Overview of Kalman Filter Theory and Navigation Applications

Day 3

Michael L. Carroll

Feb 25, 2004

Day 3: Practical Kalman Filtering, Strapdown Navigation and Inertial Error Dynamics
Segments

- Practical Kalman Filtering: Some Numerical Considerations
- The Navigation Problem and Reference Frames
- Inertial Error Dynamics
- The Basic 9-State Kalman Filter
- Challenges of INS Filters

Day 3, Segment 1
Some Practical Considerations (Numerical)
Topics

- Kalman Filter Theory vs. Reality
- Numerical Stability
- Round-Off Errors
- Matrix Inversion
- UD Factorization

Kalman Filter Theory vs. Reality

- Kalman theory based on multiple assumptions
  - All states are modeled correctly
  - Noise parameters are accurately known
  - Initial values are appropriate
  - Process and measurement noise are both Gaussian
  - Process noise and measurement noise are uncorrelated
  - Measurements are taken at regularly scheduled intervals

Kalman Filter Theory vs. Reality

- How do you select appropriate sensors for the problem at hand?

- Some obvious general rules:

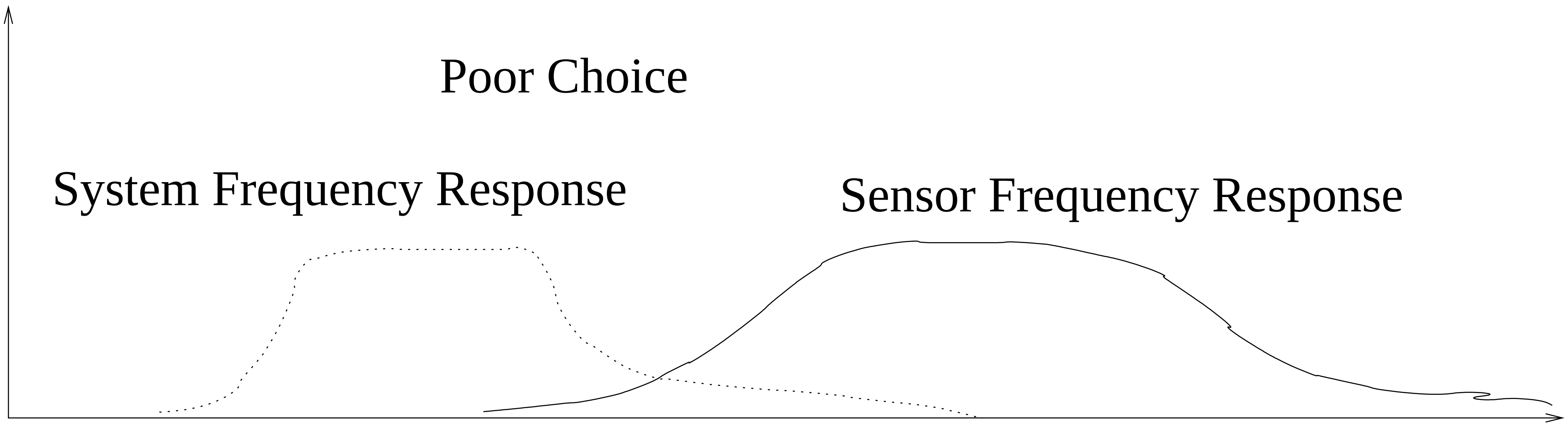
  - the lower the sensor noise characteristics, the more accurate the estimates (narrow sensor noise variance)

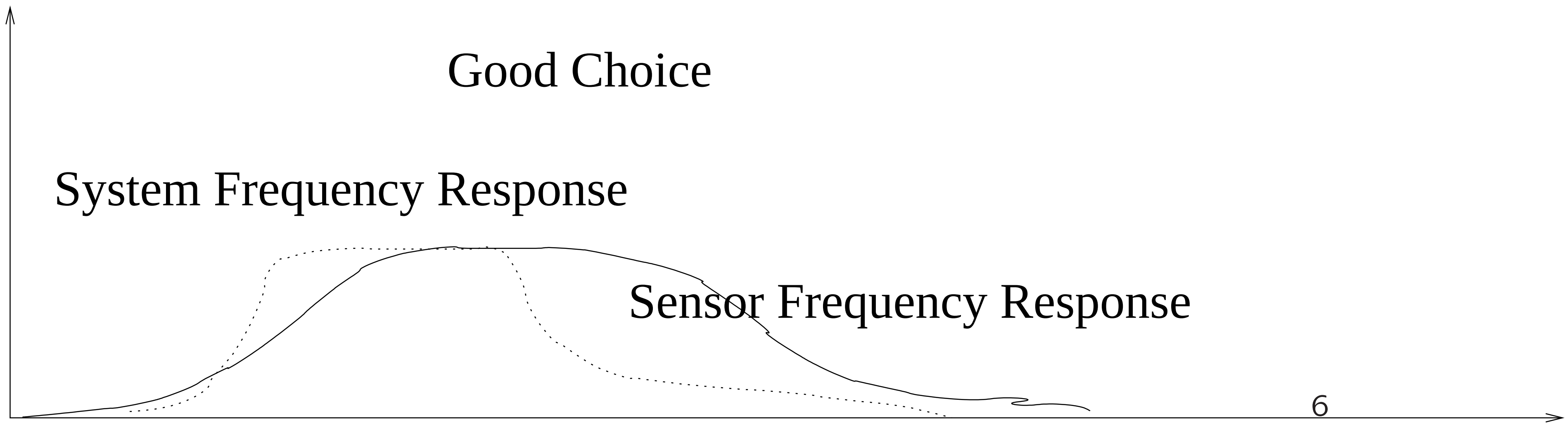
  - the more the merrier, i.e., the more sensors the better

  - maximize the number of directly observable states

Kalman Filter Theory vs. Reality

- Some less obvious general rules:
  - short-term / long-term complementarity (e.g, GPS vs. INS)
  - frequency response of sensor should match that of system
  - noise spectrum of sensor should be maximally disjoint from that of system


Poor Choice
System Frequency Response
Sensor Frequency Response


Good Choice
System Frequency Response
Sensor Frequency Response

Kalman Filter Theory vs. Reality

- Choose sensors to match states.
  E.g., consider the state model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

  with the measurement model

$$z = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + v$$

  Here x_1 is observable but x_2 is not.

Kalman Filter Theory vs. Reality: Sensor Selection

- Examine the H matrix that you would use with this sensor!

- If it renders some states unobservable, then consider adding additional sensors for those states, or replacing this sensor with one that is more favorable.

