Aalto University
School of Electrical Engineering

Pose Estimation, Inertial Navigation Systems (INS), Inertial Measurement Unit (IMU)

Arto Visala

26.3.2019

Outline

- Introduction
- Mathematics of Inertial Navigation
- Errors and Aiding in Inertial Navigation
- Example: Simple Odometry Aided AHRS
- Use of cheap MEMS IMUs in robotics
- Summary

History, Inertial Navigation Systems (INS)

- Historical roots in German Peenemunde Group.
- Modern form credited to Charles Draper et al. @MIT.
- 1940s Germany:
  - V2 program, gyroscopic guidance
- 1950s Draper Labs, MIT:
  - Shuler tuned INS
  - Floated rate integrating gyros (0.01
  - deg/hr)
- 1960s DTGs
  - not floated or temp compensated
- 1970s RLGs, USA
- 1980s Strapdown INS

V2

Introduction INS

Advantages

– Most accurate dead reckoning available.

– Useful in wide excursion (outdoor) missions.

– Work anywhere where gravity is known.

– Are jamproof - require no external information.

– Radiate nothing - exhibit perfect stealth.

Disadvantages

– Cannot sense accelerations of unpowered space flight.

– Most errors are time dependent.

– Requires input of initial conditions.

Mathematics of Inertial Navigation

Use Inertial Properties of Matter

– Accelerometers

– Gyros

Do "Dead Reckoning"

– Integrate acceleration twice

IMU

$$\vec{v}(t) = \vec{v}(0) + \int_0^t \vec{a}\,dt$$

$$\vec{r}(t) = \vec{r}(0) + \int_0^t \vec{v}\,dt$$

Computation

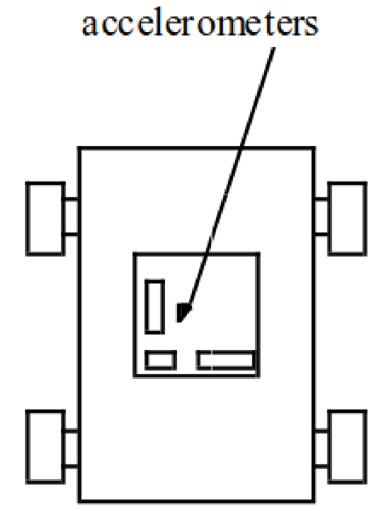
Mathematics of Inertial Navigation (Naïve Concept)

Just integrating 3 accels will not work for a lot of reasons:

– Accelerometers measure wrong quantity.

– They measure it in wrong reference frame.

– They represent it in wrong coordinate system.



$$\vec{f} \neq \vec{a}$$

$$\vec{a}_v^{\,i} \neq \vec{a}_v^{\,e}$$

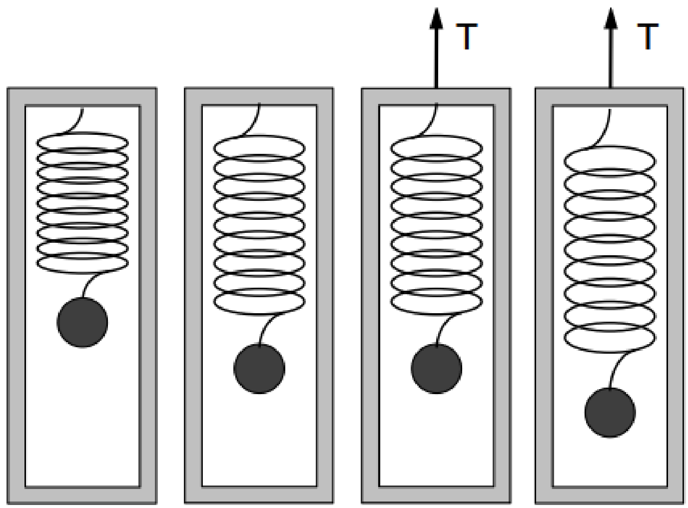
$${}^{v}\bar{a}_e^{v} \neq {}^{e}\bar{a}_e^{v}$$

The quest for ever better engineering solutions to these problems is the primary reason for the complexity of the

modern INS.

