GPS/관성센서 통합에 의한 측위 및 응용

LEC1 INS FUNDAMENTALS

2005/7/14

이형근 (hyknlee@hau.ac.kr)
Navigation?

- To accurately determine position and velocity relative to a known reference

- To plan and execute the maneuvers necessary to move between desired locations
Classification of Nav. Systems

- Radio Navigation Systems
  - GPS, Galileo
  - VOR/DME, ILS, TACAN, Loran

- Dead reckoning systems
  - Inertial Navigation Systems (INS)
  - Gyro/tachometer systems: Land applications
Dead Reckoning

- **Geometry**

- **Mechanization**

![Diagram showing geometry and mechanization of dead reckoning with relevant equations and components like speed indicator, azimuth indicator, and computer.]
2-D Navigation

- Instantaneous X position
- Instantaneous Y position
- Direction of travel
- Instantaneous heading of vehicle
- Track
- Train
Simple Mechanism of INS

Vehicle Frame

Gimbal

Inertial Stabilization Means

Stable Platform

X axis Force Sensor \( \frac{1}{m_x} \)

\[ \int \]

\( V_{ox} \)

\( R_{ox} \)

\( R_x(t) \)

Y axis Force Sensor \( \frac{1}{m_y} \)

\[ \int \]

\( V_{oy} \)

\( R_{oy} \)

\( R_y(t) \)

Z axis Force Sensor \( \frac{1}{m_z} \)

\[ \int \]

\( V_{oz} \)

\( R_{oz} \)

\( R_z(t) \)
Gimballed INS vs. Strapdown INS

\[
\vec{v} = \int_0^t \vec{a} \, dt + \vec{v}_0
\]

\[
\vec{R} = \int_0^t \vec{v} \, dt + \vec{R}_0
\]
Strapdown INS (SDINS)

가속도계

IMU

자이로

속도

중력 계산

위치

\[ L = \frac{V_N}{R_m + h} \]
\[ i = \frac{V_E}{(R_m + h) \cos \theta} \]
\[ h = -V_D \]

\[ \dot{C}_b^n = C_b^n \Omega_{nb}^b \]

\[ \dot{C}_b^n = \int dt \]

\[ \omega_{in}^b \]

\[ \omega_{ib}^b + \omega_{nb}^b \]

\[ \omega_{en}^n \]
**Historical Development (1)**

- **Earliest times**
  - *Navigation by observation*
  - *Polynesians cross the Pacific Ocean about two millennia ago*
- **13th century**
  - *Compass*, which could be used irrespective of visibility
  - *Sextant*, which enabled position fixes to be made accurately on land
- **17th century**
  - *Isaac Newton defined the laws of mechanics and gravitation*, which are the fundamental principles of inertial navigation
- **Early 18th century**
  - A stabilized sextant by Serson
  - An accurate chronometer by Harrison
Historical Development(2)

- **19th century**
  - Gyroscopic effect in 1852 by Foucault
  - Rotational motion of the Earth and the demonstration of rotational dynamics by Bohneberger, Johnson, and Lemarle
  - Ringing of hollow cylinders, a phenomenon later applied to solid state gyroscopes, in 1890 by Prof. G.H.Bryan

- **20th century**
  - Gyrocompass
  - Schuler tuning: \(2\pi\sqrt{R/g} \approx 84\text{min},\) undamped natural period
  - Application of the gyroscopic effect to control and guidance by Sperry brother
  - Stable platform for fire control systems for guns on ship in the 1920s
  - Demonstration of the principles of inertial guidance in the V2 rocket during World War II
  - Strapdown technique for navigation in 1949
Historical Development (3)

- 20th century (cont’d)
  - More accurate sensors in the 1950s: 15deg/hr → 0.01deg/hr by Prof. Charles Stark Draper, MIT
  - INS using the so-called stable platform technology became standard equipment in military aircraft, ships and submarines during the 1960s
  - Ring laser gyroscope
  - Ballistic missile and exploration of space
  - In the last two decades, the application of the microcomputer
  - Development of gyroscopes with large dynamic ranges enabling the strapdown principle to be realized

- Modern-day inertial Navigation system
  - Diverse applications: robotics, racing or high performance motor car and for surveying underground well and pipelines
  - Call for navigation systems having a very broad range of performance capabilities
Strapdown Sensor and Applications

GYROSCOPE ERROR (°/HOUR)


PLATFORM SOLUTION NECESSARY
STRAPDOWN RATE INTEGRATING GYROSCOPES
CONVENTIONAL GYROSCOPES
STRAPDOWN SOLUTION POSSIBLE

SUBMARINE INS
SHIP INS
AIRCRAFT INS
AIDED INS
AIRCRAFT AUTOPILOT
STRATEGIC MISSILE INS
TACTICAL MISSILE INS
SPACECRAFT INS
AIRCRAFT AHRS
TACTICAL MISSILE GUIDANCE
LAND VEHICLES INS
UAV GUIDANCE
TORPEDO GUIDANCE