Filter Divergence

- Kinds of Divergence
- Causes and Remedies

Kinds of Divergence

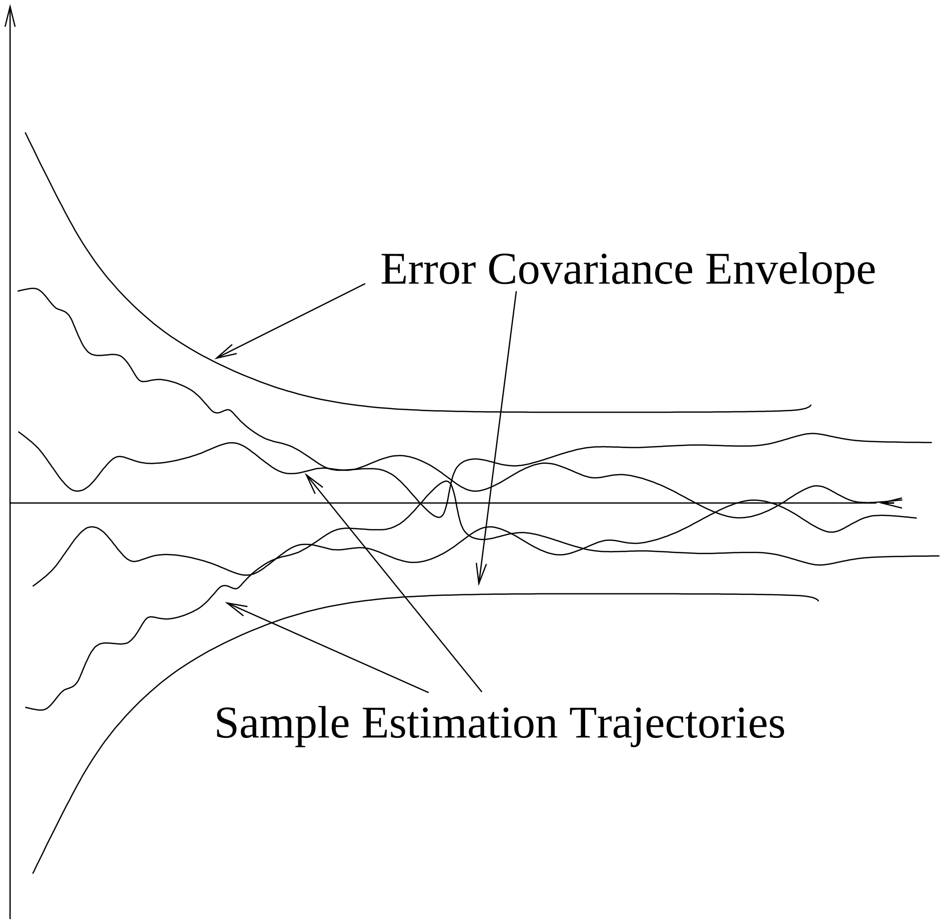
- Mean estimation error is supposed to converge to 0
  - $\tilde{x}_k = \widehat{x}_k(+) - x_k$
  - $\mathsf{E}\left[\tilde{x}_k\right] \to 0$
  - Estimate is unbiased: $\mathsf{E}\left[\widehat{x}_k\right] = x_k$

- Covariance matrix for estimation error is same as state estimation error covariance matrix: $P_k(+) = \mathsf{E}\left[\tilde{x}_k(+)\tilde{x}_k^{\mathsf{T}}(+)\right]$

Kinds of Divergence

- For optimal filter, P will indicate convergence or divergence.
  - One common form of divergence due to round-off will send P to zero

- For suboptimal filters, P will not necessarily indicate divergence
  - P may seem to indicate nominal steady-state values and yet the state estimate may go diverge from the true value



Error Covariance Envelope
Sample Estimation Trajectories

Kinds of Divergence

- Going to sleep: error covariance becomes so small that measurements are no longer believed by the filter

- Loss of symmetry in P

- Loss of positive definiteness: negative numbers on the diagonal!

Numerical Stability

- System stability: Bounded input results in bounded output
- Numerical stability: small errors at each computation step do not grow without bound
- Matrix Conditioning

Matrix Conditioning

- Ill-Conditioned vs. Well-Conditioned
- Condition number of matrix:

$$\text{cond}(M) = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}$$

where λ_{max} is maximum eigenvalue of M, while λ_{max} is the minimum eigenvalue of M.

Matrix Conditioning

- If cond(M) ≈ 1, then M is well-conditioned for matrix inversion

- If cond(M) $>> 1$, then M is ill-conditioned for inversion

Round-Off Errors

- Computer round-off in floating point arithmetic

- Unit round-off value is largest number ε such that $1 + \varepsilon = 1$

- In Matlab, the global symbol "eps" stores the unit round-off value

Round-Off Errors

- Round-off errors generally plague the classical covariance update algorithm

$$P_k(+) = [I - K_k H_k]\, P_k(-)$$

- Alternative, called the Joseph form, is more robust:

$$P_k(+) = [I - K_k H_k]\, P_k(-)\, [I - K_k H_k]^{\top} + K_k R_k K_k^{\top}$$

This form in turn leads to the De Vries mechanization of the Kalman gain equation. To see this, first expand the Joseph form (omitting subscripts for clarity):

$$P(+) = P(-) - KHP(-) - P(-)H^{\top}K^{\top} + KHP(-)H^{\top}K^{\top} + KRK^{\top}.$$

Define

(1) $$F = PH^\top$$

and

(2) $$D = HF + R.$$

We note that D is symmetric, because

$$\begin{aligned}
D^\top &= [HF + R]^\top \\
&= [HPH^\top + R]^\top \\
&= [HPH^\top]^\top + R^\top \\
&= [H^\top]^\top P^\top H^\top + R^\top \\
&= HPH^\top + R \qquad \text{(because } P \text{ and } R \text{ are symmetric)} \\
&= HF + R \\
&= D.
\end{aligned}$$

The gain in standard Kalman gain equation

$$K_k = P_k(-)H_k^{\mathsf{T}}\left[H_k P_k(-)H_k^{\mathsf{T}} + R_k\right]^{-1}$$

can be written as

$$K = FD^{-1}.$$

Using Eqs. (1) and (2), we compute

$$\begin{aligned} P(+) &= P(-) - KF^{\mathsf{T}} - FK^{\mathsf{T}} + KDK^{\mathsf{T}} \\ &= P(-) - K\left[F^{\mathsf{T}} - \frac{DK^{\mathsf{T}}}{2}\right] - \left[F - \frac{KD}{2}\right]K^{\mathsf{T}}. \end{aligned}$$

Letting $G = F - \dfrac{KD}{2}$, we obtain

(3) $$P(+) = P(-) - GK^{\mathsf{T}} - \left[GK^{\mathsf{T}}\right]^{\mathsf{T}}.$$

The De Vries mechanization implements Eq. (3) by subtracting the ij^{th} element of GK^{T} from both the ij^{th} and ji^{th} elements of $P(-)$. The De Vries form of the covariance update equation is valid for any arbitrary gain matrix K, not just the optimal gains computed from Eqs. (1) and (2).