Problem 1: Equivalence

- Accelerometers don't measure acceleration.
- Specific force is: $\vec{f} = \vec{a} - \vec{g}$
- Fix: must know gravity, then: $\vec{a} = \vec{f} + \vec{g}$



Freefall (Space) | At Rest (Earth) | Accelerating $a = 9.8\ m/s^2$ | Both

Problem 2: Inertial Frame of Reference

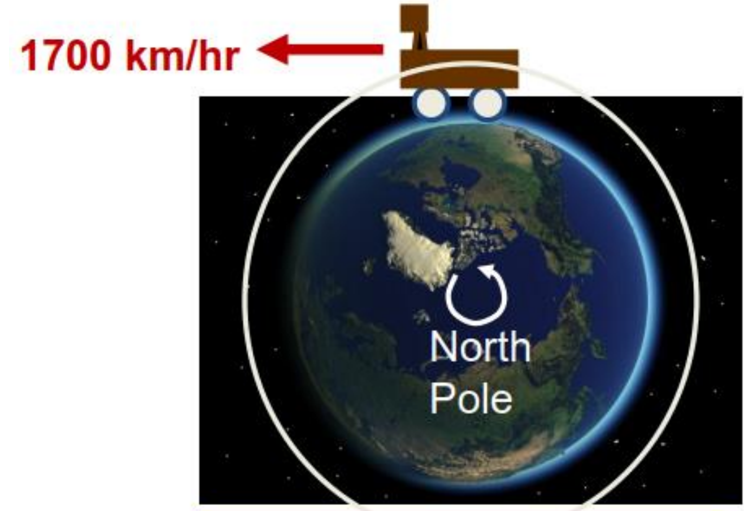
Now have inertial acceleration.

– Want earth-referenced acceleration.

Fix: account for earth angular velocity:

– “Apparent forces”.

– “gravity”, not gravitation



Problem 3: Body Coordinates

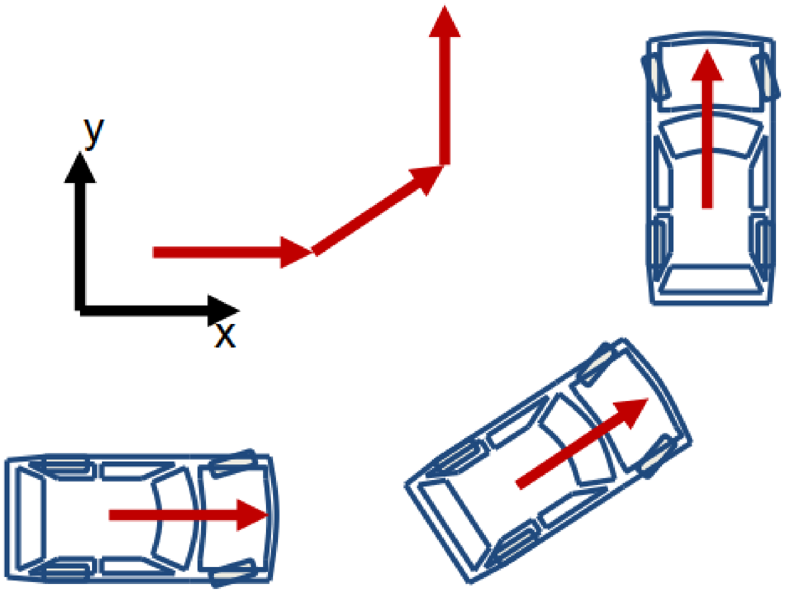
Accelerometers are fixed to vehicle.

– Want to integrate in the world frame.

Need to know instantaneous heading.

So…, track orientation

– Use gyros.



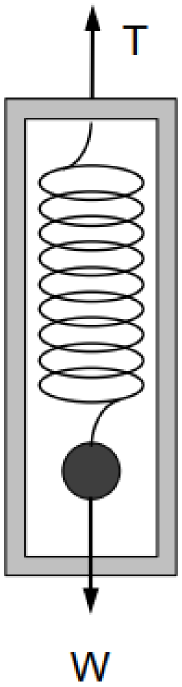
$$\theta(t) = \int_0^t \omega(t)\,dt$$

First Fix: Specific Force to Acceleration

We know specific force is not acceleration.