STABILISATION APPLICATIONS
Earth
Earth surface, Geoid, and Ellipsoid

- Geoid is defined as level surface of gravity field with best fit to mean sea level. (Maximum difference between geoid and mean sea level is about 1 m)

- Ellipsoid defines an approximated surface to simplify geometries and computations regarding the Earth.
**WGS-84 Ellipsoid (Earth Model)**

- **semi-major Axis**: \( a = 6378137 \) (m)
- **semi-minor Axis**: \( b = 6356752.3142 \) (m)
- **flatness**: \( f = (a-b)/a = 1/298.257223563 = 0.00335281066475 \)
- **eccentricity**: \( e = [f(2-f)]^{1/2} = 0.08181919084262 \)
Geodetic and Geocentric Latitudes

\[ e = \sqrt{\frac{a^2 - b^2}{a^2}} = \sqrt{1 - f^2} \text{ : eccentricity} \]

\[ f = \frac{b}{a} \text{ : flatness} \]

\[ (1) \quad y' = \frac{b^2}{a^2} x_p \]

\[ (2) \quad y = \frac{b^2}{a^2} x + (1 - \frac{a^2}{b^2}) x_p \]

\[ (3) \quad a^2 x_p x = (a^2 - 1) x_p \Rightarrow \frac{a^2}{b^2} - 1 \frac{x_p}{x} = (a^2 - 1) \]

\[ \Rightarrow x = \frac{b^2}{a^2} (a^2 - 1) x_p \Rightarrow x = e^2 x_p \]

\[ (4) \quad r \sin(D) = e^2 x_p \sin(L) \]

\[ (5) \quad D = \sin^{-1} \left( \frac{e^2 x_p \sin(L)}{r} \right) \text{ : latitude difference} \]
**Gravity Model**

\[
g^n = \begin{bmatrix} 0 \\ 0 \\ g_z(L, h) \end{bmatrix} + g_{anomaly}
\]

\[
g_z(L, h) = g_z(L) - [3.0877 \times 10^{-6} - 0.0044 \times 10^{-6} \sin^2(L)]h + 0.072 \times 10^{-12} h^2 \quad (m/s^2)
\]

\[
g_z(L) = g_0[1 + 0.0053024 \sin^2(L) - 0.0000058 \sin^2(2L)] \quad (m/s^2)
\]

\[
g_0 = 9.780327 \quad (m/s^2)
\]

* \( h \) is measured in \( m \).

* Magnitude of \( g_{anomaly} := \begin{bmatrix} \xi_{anomaly} & -\eta_{anomaly} & \delta_{anomaly} \end{bmatrix}^T \) caused by the gravity deflection is typically less than \( 10^{-5} g_0 \).
Meridian Radius of Curvature

\[ R_L = \frac{R_0 (1-e^2)}{(1-e^2 \sin^2 L)^{3/2}} \]

- meridian radius of curvature at a given latitude
**Transverse Radius of Curvature**

\[
R_t = \frac{R_0}{(1 - e^2 \sin^2 \theta)^{1/2}}
\]

- min \( R_t \cos(L) \) (= 0) occurring at poles
- max \( R_t \cos(L) \) (= \( R_0 \)) occurring at equator

: transverse radius of curvature
Coordinate Systems
Coordinate systems related to INS

Greenwich meridian plane

Equatorial plane
**Coordinate Systems: Inertial Frame (i-frame)**

- Inertial Frame (i-frame) – A reference frame in which Newton’s laws of motion apply. Non-accelerating but may be in uniform linear motion. An orthogonal coordinate system.
Coordinate Systems: ECEF Frame (e-frame)

- Earth-Centered Earth-Fixed (ECEF) Frames (e-frame) – Its origin fixed to the center of the earth. The axes rotate relative to the inertial frame with a frequency of

\[
\omega_{ie} \approx \frac{1 + 365.25 \text{ cycles}}{365.25 \text{ hr}} \frac{2\pi \text{ rad/cycle}}{3600 \text{ sec/hr}} = 7.292115 \times 10^{-5} \text{ rad/sec}
\]

- because of the daily earth rotation and yearly revolution about the sun.
Locally-Level Frame (n-frame) – The z-axis points toward the interior of the ellipsoid along the ellipsoid normal. The x-axis points toward true north. The y-axis follows the right-handed rule.
**Coordinate Systems: Body Frame (b-frame)**

- **Body Frame (b-frame)** – The origin is usually at the center of gravity of the vehicle of interest. The x-axis is defined in the forward direction. The z-axis is defined pointing to the bottom of the vehicle. The y-axis completes the right-handed orthogonal coordinate system.
Coordinate Transformation
**Transformation about a Single-Axis**