Matrix Inversion

- Matrix inversion is always risky due to potential ill-conditioning
- One way to avoid this is to process measurements sequentially as scalars
- Or at least as low-dimensional matrices

UD Factorization

- Based on Cholesky Decomposition

- Unit Triangular Matrices: U has 1's on the diagonal and all 0's below the diagonal. E.g.,

$$U = \begin{bmatrix} 1 & 5 & 7 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

UD Factorization

- Factor $HPH^{\mathsf{T}} + R$ as UDU^{T}, where U is unit upper triangular and D is diagonal.

- This then allows us to "invert" $HPH^{\mathsf{T}} + R$ by solving the equation unit upper triangular equation $UDU^{\mathsf{T}}X = H$ for the matrix X, which will yield $X = \left[HPH^{\mathsf{T}} + R\right]^{-1} H$

- This last expression is used on the covariance update equation in KH

Exercises

1. Let $v = [1 - 2]^{\mathsf{T}}$. Compute the positive definite, symmetrix matrix $M = vv^{\mathsf{T}}$

2. Use the following algorithm (Grewal & Andrews) to factor M into UDU^{T} form:

```
for j=m:-1:1
    for i=j:-1:1
        covar=P(i,j);
        for k=j+1:m
            covar=covar-U(i,k)*D(k,k)*U(j,k);
        end
        if i==j
            D(j,j)=covar;
            U(j,j)=1;
        else
            U(i,j)=covar/D(j,j);
        end
    end
end
```

Day 3, Segment 2
The Navigation Problem and Reference Frames
Topics

- Basic Navigation Problem

- Earth Referenced Navigation

- Coordinate Frame Conversion

- Basic Strapdown Navigation Equations

Basic Navigation Problem

- Where am I?
- Where am I headed?
- How fast am I going?

Basic Navigation Problem

- What's my position?
- What's my heading and attitude?
- What's my velocity?

Solution in Inertial Space

- In inertial space, it's no problem!

- Just integrate the acceleration twice to get velocity and position.

- And integrate the angular rate once to get attitude.

$$v(T) = v(0) + \int_0^T \frac{d^2R}{dt^2} dt$$

Solution in Inertial Space

- Solution easy in inertial space because ...
  - inertial instruments sense acceleration and rotational rate with respect to inertial coordinate frame
  - inertial frame attached to sensor platform
- But we don't necessarily want the answer to our questions expressed in terms of inertial coordinates!

Earth Referenced Navigation

- Terrestial navigation seeks to answer nav questions with respect to earth

- Global earth reference

- Local or relative navigation

Coordinate Frames

- Inertial
- Earth Fixed
- Local Level
- Body

Coordinate Frames

- Technically speaking, all frames are earth-centered
- Practically speaking, we think of the Body frame as centered in the body of the vehicle (usually at center of gravity).
- Local Level is thought to be located at surface of earth directly beneath the vehicle.

Inertial Reference Frames

- Non-Rotating

- Non-Accelerating

- Fixed or moving uniformly with respect to another inertial frame (fixed stars)

Inertial Reference Frames

- Nav problem "easy" in inertial frames
- Integrate accelerometer outputs once to get inertial velocity; integrate velocity to get position.
- Same with attitude, but only one integration required (usually).
- Would still have to contend with instrument imperfections.

Inertial Reference Frame Integrations

(4) $$\ddot{r} = a_{\mathsf{sf}} + \text{other external forces}$$

(5) $$v = \dot{r} = \int_0^T a_{\mathsf{sf}} dt + \int_0^T (\text{other external forces}) dt$$

