation

- GPS calculates position from ascending node:
  $\phi = \omega + v$
  - ∴ no final rotation about z-axis
- Position is calculated as:
  $x = r\cos(\phi)$
  $y = r\sin(\phi)$
- Rotating the position from orbital to inertial:
  $\vec{r}_I = R_3(-\Omega) R_1(-i)\vec{r}$
- GPS uses Longitude of Ascending Node (LAN) which combines the Right Ascension of Ascending Node (RAAN) and the Greenwich Apparent Sidereal Time (GAST) rotations as:
  $\Omega = \Omega_{LAN}(t) = \Omega_{RAAN} - \theta_{GAST}(t)$
  - Makes it easier to go to ECEF

$$R_3 = \begin{bmatrix} \cos(\Omega) & -\sin(\Omega) & 0 \\ \sin(\Omega) & \sin(\Omega) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(i) & -\sin(i) \\ 0 & \sin(i) & \cos(i) \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_3(-\Omega)\, R_1(-i) \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x'\cos(\Omega) - y'\sin(\Omega)\cos(i) \\ x'\sin(\Omega) + y'\cos(\Omega)\sin(i) \\ y'\sin(i) \end{bmatrix}$$

$x' = r\cos(\omega + v)$
$y' = r\sin(\omega + v)$
$z' = 0$
$\Omega = \Omega_{LAN}$

Orbit Perturbations

- Rockets firing interventions
- Non-central (uniform) gravitational force field
- Equitorial bulge
  - Produced torque on SV
  - Harmonic perturbations (twice per orbit)
- Gravity of sun and moon
- Solar radiation pressure

$$\ddot{\vec{r}} = -\frac{GM_{\oplus}}{r^3}\vec{r} + F_{dist}(r, \dot{r}, t)$$

$$\frac{GM_{\oplus}}{r^3}\vec{r} \gg F_{dist}(r, \dot{r}, t)$$

- GPS does not broadcast its position, but instead ephemeris correction terms (curve fits) to calculate position (Kepler Orbital mechanics

GPS Ephemeris

- Ephemeris: orbit data
- Ephemerides: individual parameter of orbit

- You provide "t"
- GPS provides "t_{oe}"
  - Nominal Ephemerides:
    - $\sqrt{a}$ - ω_0
    - e - i_0
    - M_0 - Ω_0
  - Perterbation Effects:
    - Δn, (IDOT), $\dot{\Omega}$ } secular perturbation
    - C_{ucos}, C_{usin}
    - C_{rcos}, C_{rsin} } Harmonic Perturbation
    - C_{icos}, C_{isin}

1) Non-spherical earth
2) Tidal effects
3) Solar radiation pressure

SV Position Subtilties

- "t" is transmission time
  ↳ must be corrected by transit time (approx $\frac{\text{range}}{c}$ or $\frac{\text{pseudorange}}{c}$)
  - can solve iteratively with user position (exact)
  - may account for earths rotation during transit time
- check for a vs $\sqrt{a}$

SV Clock Data Corrections

- $B^k = B^k_{broadcast} + dB^k$
- $B^k_{broadcast} = \underbrace{af_0}_{\text{clock bias}} + \underbrace{af_1}_{\text{clock rate}}(\underbrace{T_{tr}}_{\text{Transmission (based on GPS Time)}} - \underbrace{T_{oc}}_{\text{Time of correction}}) + \underbrace{af_2}_{\text{FND}}(T_{tr} - T_{oc})^2 + \underbrace{\Delta t_{rel}}_{\text{based on orbit}} + \underbrace{T_{gd}}_{\text{from SV}}$

Ephemeris Updates

- Ephemeris are updated every 2 hours
- Issue of Data Ephemeris (IODE)
  - change of IODE indicates an update to the ephemeris
- Ephemeris are good for 4 hours (maintains spec)
- Ephemeris is group of "ephemerides"
- Must check for clock rollover of $t - t_{oe}$ at

How to Get full set of Ephemerides?

- Navigation Message (data broadcast from each SV)
  - → can only get data from SV's being tracked
- Overlaid on GPS code "chips"
- GPS C/A code repeats 20 times per bit
- 50 bits/sec
- 1500 bits = 1 frame , 1 frame = 30 sec
  - ↳ IT TAKES 30 sec TO RECIEVE ALL EPHEMERIS TO COMPUTE THE SV POSITIONS

Subframes

10, 30 bit words forming 6s subframe

5 subframes form 30s frame (1500 bits)

#			
1	TLM	HOW	Block 1: Clock Correction
2	TLM	HOW	Block 2: Ephemeris
3	TLM	HOW	Block 3: Ephemeris
4	TLM	HOW	Block 4: Message
5	TLM	HOW	Block 5: Almanac

↳ 25 frames required for complete almanac
- used to determine when to start tracking different SV

TLM + HOW

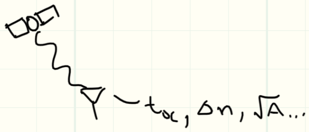
- TLM begins with an 8 bit sync pattern
  - 10001011 (0x8B)
  - occurs every 6 sec
- HOW is 17 most significant bits (MSB) of the 19 bit Time of Week (TOW) count
  - 6 seconds of resolution
- GPS Time is 29 bits
  - 10 bits for week (1024)
  - 19 bits for TOW (1.5 sec increments)

Overview

- Subframe = 6 sec = 300 bits
- 5 subframes per frame
- Subframes 1-3 repeat every 30 sec
- Subframes 4-5
  - 25 "pages" for each, repeating after page number 25
  - Pages increment every 30 sec
  - 25 x 30 seconds = 12.5 minutes to guarantee reception of all 25 pages

2/20/23

1) orbit is model / diff EQ (6th Order)
2) Actual orbit is not perfect
   - Report SV PVT
   - Corrected orbital parameters
     - ↳ provide "curve fit adjustments" to nominal orbit params



$$t_{TRANS} = t_{RECD} - \frac{r}{c} \quad \rho$$

? ?

SV POS

↳ my pos estimate

GPS Time

- GPS uses its own reference
- GPS → UTC ≈ 18 seconds
- GPST = # weeks + # second in week (12:00 am sunday)
- GPS weeks is Modulo 1024
  - Midnight Jan. 6, 1980
  - Midnight Aug. 22, 1999
  - Midnight Apr. 6, 2019

  } Week 0
- 604,800 seconds in a GPS week
- $b^j = t^j - t_{GPS} = af_0 + af_1(t - t_{oc}) + af_2(t - t_{oc})^2 + \Delta t_R - T_{GD}$

Oscillator Stability

$$\frac{f - f_0}{f_0} = \frac{\Delta f}{f_0} = \frac{\Delta t}{T}$$

$$f(t) = f_0 + \Delta f + \dot{f}(t - t_0) + \eta$$

$$\Delta t = \int f(t) = \Delta t(t_0) + \frac{\Delta f}{f_0}(t - t_0) + \frac{\dot{f}}{2f_0}(t - t_0)^2 + \int \eta \, dt$$

- GPS Time is maintained to ~10 ns

Stability Analysis of Clocks

↳ Allan variance

- $\sigma_y^2(\tau) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(y_{i+1} - y_i)^2$
- $N = \frac{\tau}{T_s}$