- A non-zero vector fixed in space can be expressed with respect to various frames.
- If we express \( \mathbf{r} \) with respect to the \( i \)-frame and \( e \)-frame, they are summarized as

\[
\begin{bmatrix}
  r^e_x \\
  r^e_y \\
  r^e_z 
\end{bmatrix} = C^e_i \begin{bmatrix}
  r^i_x \\
  r^i_y \\
  r^i_z 
\end{bmatrix}, \quad
\begin{bmatrix}
  r^e_x \\
  r^e_y \\
  r^e_z 
\end{bmatrix} = \begin{bmatrix}
  \cos \psi & \sin \psi & 0 \\
  -\sin \psi & \cos \psi & 0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  r^i_x \\
  r^i_y \\
  r^i_z 
\end{bmatrix}
\]

- \( r^e_x = r^i_x \cos \psi + r^i_y \sin \psi \)
- \( r^e_y = -r^i_x \sin \psi + r^i_y \cos \psi \)
- \( r^e_z = r^i_z \)
**Successive Rotations: Euler Angles**

- Euler angles from n-frame to b-frame:
  1. $\psi$ - rotation about $z$
  2. $\theta$ - rotation about $y'$
  3. $\phi$ - rotation about $x''$

![Diagram of Euler angles](image)
Thus, the total transformation matrix can be decomposed by three elementary transformation matrices as follows.

\[ C_n^b = C_n^{''''} C_n^{'''} C_n' \]

where

\[ C_n^{'''}(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

\[ C_n^{'''}(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \]

\[ C_n^{'''}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \]

Therefore,

\[ C_n^b = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \]
Properties of the Transformation Matrix

(1) Inverse transformation

\[ C_m^i = (C_i^m)^{-1} \]

(2) Orthonormality

\[ (C_i^m)^T C_i^m = I \]

(3) Transpose matrix equals inverse matrix by (1) and (2)

\[ (C_i^m)^T = (C_i^m)^{-1} = C_m^i \]
Utilization of Transformation Matrix in SDINS

\[ C^b_n = (C^n_b)^T \]

- b-frame
  - roll \( \phi \)
  - pitch \( \theta \)
  - yaw (heading) \( \psi \)

- n-frame

BODY MOUNTED ACCELEROMETERS

BODY MOUNTED GYROSCOPES

ATTITUDE COMPUTER

RESOLUTION OF SPECIFIC FORCE MEASUREMENTS

\[ f^b \]

\[ f^n \]

NAVIGATION COMPUTER

GRAVITY COMPUTER

CORIOLIS CORRECTION

\[ g^n \]

\[ \alpha_n^L + \alpha_v^n \]

INITIAL ESTIMATES OF ATTITUDE

INITIAL ESTIMATES OF VELOCITY & POSITION

POSITION INFORMATION

POSITION & VELOCITY (\( V^n_w \)) ESTIMATES

http://eaa.hang@ong.ac.kr/nisl/
Attitude Differential Equation
**Attitude Differential Equation**

\[
\dot{C}_b^n = C_b^n \left\langle \omega_{nb}^b \right\rangle
\]

From the definition of differential equations, it is obvious that

\[
\dot{C}_b^n = \lim_{\Delta t \to 0} \frac{\Delta C_b^n}{\Delta t} = \lim_{\Delta t \to 0} \frac{C_b^n(t + \Delta t) - C_b^n(t)}{\Delta t}.
\]

(1)

\(C_b^n(t + \Delta t)\) can be decomposed as follows where small angle approximation is utilized (i.e., products of small angles are neglected).

\[
C_b^n(t + \Delta t) = C_b^n(t)C(\delta\psi)C(\delta\theta)C(\delta\phi) = C_b^n(t)\left[ I_{3x3} + \delta\mathbf{C} \right]
\]

(2)

where

\[
C(\psi) = \begin{bmatrix}
\cos \delta\psi & \sin \delta\psi & 0 \\
-\sin \delta\psi & \cos \delta\psi & 0 \\
0 & 0 & 1
\end{bmatrix} \cong \begin{bmatrix}
1 & \delta\psi & 0 \\
-\delta\psi & 1 & 0 \\
0 & 0 & 1
\end{bmatrix},
\]

\[
C(\theta) = \begin{bmatrix}
1 & 0 & -\delta\theta \\
0 & 1 & 0 \\
\delta\theta & 0 & 1
\end{bmatrix},
\]

\[
C(\phi) = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & \delta\phi \\
0 & -\delta\phi & 1
\end{bmatrix},
\]

\[
\frac{\delta\phi}{\delta\theta} = \frac{\omega_x}{\omega_y} \Delta t = \omega_{nb}^b \Delta t.
\]