The fundamental equation of inertial navigation is Newton's 2nd law applied to the accelerometers:

$$\sum F = \vec{T} + \vec{W} = m\vec{a}^{i}$$



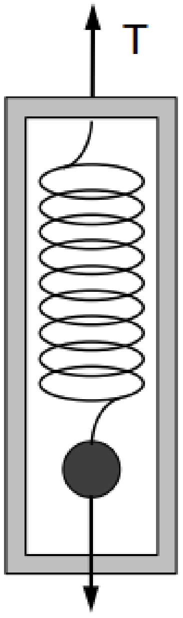
Need to solve for acceleration….

First Fix: Specific Force to Acceleration

Solving for acceleration:

$$\vec{a}^i = \frac{\vec{T}}{m} + \frac{\vec{W}}{m} = \vec{t} + \vec{w}$$

Gravitational field

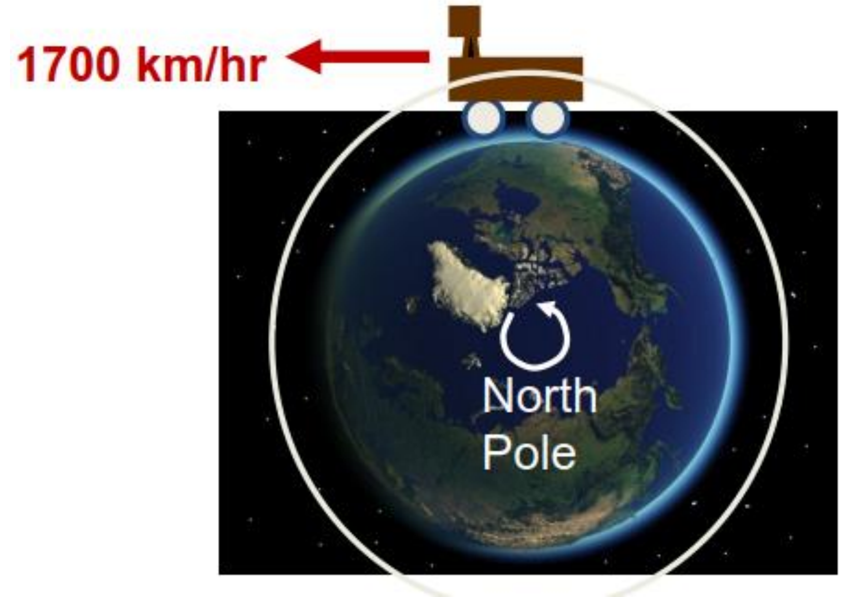


Note: you need to know the gravitational field anywhere you want to do inertial navigation.

Second Fix: Remove Apparent Forces

Moving vehicle is a moving reference frame.

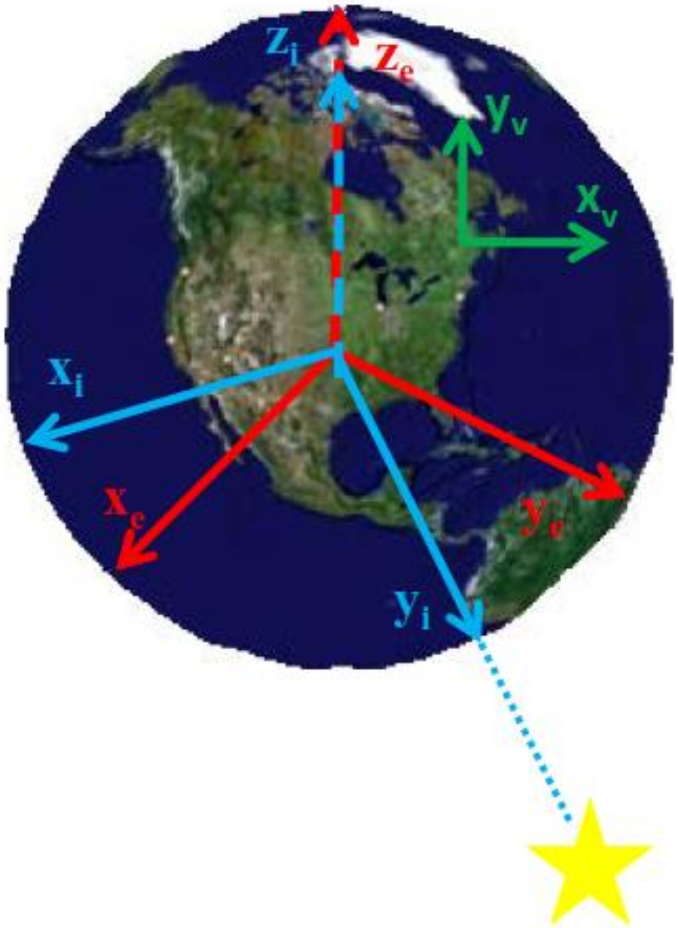
- Hence, sensors on-board will sense apparent forces.
- Remove them with Coriolis law.