(6) $$r = \int_0^T v dt$$

Earth Fixed = ECEF = Earth Centered Earth Fixed

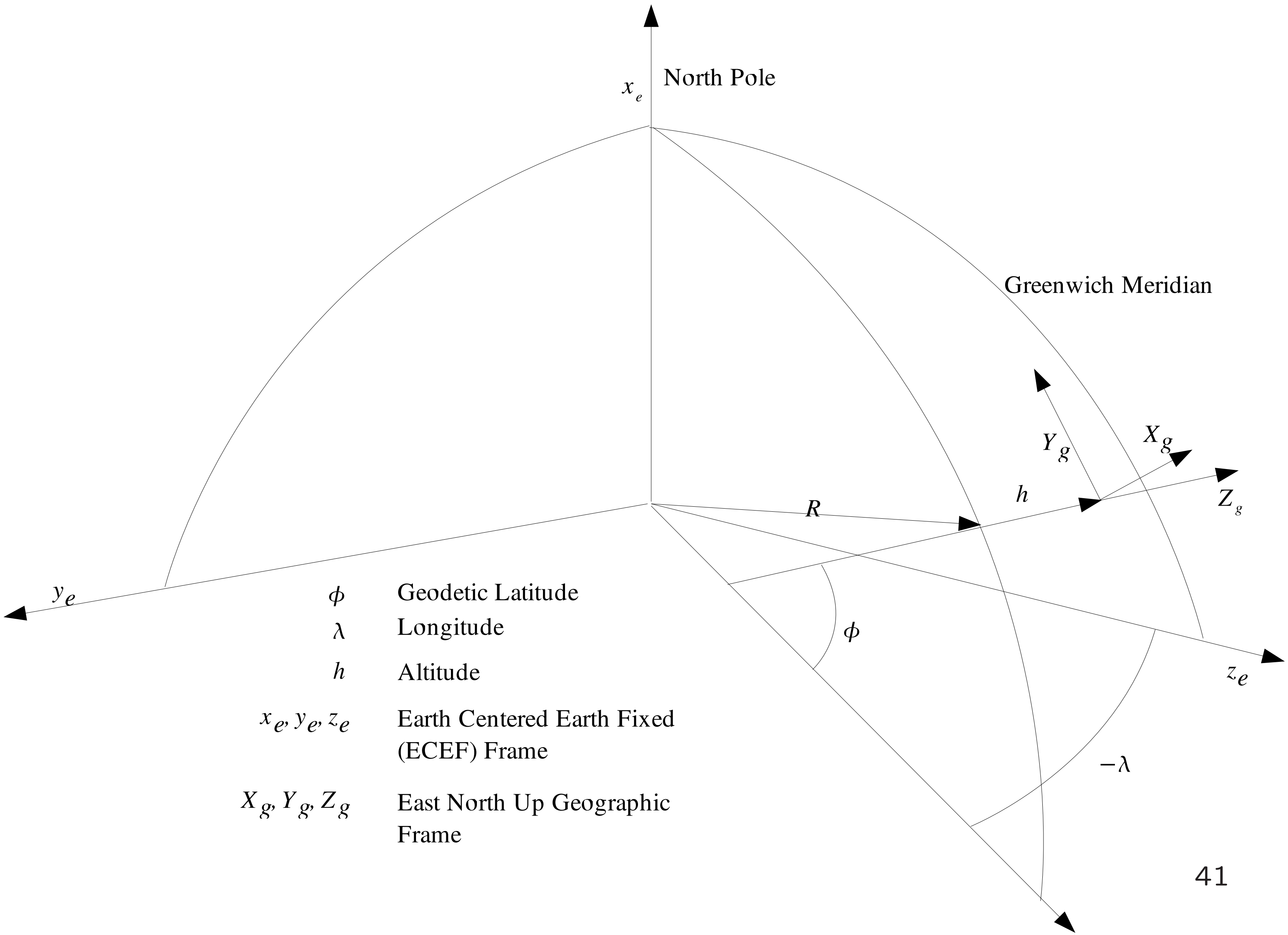
- Rotating frame, i.e., non-inertial

- Cartesian (X, Y, Z in meters)

- Spherical (latitude ϕ, longitude λ, altitude h)
  - geodetic

Local Level

- Tangent to earth's surface but translated to center of earth.
- Vertical Z-axis is normal to earth's surface
- X, Y-axes are level
- Several kinds: Y north-slaved or wander azimuth
- Y makes wander angle α with respect to true north (ENU=East, North, Up).


x_e
North Pole
Greenwich Meridian
Y_g
X_g
h
Z_g
R
y_e
ϕ
z_e
$-\lambda$
ϕ Geodetic Latitude
λ Longitude
h Altitude
x_e, y_e, z_e Earth Centered Earth Fixed (ECEF) Frame
X_g, Y_g, Z_g East North Up Geographic Frame

Body Frame

- Attached to vehicle

- For aircraft:

  - X out the nose

  - Y out the right wing

  - Z down

Coordinate Frame Conversion

- All frames are earth-centered, hence differ only by rotation
- Rotation represented by direction cosine matrices (DCM)
- C_A^B = DCM transformation from frame A to frame B

Coordinate Frame Conversion

- "Direction Cosine" comes from dot product:

$$\langle x, y \rangle = x \cdot y = \|x\| \, \|y\| \cos\theta_{xy}$$

  where θ_{xy} is the angle between the two vectors x and y.

- If x and y are unit vectors, then $\|x\| = \|y\| = 1$ and we have:

$$\langle x, y \rangle = x \cdot y = \cos\theta_{xy}$$

Coordinate Frame Conversion

- If $U = \{u_1, u_2, u_3\}$ and $V = \{v_1, v_2, v_3\}$ are two sets of orthonormal basis vectors, then the DCM relating one frame to the other is given by:

$$C_U^V = \begin{bmatrix} \langle u_1, v_1 \rangle & \langle u_1, v_2 \rangle & \langle u_1, v_3 \rangle \\ \langle u_2, v_1 \rangle & \langle u_2, v_1 \rangle & \langle u_2, v_3 \rangle \\ \langle u_3, v_1 \rangle & \langle u_3, v_2 \rangle & \langle u_3, v_3 \rangle \end{bmatrix}$$
$$= \begin{bmatrix} \cos\theta_{11} & \cos\theta_{12} & \cos\theta_{13} \\ \cos\theta_{21} & \cos\theta_{22} & \cos\theta_{23} \\ \cos\theta_{31} & \cos\theta_{32} & \cos\theta_{33} \end{bmatrix}$$

Coordinate Frame Conversion

- DCMs are orthogonal matrices $\Rightarrow \left[C_A^B\right]^{-1} = \left[C_A^B\right]^\mathsf{T}$

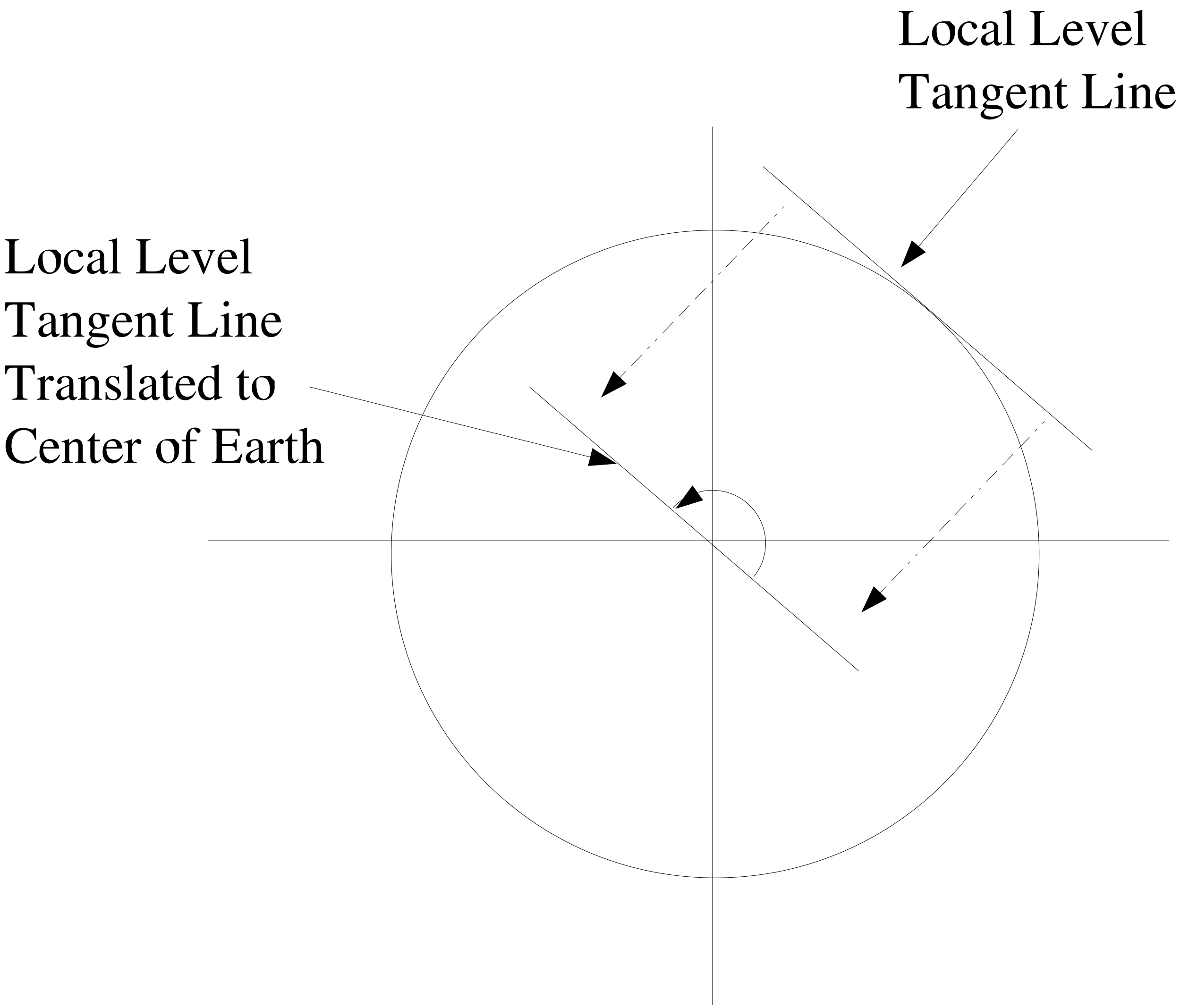
- Third row (or column) is vector cross product of other two $\Rightarrow$ only six independent parameters