(3)
By combining Eqs. (2) and (3), it is easy to verify that

\[ C(\delta \psi)C'(\delta \theta)C'(\delta \phi) = I_{3 \times 3} + \delta \mathcal{C}, \]

where

\[ \delta \mathcal{C} = \begin{bmatrix} 0 & -\delta \psi & \delta \theta \\ \delta \psi & 0 & -\delta \phi \\ -\delta \theta & \delta \phi & 0 \end{bmatrix} = \begin{bmatrix} \omega^{b}_{nb} \end{bmatrix} \Delta t. \]

\[ \langle \omega \rangle: \ 3 \times 3 \ \text{skew-symmetric matrix based on} \ 3 \times 1 \ \text{vector angle} \ \omega. \]

Substituting Eqs. (2) and (5) to (1) yields

\[ \dot{C}^{n}_{b} = \lim_{\Delta t \to 0} \frac{C^{n}_{b} \begin{bmatrix} \omega^{b}_{nb} \end{bmatrix} \Delta t}{\Delta t} = C^{n}_{b} \begin{bmatrix} \omega^{b}_{nb} \end{bmatrix}. \]
Position and Velocity Differential Equations
Parameterization, Differentiation, and Frames
Vector
- an arrow (rod) consisting of starting and finishing points

Parameterization of a vector w.r.t. a frame
- The same vector can be represented by different parameterizations if reference frames are different.

Differentiation of a vector w.r.t. a frame
- Differentiation of the same vector can result in different vectors if reference frames for differentiation are different. We ride the reference frame for differentiation and watch the changes of the vector.

Differentiation of a vector w.r.t. a frame and its parameterization w.r.t. another frame

\[
\begin{align*}
R^m &= \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} \\
&= x_m I_m + y_m J_m + z_m K_m \\
\frac{dR}{dt}_m &= \begin{bmatrix} \frac{dx_m}{dt} \\ \frac{dy_m}{dt} \\ \frac{dz_m}{dt} \end{bmatrix}_m \\
\left(\frac{dR}{dt}_m\right)^i &= \left(\begin{bmatrix} \frac{dx_m}{dt} \\ \frac{dy_m}{dt} \\ \frac{dz_m}{dt} \end{bmatrix}_m\right)^i
\end{align*}
\]
Differentiation w.r.t. Different Frames

Given

\[ R^i = C^i_m R^m \]  \hspace{1cm} (7)

differentiate

\[ \dot{R}^i = \dot{C}^i_m R^m + C^i_m \dot{R}^m \]  \hspace{1cm} (8)

\[ \dot{R}^m = \begin{bmatrix} \dot{x}_m \\ \dot{y}_m \\ \dot{z}_m \end{bmatrix} = \left( \frac{dR}{dt} \right)_m^m \]  \hspace{1cm} (9)

\[ \dot{R}^i = \left( \frac{dR}{dt} \right)_i^i \]

\[ \dot{C}^i_m = C^i_m \left\langle \omega^m_{im} \right\rangle = C^i_m \left( \omega^m_{im} \times \right) \]
Apply (9) to (8):

$$\left( \begin{bmatrix} \frac{dR}{dt} \end{bmatrix}_i \right)^i = C_m^i \langle \omega_{im}^m \rangle R^m + C_m^i \left( \begin{bmatrix} \frac{dR}{dt} \end{bmatrix}_m \right)^m$$

$$= \langle \omega_{im} \rangle C_m^i R^m + C_m^i \left( \begin{bmatrix} \frac{dR}{dt} \end{bmatrix}_m \right)^m$$

$$= \left( \langle \omega_{im} \rangle R + \left[ \begin{bmatrix} \frac{dR}{dt} \end{bmatrix}_m \right]^i \right)$$

$$= \left( \langle \omega_{im} \rangle R + \left[ \begin{bmatrix} \frac{dR}{dt} \end{bmatrix}_m \right]^i \right)$$

(10)

Restore vector parameterization (i-frame) to vector:

$$\left[ \begin{bmatrix} \frac{dR}{dt} \end{bmatrix}_i \right] = \left[ \begin{bmatrix} \frac{dR}{dt} \end{bmatrix}_m \right] + \langle \omega_{im} \rangle R$$

(11)

**Coriolis Equation**
**Specific Force Equations in a Moving Frame**

For brevity, define the differentiation operators

\[
p_i = \left[ \frac{d}{dt} \right]_i \quad p_e = \left[ \frac{d}{dt} \right]_e \quad p_m = \left[ \frac{d}{dt} \right]_m
\]  

(12)

We already know that

\[
p_i R^m = p_m R^m + \omega_{im} \times R^m \quad \text{(Coriolis equation)}
\]  

(13)

\[
f^m = p_i^2 R^m - G(R)^m \quad \text{(Specific force equation)}
\]  

(14)

The two equations are combined as follows

\[
f^m = p_m p_i R^m + \omega_{im} \times p_i R^m - G(R)^m
\]

\[
\{* \quad p_i R^m = p_e R^m + \omega_{ie} \times R^m \}
\]

\[
= p_m p_e R^m + p_m (\omega_{ie} \times R^m) + \omega_{im} \times (p_e R^m + \omega_{ie} \times R^m) - G(R)^m
\]

\[
= p_m p_e R^m + (p_m \omega_{ie}) \times R^m + \omega_{im} \times (p_m R^m) + \omega_{im} \times (\omega_{ie} \times R^m) - G(R)^m
\]
\[ \begin{align*}
\varepsilon_m p_e R^m + (p_m \omega_{ie}^m) \times (p_m R^m) + \omega_{ie}^m \times (p_m R^m) + \omega_{ie}^m \times (\omega_{me}^m \times R^m) \\
+ \omega_{im}^m \times (p_e R^m) + \omega_{im}^m \times (\omega_{ie}^m \times R^m) - G(R)^m
\end{align*} \]

\[ \begin{align*}
\varepsilon_m = p_m \omega_{ie}^m = p_m \omega_{ie}^m + \omega_{mi}^m \times \omega_{ie}^m = p_m \omega_{ie}^m - \omega_{im}^m \times \omega_{ie}^m
\end{align*} \]

\[ \begin{align*}
\varepsilon_m = p_m \omega_{ie}^m - \omega_{im}^m \times \omega_{ie}^m
\end{align*} \]

\[ \begin{align*}
\omega_{im}^m \times (\omega_{ie}^m \times R^m) = \omega_{ie}^m \times (\omega_{im}^m \times R^m) + (\omega_{im}^m \times \omega_{ie}^m) \times R^m
\end{align*} \]

\[ \begin{align*}
\omega_{im}^m \times (\omega_{ie}^m \times R^m) = \omega_{ie}^m \times (\omega_{im}^m \times R^m) + (\omega_{im}^m \times \omega_{ie}^m) \times R^m
\end{align*} \]

\[ \begin{align*}
\omega_{im}^m \times \omega_{me}^m = \omega_{ie}^m
\end{align*} \]

\[ \begin{align*}
\omega_{im}^m \times (\omega_{im}^m \times R^m) = \omega_{im}^m \times (\omega_{im}^m \times R^m) + (\omega_{im}^m \times \omega_{ie}^m) \times R^m
\end{align*} \]

where the gravity is defined from the gravitational acceleration as follows.

\[ g^m = G(R)^m - \omega_{ie}^m \times \omega_{ie}^m \times R^m \] (16)
Velocity Differential Equation

\[ \dot{V}^n = C_b^b f^b - [2(\omega^{n}_{ie} \times) + (\omega^{n}_{en} \times)]V^n + g^n \] (17)

Set \( m = n \) and define

\[ V^n = \begin{bmatrix} V_\nu \\ V_\theta \\ V_\phi \end{bmatrix} = p_e R^n = \left( \begin{bmatrix} dR \\ dt \end{bmatrix} e \right)^n \] (18)

Then

\[ p_n p_e R^n = \dot{V}^n \] (19)

\[ \dot{V}^n = f^n - [2(\omega^{n}_{ie} \times) + (\omega^{n}_{en} \times)]V^n + g^n \] (20)

where

\[ \omega^{n}_{ie} = \begin{bmatrix} \Omega_N \\ 0 \\ -\Omega_D \end{bmatrix}, \quad \Omega = ||\omega_{ie}||, \quad \omega^{n}_{en} = \begin{bmatrix} R_N \\ R_\theta \\ R_\phi \end{bmatrix} = \begin{bmatrix} V_\nu/(R_1 + h) \\ -V_\theta/(R_2 + h) \\ -V_\phi \tan L/(R_3 + h) \end{bmatrix} = \begin{bmatrix} l \cos L \\ -L \\ -l \sin L \end{bmatrix} \] (21)

\[ R_L = \frac{R_0(1 - e^2)}{(1 - e^2 \sin^2 L)^{\frac{3}{2}}}, \quad R_1 = \frac{R_0}{(1 - e^2 \sin^2 L)^{\frac{3}{2}}} \] (22)
\[
\dot{L} = \frac{V_N}{R_L + h}
\]
\[
\dot{i} = \frac{V_E}{(R_l + h) \cos L}
\]
\[
\dot{h} = -V_D
\]
Quatetion-based Attitude Algorithm
What Is Non-Commutativity Error?

\[ \dot{C}_b^i = C_b^i \langle \omega_{ib}^b \rangle \quad \text{for implementation} \]

\[ C_b^i (t_{k+1}) = C_b^i (t_k) \left( I_{3 \times 3} + \langle \tilde{\omega}_{ib}^b (t_k) \rangle \right) \]

\[ \tilde{\omega}_{ib}^b (t_k) = \begin{bmatrix} \omega_x (t_k) & \omega_y (t_k) & \omega_z (t_k) \end{bmatrix}^T = [a \ a \ a]^T \]