Coordinate Frame Conversion: Most important DCMs

- Body to Local Level: C_B^L
  - sometimes called Body to Nav and notated C_B^N
  - yields attitude info (Euler angles for roll, pitch, yaw)
- Local Level to Earth: C_L^E
  - yields position info and used to transform earth rate to local level

Coordinate Frame Conversion: Most important DCMs

- Visualize C_L^E's determination of position as follows:
  - As vehicle moves, local level frame changes its orienation with respect to earth's center in order to maintain local level with respect to earth's surface
  - If you think of local level frame as attached to earth's surface, you get distracted by the displacement of the frame
  - If you move local level frame to center of earth, the displacement disappears and all that remains is the rotations


Local Level
Tangent Line
Local Level
Tangent Line
Translated to
Center of Earth

Strapdown Navigation: Strapdown vs. Gimbaled

- Gimbaled INS:
  - Gyro stabilized platform
  - Gimbal assembly and torque motors
  - Resolvers, slip rings, bearings, etc. to control platform
  - Goal: Maintain stable, locally level or inertial reference frame

Strapdown Navigation: Strapdown vs. Gimbaled

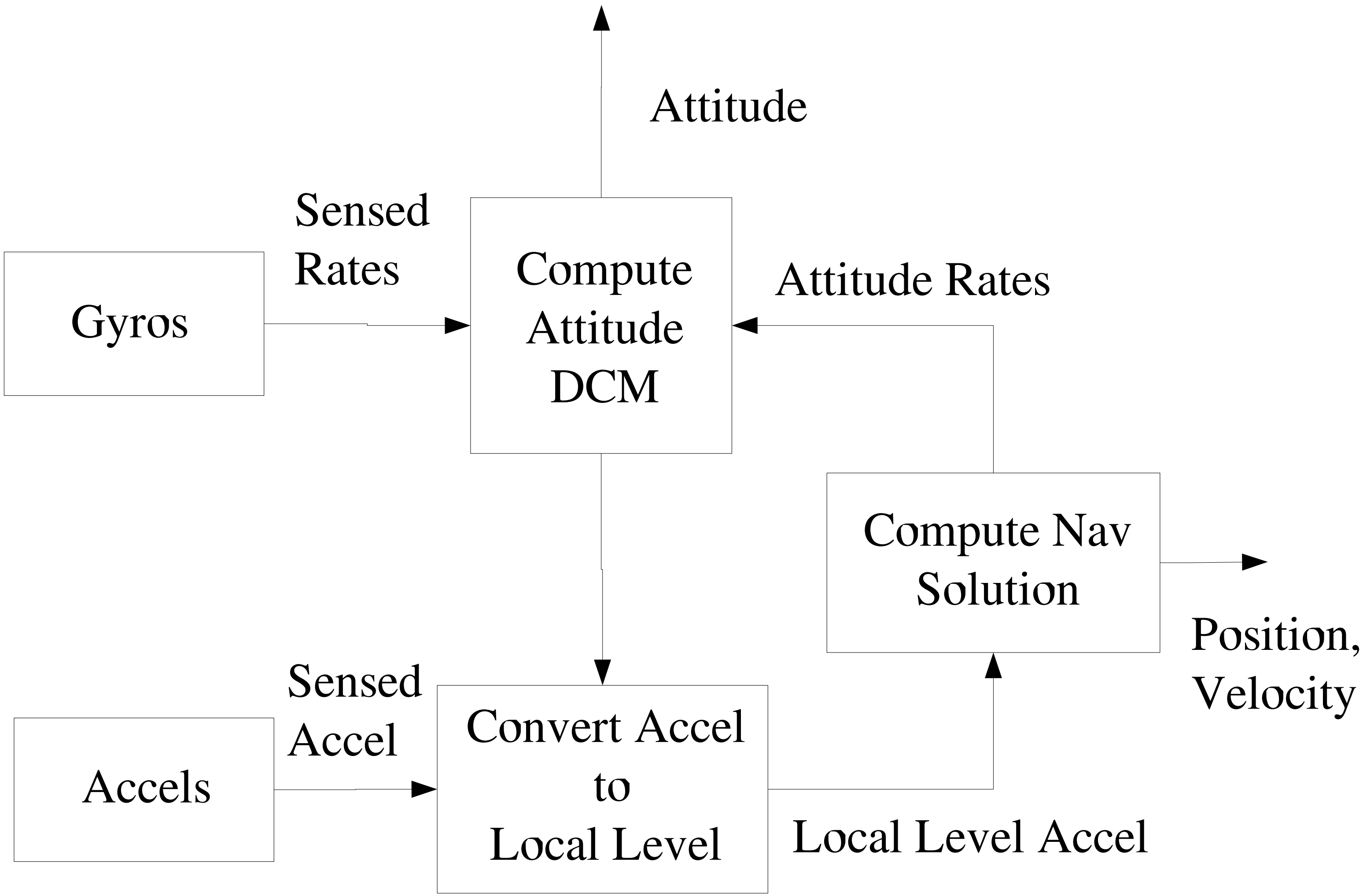
- Gimbaled INS Issues
  - Reliability and Cost: Lots of moving parts!
  - Body rate and acceleration information not directly available
  - Rate signals are generally noisy and have low bandwidth

Strapdown Navigation: Strapdown vs. Gimbaled

- Strapdown INS:
  - Inertial instruments strapped to body frame
  - Requires local level accelerations to be computed using direction cosine matrix C_B^L
  - C_B^L derived from gyro-sensed body rates
  - Strapdown systems can be implemented with either rate gyros or attitude gyros

Strapdown Navigation: Strapdown vs. Gimbaled

- Strapdown INS Advantages
  - Immediate detection of body rates and accelerations
  - Fewer moving parts
  - Potential redundancy advantages
- Gimbaled systems are nearly obsolete



Strapdown INS Block Diagram

Basic Strapdown Navigation Equations

- Need to compute attitude, velocity, and position
- I.e., need to compute C_B^L and C_L^E
- These DCMs are dynamic quantities
- C_B^L determines attitude and can be rapidly varying
- C_L^E determines position and is slowly varying (relative to attitude dynamics)

Strapdown Navigation Equations

- Three differential equations
  - Equation for body to local level transformation C_B^L
  - Equation for velocity (uses C_B^L)
  - Equation for local level to earth for position DCM C_L^E

Strapdown Navigation Equations

- Two Algebraic Equations
  - Equation for converting velocity to transport rate (to be explained)
  - Equation for converting earth rate to local level coordinates

Strapdown Navigation Equations
The Three Differential Equations

(7) $$\dot{C}_B^L = C_B^L \Omega_{IB}^B - \left(\Omega_{IE}^L + \Omega_{EL}^L\right) C_B^L$$

(8) $$\dot{v}^L = C_B^L a_{\mathsf{sf}}^B + g' - \left(2\Omega_{IE}^L + \Omega_{EL}^L\right) v^L$$

(9) $$\dot{C}_L^E = C_L^E \Omega_{EL}^L$$

Strapdown Navigation Equations
The Two Algebraic Equations

(10) $$\omega_{EL} = \frac{1}{r} u_r^L \times v^L$$

(11) $$\omega_{IE}^L = \left[C_L^E \right]^{\mathsf{T}} \omega_{IE}^E$$

Strapdown Navigation Equations
The C_B^L Equation (7)

$$\dot{C}_B^L = C_B^L \Omega_{IB}^B - \left(\Omega_{IE}^L + \Omega_{EL}^L\right) C_B^L$$

- All the Ω matrices represent rotation vector cross products in skew-symmetric matrix format
  - Ω_{IB}^B is rotational rate of aircraft in body coordinates
  - Ω_{IE}^L is earth rate in local level coordinates
  - Ω_{EL}^L is transport rate in local level coordinates

Strapdown Navigation Equations: Detour into Cross Product Matrices

- The cross prouct of $\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$ with any other vector v is the same as multiplying v on the left by $\Omega = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix}$

- $\omega \times v = \Omega v$

Strapdown Navigation Equations
The C_B^L Equation (7)

$$\dot{C}_B^L = C_B^L \Omega_{IB}^B - \left(\Omega_{IE}^L + \Omega_{EL}^L\right) C_B^L$$

- Transport rate is the rotational rate of the aircraft due to motion over the surface of the earth. Is a function of velocity.

- Ω_{IB}^B is total inertial angular rate sensed by gyros and includes effects of transport and earth rates.

- Hence transport and earth rate effects have to be subtracted off.

Strapdown Navigation Equations

The Strapdown Velocity Equation (8)

$$\dot{v}^L = C_B^L a_{\mathsf{sf}}^B + g' - \left(2\Omega_{IE}^L + \Omega_{EL}^L\right) v^L$$

- Sometimes referred to as the velocity rate equation

- a_{sf}^B is specific force sensed by accelerometers in body coordinates (because the accels are strapped to the body – hence the term strapdown)

Strapdown Navigation Equations

The Transport Rate Equation (9)

$$\dot{C}_L^E = C_L^E \Omega_{EL}^L$$

- Connected to velocity rate equation by the first algebraic equation (10):

$$\omega_{EL} = \frac{1}{r} u_r^L \times v^L$$

- Eq. (9) Sometimes referred to as position rate equation, because $dcmLE$ directly implies the vehicle position

Strapdown Navigation Equations
Position Rate / Transport Rate Equations (Geographic)

- North transport rate:

(12) $$\rho_N = \frac{v_e}{R_e}\left[1 - \frac{h}{R_e} - e\,\mathsf{sin}^2\,\phi\right]$$

- East transport rate:

(13) $$\rho_E = -\frac{v_e}{R_e}\left[1 - \frac{h}{R_e} - e\left(1 - 3\,\mathsf{cos}^2\,\phi\right)\right]$$

where e is ellipticity (approx. $\frac{1}{298}$) and R_e is mean equatorial earth radius

Strapdown Navigation Equations
Position Rate / Transport Rate Equations (Geographic)

- Latitude Rate:

(14) $$\dot{\phi} = -\rho_e$$

- Longitude Rate:

(15) $$\dot{\lambda} = \rho_n \sec\phi$$

Strapdown Navigation Equations

- Geographic (north-slaved) mechanization has singularities at poles
- Two other approaches to local level coordinates:
  - Wander azimuth: vertical component of transport rate ρ_z= vertical component of earth rate
  - Free Azimuth: vertical component of transport rate $\rho_z = 0$

Strapdown Navigation Equations
Position Rate / Transport Rate Equations (Wander and Free Azimuth)

- X Transport Rate:

$$(16)\quad \rho_x = -\frac{v_y}{R_e}\left[1 - \frac{h}{R_e} - e\left(1 - 3d_{22}^2 - d_{21}^2\right)\right] - \frac{v_x}{R_e}\left(2d_{21}d_{22}\right)e$$

- Y Transport Rate:

$$(17)\quad \rho_y = \frac{v_x}{R_e}\left[1 - \frac{h}{R_e} - e\left(1 - 3d_{21}^2 - d_{22}^2\right)\right] + \frac{v_y}{R_e}\left(2d_{21}d_{22}\right)e$$

Strapdown Navigation Equations
Position Rate / Transport Rate Equations (Wander and Free Azimuth)

- Azimuth Rate:

$$\rho_z = \begin{cases} -\Omega_e d_{23}, & \text{if free azimuth;} \\ 0, & \text{if wander azimuth} \end{cases}$$

- In equations above, the matrix $\left[d_{ij}\right]$ is the direction cosine matrix C_L^E from local level (L) frame to earth centered earth fixed (E) frame.