<table>
<thead>
<tr>
<th></th>
<th>x-direction</th>
<th>y-direction</th>
<th>z-direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>\omega_x(t)</td>
<td>ε</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>t_{k-1}</td>
<td>t_k</td>
<td></td>
</tr>
<tr>
<td>\omega_y(t)</td>
<td>ε</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>t_{k-1}</td>
<td>t_k</td>
<td></td>
</tr>
<tr>
<td>\omega_z(t)</td>
<td>ε</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>t_{k-1}</td>
<td>t_k</td>
<td></td>
</tr>
</tbody>
</table>

*0 < \varepsilon < (t_k - t_{k-1})

ambiguous!
Need for Fast Computation

- Attitude is among most important information provided by inertial navigation systems

- The example illustrates how rotation sequence ambiguity (Non-commutivity error) occurs due to non-zero sampling interval for digitization.

- To minimize this error source, we should sample gyro outputs as fast as possible.

- For this purpose, we need an attitude algorithm that is numerically efficient and stable

  -> "quaternions"
**Attitude Algorithm Structure**

- At each submajor interval, **gyro outputs** are sampled
- At each minor interval, **quaternions** are updated
- At each major interval, **transformation matrices** are updated
Coning algorithm?

- Among the various attitude dynamics, conning motions stimulate largest non-commutivity errors.
- The analysis of attitude error under the conning motion is very important in SDINS since it is the major environmental error source.
2-Interval Coning Algorithm

- incremental angle from gyro output

\[
\theta_1 := \int_{T}^{T+h/2} \omega(\tau) d\tau \quad \theta_2 := \int_{T+h/2}^{T+h} \omega(\tau) d\tau
\]

- (incremental) rotation vector

\[
\Phi = \theta_1 + \theta_2 + \frac{2}{3} \langle \theta_1 \rangle \theta_2 = \begin{bmatrix}
\phi_x \\
\phi_y \\
\phi_z
\end{bmatrix}
\]

- (incremental) quaternion

\[
Q = \begin{bmatrix}
q_0 \\
q_1 \\
q_2 \\
q_3
\end{bmatrix} = \begin{bmatrix}
\cos(\phi/2) \\
\sin(\phi/2) \cos(\phi_x) \\
\sin(\phi/2) \cos(\phi_y) \\
\sin(\phi/2) \cos(\phi_z)
\end{bmatrix} = \begin{bmatrix}
q_0 \\
q_1 \\
q_2 \\
Q_{sub}
\end{bmatrix}
\]

\[\Phi = \phi \mathbf{n}, \quad \phi = \sqrt{\phi_x^2 + \phi_y^2 + \phi_z^2}\]
- **Quaternion update**

  **by incremental quaternion**

  \[
  Q_{k+1} = Q \otimes Q_k \quad \text{(by quaternion multiplication)}
  \]

  \[
  = \Pi(Q) \cdot Q_k \quad \text{(by matrix vector multiplication)}
  \]

  **where**

  \[
  \Pi(Q) := \begin{bmatrix}
  q_0 & -Q_{sub}^T \\
  Q_{sub} & q_0 I_{3 \times 3} + \langle Q_{sub} \rangle
  \end{bmatrix}
  \]

  \[
  Q = \begin{bmatrix}
  q_0 \\
  Q_{sub}
  \end{bmatrix}
  \]
Transformation Matrix by Quaternion

\[ C_b^n = \begin{bmatrix}
q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\
2(q_0 q_3 + q_1 q_2) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\
2(q_1 q_3 - q_0 q_2) & 2(q_0 q_1 + q_2 q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2
\end{bmatrix} \]

\[ C_b^n = \begin{bmatrix}
\cos\theta \cos\psi & \sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi & \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\
\cos\theta \sin\psi & \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi & \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \\
-\sin\theta & \sin\phi \cos\theta & \cos\phi \cos\theta
\end{bmatrix} \]

\[ \phi = \arctan\left( \frac{c_{23}}{c_{33}} \right) \]

\[ \psi = \arctan\left( \frac{c_{12}}{c_{11}} \right) \]

\[ \theta = \arctan\left( \frac{-c_{13}}{\sqrt{1 - c_{13}^2}} \right) \]

\[ q_0 = C_{\phi/2} C_{\theta/2} C_{\psi/2} + S_{\phi/2} S_{\theta/2} S_{\psi/2} \]

\[ q_1 = S_{\phi/2} C_{\theta/2} C_{\psi/2} - C_{\phi/2} S_{\theta/2} S_{\psi/2} \]

\[ q_2 = C_{\phi/2} S_{\theta/2} C_{\psi/2} + S_{\phi/2} C_{\theta/2} S_{\psi/2} \]

\[ q_3 = C_{\phi/2} C_{\theta/2} S_{\psi/2} - S_{\phi/2} S_{\theta/2} C_{\psi/2} \]
• By applying similar procedure, one can also get 3-, 4- and 5-sample algorithms.

• 3-sample coning algorithm is the most popular method for practical implementation.

\[ \Phi = \theta_1 + \theta_2 + \theta_3 + 0.4125\langle \theta_1 \rangle \theta_3 + 0.7125\langle \theta_2 \rangle (\theta_3 - \theta_1) \]

• There is also a branched method utilizing not only the angles in the same minor interval but also the fraction of angles sampled in the previous minor interval, i.e.
Summary of SDINS Algorithm
\[ \dot{V}^n = f^n - [2(\omega_{ie}^n \times) + (\omega_{en}^n \times)]V^n + g^n \]

\[ \dot{L} = \frac{V_N}{R_L + h} \quad \dot{i} = \frac{V_E}{(R_L + h)\cos L} \quad \dot{h} = -V_D \]

\[ \omega_{ie}^n + \omega_{en}^n \]

\[ G^{\text{GRAVITY COMPUTER}} \]

\[ C_b^n \]

\[ V^n \]

\[ (L, l, h) \]

\[ \omega_{ib}^n \]

\[ C_b^n \]

\[ \dot{C}_b^n = C_b^n (\omega_{ib}^n \times) \]

(coning algorithm)

\[ f^b \]

\[ C_b^m f^b \]

\[ 3 \text{ ACCELEROMETERS} \]

\[ 3 \text{ GYROSCOPES} \]

\[ \text{INITIAL ESTIMATES OF VELOCITY & POSITION} \]

\[ \text{INITIAL ESTIMATE OF ATTITUDE} \]
Error Modeling
- Let

\[ y = F(x, a, b, u) \]

then

\[ \delta y = \delta F = \frac{\partial F}{\partial x} \delta x + \frac{\partial F}{\partial a} \delta a + \frac{\partial F}{\partial b} \delta b + \frac{\partial F}{\partial u} \delta u \]
Perturbation Method 2

- Consider the following dynamic equation
  \[ \dot{x} = ax + bu \]  
  where \( a, x, b, \) and \( u \) denote true values of interest.

- In actual situation, we never know the true values shown in Eq. (p-1).
  However, we can utilize the estimates \( \hat{a}, \hat{x}, \hat{b}, \) and \( \hat{u} \) instead of the true values \( a, x, b, \) and \( u \) as follows.
  \[ \dot{x} = \hat{a}\hat{x} + \hat{b}\hat{u} \]  
  where
  \[ \hat{x} := x + \delta x, \quad \hat{a} := a + \delta a, \quad \hat{b} := b + \delta b, \quad \hat{u} := u + \delta u, \]
  and \( \delta a, \delta x, \delta b, \) and \( \delta u \) denote error terms.

- Substituting (p-3) to (p-1) and subtracting (p-2), we obtain
  \[ \ddot{x} = \dot{\hat{a}}\delta x + \dot{\hat{b}}\delta u + \delta \hat{a}\hat{x} + \delta \hat{b}\hat{u} \]
  where products of errors are neglected.
Example: function

\[ R_L = \frac{R_0(1 - e^2)}{(1 - e^2 \sin^2 L)^{3/2}}, \quad \delta R_L = R_{LL} \delta L \]

\[ R_{LL} := \frac{\partial R_L}{\partial L} = -\frac{3}{2} \frac{R_0(1 - e^2)}{(1 - e^2 \sin^2 L)^{5/2}} \frac{\partial}{\partial L} \left(- e^2 \sin^2 L \right) \]

\[ = -\frac{3}{2} \frac{R_0(1 - e^2)}{(1 - e^2 \sin^2 L)^{5/2}} \left(- 2e^2 \sin L \cos L \right) \]

\[ = \frac{3R_0(1 - e^2)e^2 \sin L \cos L}{(1 - e^2 \sin^2 L)^{5/2}} \]
Example: (error) time propagation

\[ \dot{L} = \frac{V_N}{R_L + h} \]

\[ \delta \dot{L} = \frac{\partial \dot{L}}{\partial R_L} \delta R_L + \frac{\partial \dot{L}}{\partial h} \delta h + \frac{\partial \dot{L}}{\partial V_N} \delta V_N \]

\[ \frac{\partial \dot{L}}{\partial R_L} = -\frac{V_N}{(R_L + h)^2} = \frac{1}{(R_L + h)} \frac{(-V_N)}{(R_L + h)} = \frac{\rho_E}{R_L + h} \]

\[ \frac{\partial \dot{L}}{\partial V_N} = \frac{1}{R_L + h} \]

\[ \frac{\partial \dot{L}}{\partial h} = -\frac{V_N}{(R_L + h)^2} = \frac{1}{(R_L + h)} \frac{(-V_N)}{(R_L + h)} = \frac{\rho_E}{R_L + h} \]
**Example: (indirect) meas. eq.**