Exercises

1. Compute direction cosine matrix C_U^V, where $U = \{(1,0,0),(0,1,0),(0,0,1)$ and $V = \{(-1,0,0),(0,1,0),(0,0,-1)\}$

2. Suppose you are starting at the equator and travelling due north, and your velocity is 5 m/sec. Compute the transport rates ρ_n and ρ_e associated with this velocity and heading. You may use the following constants: $\Omega_e = 7.2921151467 \times 10^{-5}$ rad/sec and $R_e = 6,378,137$ meters

Day 3, Segment 2 (Continued)
Inertial Error Dynamics
Topics

- Imperfect Inertial Instruments
- From Strapdown Equations to Error Equations
- Initialization: Alignment

Imperfect Inertial Instruments

- Accelerometers and Gyros have inherent imperfections
- Bias, scale factor, g^2 terms, noise,
- Inertial sensor clusters also have errors: misalignments
- Many error sources are calibrated out using rate tables during ATP
- Residual effects will influence strapdown nav equations

Imperfect Inertial Instruments

- Strapdown nav equations very sensitive to gyro errors

- Gyro biases yield erroneous C_B^L matrix which in turn causes erroneous accelerations to be computed in local level frame

- Thus, gyro errors affect not only attitude, but also position and velocity

Simple Inertial Error Models

- Gyros

$$\epsilon\omega_x = \Delta B_{gx} + \Delta K_{gx}\omega_x + \theta_{gxy}\omega_y + \theta_{gxz}\omega_z$$

$$\epsilon\omega_y = \Delta B_{gy} + \Delta K_{gy}\omega_y + \theta_{gyz}\omega_z + \theta_{gyx}\omega_x$$

$$\epsilon\omega_z = \Delta B_{gz} + \Delta K_{gz}\omega_z + \theta_{gzx}\omega_x + \theta_{gzy}\omega_y$$

where ΔB_{gi} is the i^{th} gyro bias, ΔK_{gi} the i^{th} gyro scale factor, and θ_{gij} is the misalignment of the i^{th} gyro axis toward the j^{th} gyro axis.

Simple Inertial Error Models

- Accelerometers

$$\epsilon a_x = \Delta B_{ax} + \Delta K_{ax} a_x + \theta_{axy} a_y + \theta_{axz} a_z$$

$$\epsilon a_y = \Delta B_{ay} + \Delta K_{ay} a_y + \theta_{ayz} a_z + \theta_{ayx} a_x$$

$$\epsilon a_z = \Delta B_{az} + \Delta K_{az} a_z + \theta_{azx} a_x + \theta_{azy} a_y$$

where ΔB_{ai} is the i^{th} accelerometer bias, ΔK_{ai} the i^{th} accelerometer scale factor, and θ_{aij} is the misalignment of the i^{th} accelerometer axis toward the j^{th} accelerometer axis.

Bias Models

- Bias in both types of instruments can be further broken down into several components:
  - fixed bias – constant from turn-on to turn-on – should be calibrated out
  - bias stability – varies from turn-on to turn-on – can't be calibrated out, but can be characterized in terms of variance – can model as random constant
  - bias drift – usually modeled as random walk

Scale Factor Models

- Scale factor can also be modeled as a sum of three components
- Often in nav filters, scale factor residual is modeled as a random constant
- Important to keep in mind that the contribution of scale factor error is a function of input (rate or acceleration)

Higher Order Models

- In expanding the output error as a Taylor series function of input, bias, scale factor, and misalignment show up as zero and first order terms.

- Higher order terms are also present, especially second order terms.

- In some calibration and alignment filters, such higher order effects are modelled

- SNU-84-1 includes error budgets associated with all the combined, higher order effects.

From Strapdown Equations to Error Equations

- Nav equations require perfect inertial instruments

- Instruments are not perfect

- Inertial instrument error sources (bias, scale factor errors, etc.) generate erroneous rate and acceleration inputs to the nav equations

- Need to figure out the effects on the outputs of these equations

Strapdown Error Equations

- Nature of Errors
- Error Operator
- Error Equations

Strapdown Error Equations: Nature of Errors

- Errors propagate according to derived equations
- Errors are generally small, so approximating assumptions help simplify the math
  - linearize otherwise non-linear dynamics
  - small angle assumptions: $\sin\theta \approx \theta$

Strapdown Error Equations: Error Operator

If A is any quantity, we define error operator as follows:

$$\delta A = \widehat{A} - A, \tag{18}$$

where A is the true quantity of interest and $\widehat{A}$ is an apparent quantity. The latter could be a measured value, and estimated value, or some computed value. This means that

$$\widehat{A} = A + \delta A$$

i.e., the apparent value is the true value plus some error.

Strapdown Error Equations: Error Operator

Error operator satisfies:

$$\delta(AB) = \delta A \cdot B + A \cdot \delta B \tag{19}$$

(Just like differentiation)

Strapdown Error Equations

Applying these to the three differential equations, yields three error equations:

(20)

$$\delta\dot{C}_B^L = \delta C_B^L \Omega_{IB}^B + C_B^L \delta\Omega_{IB}^B - \left(\delta\Omega_{IE}^L + \delta\Omega_{EL}^L\right) C_B^L - \left(\Omega_{IE}^L + \Omega_{EL}^L\right) \delta C_B^L$$

(21)

$$\delta\dot{v}^L = \delta C_B^L a_{\mathsf{sf}}^B + C_B^L \delta a_{\mathsf{sf}}^B g' - \left(2\delta\Omega_{IE}^L + \delta\Omega_{EL}^L\right) v^L - \left(2\Omega_{IE}^L + \Omega_{EL}^L\right) \delta v^L$$

(22)

$$\delta\dot{C}_L^E = \delta C_L^E \Omega_{EL}^L + C_L^E \delta\Omega_{EL}^L$$

Introducing Small Angular Rotations for Tilt and Position

- By introducing $\phi^L = \begin{bmatrix} \phi_x^L & \phi_y^L & \psi_z^L \end{bmatrix}^\mathsf{T}$, and a small angular position angle error $e = \begin{bmatrix} e_x^L & e_y^L & e_z^L \end{bmatrix}^\mathsf{T}$, the attitude error equation (20) can be rewritten as:

(23) $$\dot{\phi}^L = -\left(\Omega_{IE}^L + \Omega_{EL}^L\right)\phi^L - C_B^L \delta\omega_{IB}^B + \Omega_{IE}^L e^L + \delta\omega_{EL}^L$$

where $\delta\omega_{EL}^L = \frac{1}{R_e}\left(u_R^L \times \delta v^L\right) - \frac{\delta h}{R_e^2}\left(u_R^L \times v^L\right) + \delta\rho_R u_R^L$

Introducing Small Angular Rotations for Tilt and Position

- Similarly, the velocity equation can be rewritten:

$$\delta\dot{v}^L = C_B^L \delta a_{\mathsf{sf}}^B + a_{\mathsf{sf}}^B \times \phi^L - \left(2\omega_{IE}^L + \omega_{EL}^L\right) \times \delta v^L + \\ - \left(\delta\omega_{EL}^L - 2e^L \times \omega_{IE}^L\right) \times v^L + \frac{2g\delta h}{R_e} u_R^L + \delta g^L$$

  while the equation for the position angle error e^L itself is

$$\dot{e}^L = -\Omega_{EL}^L e^L + \delta\omega_{EL}^L \qquad (24)$$

Further Simplifications Using Telescope Pointing Angle Error

- $\psi^L = \phi^L - e^L$

- Leads to:

$$\Delta \dot{R}^L = \delta v^L - \Omega^L_{EL} \Delta R^L \tag{25}$$

and

$$\delta \dot{v}^L = C^L_B \delta a^B_{\mathsf{sf}} + a^B_{\mathsf{sf}} \times \psi^L - \frac{g}{R_e} \Delta R^L_H - \left(2\omega^L_{IE} + \omega^L_{EL}\right) \times \delta v^L +$$

$$- \left(\delta\omega^L_{EL} - 2e^L \times \omega^L_{IE}\right) \times v^L + \frac{2g\delta h}{R_e} u^L_R + \delta g^L \tag{26}$$

Further Simplifications Using Telescope Pointing Angle Error

- Finally, the equations for the telescope pointing angle errors are:

$$\dot{\psi}^L = -\left(\Omega^L_{IE} + \Omega^L_{EL}\right)\psi^L - C^L_B \delta\omega^B_{IB} \tag{27}$$

Strapdown Error Equations

- Three error equations yield three vector quantities of interest
  - position error vector (3) ΔR derived from δC_L^E
  - velocity error vector (3) ΔV^L (or δv^L)
  - attitude error vector (3) ϕ_x, ϕ_y, ψ_z from δC_B^L

Strapdown Equations

- See Matlab / Simulink Models
  - Strapdown Navigation Equations: stravnav.mdl
  - Strapdown Navigation Error Equations: straperr.mdl

Initialization: Alignment

- Coarse Leveling
  - Gimbal system limited by platform torquing rates
  - Strapdown systems have no real rate limits
  - Time average of accel outputs scaled by $1/g$
  - Yields third row of DCM to begin fine alignment
- Fine Alignment
- Alignment filter aids in further leveling

Day 3, Segment 3
The Basic 9-State Kalman Filter
Topics

- Whole State Filters vs. Error State Filters

- The Canonical Nine States

- INS Error State Dynamics and Transition Matrix

Whole State vs. Error State Filters

- Whole state filters ...
  - model whole position, velocity, attitude
  - deal with large and rapidly varying quantities
  - are highly non-linear

Whole State vs. Error State Filters

- Error state filters ...
  - remove some of the linearities by differencing outputs from aiding sensors with corresponding quantities from INS
  - exhibit slowly varying dynamics: easier to linearize
  - have smaller quantities that allow us to take advantage of small angle assmptions

The Canonical Nine States

- 3 position errors: $\Delta R_x, \Delta R_y, \Delta R_z$ in local level coordinates
- 3 velocity errors: $\Delta V_x, \Delta V_y, \Delta V_z$ in local level coordinates
- 3 attitude errors: ϕ_x, ϕ_y, ψ_z
  - ϕ_x, ϕ_y known as tilt errors
  - ψ_z known as azimuth error

Exercises

1. Rewrite equation (20)

$$\delta \dot{C}_B^L = \delta C_B^L \Omega_{IB}^B + C_B^L \delta \Omega_{IB}^B - \left(\delta \Omega_{IE}^L + \delta \Omega_{EL}^L \right) C_B^L - \left(\Omega_{IE}^L + \Omega_{EL}^L \right) \delta C_B^L$$

to show the terms in the canonical order with position error, velocity error, and then attitude error.

2. Determine the 7$^{\text{th}}$ row of the system matrix F for the canonical 9-state error model. Use the cross-product operator matrix as required, e.g., $\Omega_{IE}^L = \left[\ \omega_{IE}^L \times\ \right] = \begin{bmatrix} 0 & \left[\omega_{IE}^L\right]_z & -\left[\omega_{IE}^L\right]_y \\ -\left[\omega_{IE}^L\right]_z & 0 & \left[\omega_{IE}^L\right]_x \\ \left[\omega_{IE}^L\right]_y & -\left[\omega_{IE}^L\right]_x & 0 \end{bmatrix}$.

Introduce simplifying notation! E.g., set $\alpha = \omega_{IE}^L$ Hint: Use telescope pointint angle error equation (27)

Day 3, Segment 4
Challenges of INS Filters
Topics

- Nine States Are Not Enough

- Not Constant Coefficient

- Non-Linearities: Extended Kalman Filter

- Observability

Nine States Are Not Enough

- The canoncial 9 states only characterize the inertial error dynamics

- They treat of the inner coupling between inertial error sources

- However, they don't help characterize dynamics of aiding sensor errors

Nine States Are Not Enough

- They don't help estimate aiding states such as Doppler bias and boresight errors

- Or GPS receiver clock errors

- They also don't help us to estimate inertial instrument drifts such as bias drift or scale factor errors

Not Constant Coefficient

- Notice that the F matrix that has emerged (implicitly) from the 9-state inertial error dynamics is not time invariant

- It is highly dependent on trajectory data such as acceleration and angular rate profiles

- These are functions of time

- Position dependent items such as latitude are also involved

Non-Linearities: Extended Kalman Filter

- When non-linearities are present, the F matrix emerges as Jacobian of non-linear system function f: $F(t) = \frac{\partial f}{\partial x}$

- If Jacobian is evaluated at a nominal value of the trajectory, we call this a linearized filter

- If Jacobian is evaluated at the previous optimal estimate, we call it an Extended Kalman filter

Observability

- Structural observability is concerned with how H maps measurements into states

- Will making measurements as governed by H actually enable us to estimate all of the states in the state vector?

Observability

- Observability means there is a linear path from the measurements to states

- Structural observability determined by Φ and H:

If H is $1 \times n$, where n is the size of the state vector, then the state x_0 is said to be **observable** from n scalar measurements $\{z_0, \ldots, z_{n-1}\}$ if the block matrix

$$\left[H^\top \quad \Phi^\top H^\top \quad \cdots \quad (\Phi^\top)^{n-1} H^\top \right]$$

is of rank n

Observability

- Observability challenge for INS filters is usually a matter of choosing right navaids
- And also choosing the right measurement scheduling
- Also, certain kinds of divergence can be the result of poor observability (more on this later)

Exercises

- Use Eq. (25)

$$\Delta \dot{R}^L = \delta v^L - \Omega^L_{EL} \Delta R^L,$$

write the first three rows of the system dynamics matrix F in the continuous Kalman formulation:

$$\dot{x} = Fx + w$$