\[
\tilde{\rho}^j = (e_u^j)^T (R^j - X^e) + c b_{\text{clock}} + \nu^j
\]

\[
\hat{\rho}^j = (e_u^j)^T (R^j - \hat{X}^e) - c \hat{b}_{\text{clock}}
\]

\[
\hat{X}^e = X^e + \delta X^e
\]

\[
\hat{b}_{\text{clock}} = b_{\text{clock}} + \delta b_{\text{clock}}
\]

\[
z^j_\rho = \tilde{\rho}_u^j - \hat{\rho}_u^j = (e_u^j)^T \delta X^e - c \delta b_{\text{clock}} + \nu^j
\]
\[
\dot{X}_{\text{INS}} = F_{\text{INS}} X_{\text{INS}} + W_{\text{INS}}
\]

where

\[
X_{\text{INS}} = \begin{bmatrix} X_f^T & X_a^T \end{bmatrix}^T
\]

\[
X_f = \begin{bmatrix} \delta L & \delta \theta & \delta h & \delta V_N & \delta V_E & \delta V_D & \hat{\Phi}_N & \hat{\Phi}_E & \hat{\Phi}_D \end{bmatrix}^T
\]

\[
X_a = \begin{bmatrix} \nabla_x & \nabla_y & \nabla_z & \epsilon_x & \epsilon_y & \epsilon_z \end{bmatrix}^T
\]

\[
W_{\text{INS}} = \begin{bmatrix} O_{1 \times 3} & w_{aX} & w_{aY} & w_{aZ} & w_{gX} & w_{gY} & w_{gZ} & O_{1 \times 6} \end{bmatrix}^T \sim \mathcal{N}(O_{15 \times 1}, Q_{\text{INS}})
\]

\[
F_{\text{INS}} = \begin{bmatrix}
F_{11} & F_{12} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} \\
F_{21} & F_{22} & F_{23} & F_{24} & O_{3 \times 3} \\
F_{31} & F_{32} & F_{33} & O_{3 \times 3} & F_{35} \\
O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} \\
O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3}
\end{bmatrix}
\]
\[ F_{11} = \begin{bmatrix} \frac{R_{ll} \rho_E}{R_L + h} & 0 & \frac{\rho_E}{R_L + h} \\ \rho_N \left( \tan L - \frac{R_L}{R_L + h} \right) & 0 & -\rho_N \sec L \\ 0 & 0 & 0 \end{bmatrix} \quad F_{12} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \frac{1}{R_L + h} \begin{bmatrix} 0 \\ \sec L \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \]

\[ F_{21} = \begin{bmatrix} \frac{\rho_E R_{ll}}{R_L + h} V_D - \rho_N \sec^2 L + 2\Omega_N V_E - \rho_N \rho_E R_{il} \\ 2\Omega_N + \rho_N \sec^2 L + \frac{\rho_E R_{il}}{R_L + h} V_N - \left( \frac{\rho_N R_{il}}{R_L + h} - 2\Omega_D \right) V_D \\ \rho_N^2 R_{il} + \rho_E^2 R_{ll} - 2\Omega_D V_E \end{bmatrix} \begin{bmatrix} \frac{\rho_E}{R_L + h} V_D - \rho_N \rho_D \\ \frac{\rho_D}{R_L + h} V_N - \frac{\rho_N}{R_L + h} V_D \\ \rho_N^2 + \rho_E^2 \end{bmatrix} \]
$$F_{22} = \begin{bmatrix}
\frac{V_D}{R_L + h} & 2\rho_D + 2\Omega_D & -\rho_B \\
-\rho_D - 2\Omega_D & \frac{V_N \tan L + V_D}{R_L + h} & 2\Omega_N + \rho_N \\
2\rho_B & -2\Omega_N - 2\rho_N & 0 
\end{bmatrix}$$

$$F_{31} = \begin{bmatrix}
\frac{\rho_N R_{lL}}{R_L + h} & \rho_N & 0 \\
-\rho_N R_{tL} & -\rho_N & 0 \\
-\rho_N \sec^2 L - \frac{\rho_D R_{lL}}{R_L + h} & -\rho_D & 0 
\end{bmatrix}$$

$$F_{32} = \begin{bmatrix}
0 & \frac{1}{R_L + h} & 0 \\
-1 & 0 & 0 \\
0 & -\frac{\tan L}{R_L + h} & 0 
\end{bmatrix}$$

$$F_{33} = \begin{bmatrix}
0 & \frac{\Omega_D + \rho_D}{R_L + h} & -\rho_B \\
-\frac{\Omega_D}{R_L + h} & 0 & \Omega_N + \rho_N \\
\rho_B & -\Omega_N - \rho_N & 0 
\end{bmatrix}$$