Second Fix: Remove Apparent Forces

Define Frames:

- i: “inertial”, geocentric nonrotating.
- e: “earth”, geocentric, rotating.
- v: “vehicle”, fixed to accels. Also known as body frame.



Second Fix: Remove Apparent Forces

Define:

$\vec{r}_v^{\,x}$	Position of vehicle measured in frame x
$\vec{v}_v^{\,x}$	Velocity of vehicle measured in frame x
$\vec{a}_v^{\,x}$	Acceleration of vehicle measured in frame x

Second Fix: Remove Apparent Forces

Basic acceleration transformation under negligible angular acceleration:

$$\vec{a}_o^{f} = \vec{a}_o^{m} + \vec{a}_m^{f} + 2\vec{\omega}_m^{f} \times \vec{v}_o^{m} + \vec{\omega}_m^{f} \times [\vec{\omega}_m^{f} \times \vec{r}_o^{m}]$$

Let *"o"* = *v*, *"m"* = *e*, and *"f"* = *i*:

$$\vec{a}_v^{i} = \vec{a}_v^{e} + \vec{a}_e^{i} + 2\vec{\omega}_e^{i} \times \vec{v}_v^{e} + \vec{\omega}_e^{i} \times [\vec{\omega}_e^{i} \times \vec{r}_v^{e}]$$

The *i* and *e* origins are coincident. Hence:

$$\vec{a}_e^{i} = \vec{0}$$

Second Fix: Remove Apparent Forces

Also, let the earth sidereal rate be given by:

$$\vec{\omega}_e^i = \vec{\Omega}$$

Now, moving the earth acceleration to the left hand side, we have:

$$\vec{a}_v^e = \vec{a}_v^i - 2\vec{\Omega} \times \vec{v}_v^e - \vec{\Omega} \times (\vec{\Omega} \times \vec{r}_v^e)$$

Substituting for specific force:

$$\vec{a}_v^e = \vec{t} - 2\vec{\Omega} \times \vec{v}_v^e + \vec{w} - \vec{\Omega} \times (\vec{\Omega} \times \vec{r}_v^e)$$

"Gravity"

Centripetal

Second Fix: Remove Apparent Forces

The quantity: $\vec{w} - \vec{\Omega} \times (\vec{\Omega} \times \vec{r}_v^{\,e})$

Is known as “gravity” and denoted $\vec{g}$

Finally, we have “the” equation of inertial navigation.

$$\vec{a}_v^{\,e} = \left(\frac{d\vec{v}_v^{\,e}}{dt}\right)_e = \vec{f} - 2\vec{\Omega} \times \vec{v}_v^{\,e} + \vec{g}$$

Specific Force

Coriolis

Gravity

This is the derivative of the velocity relative to e as computed by an earth-fixed observer.

Second Fix: Remove Apparent Forces

The computed solution in coordinate system independent form is:

$$\vec{v}_v^e = \int^{t} [\vec{t} - 2\vec{\Omega} \times \vec{v}_v^e + \vec{g}]\,dt\big|_e + \vec{v}_v^e(t_0)$$

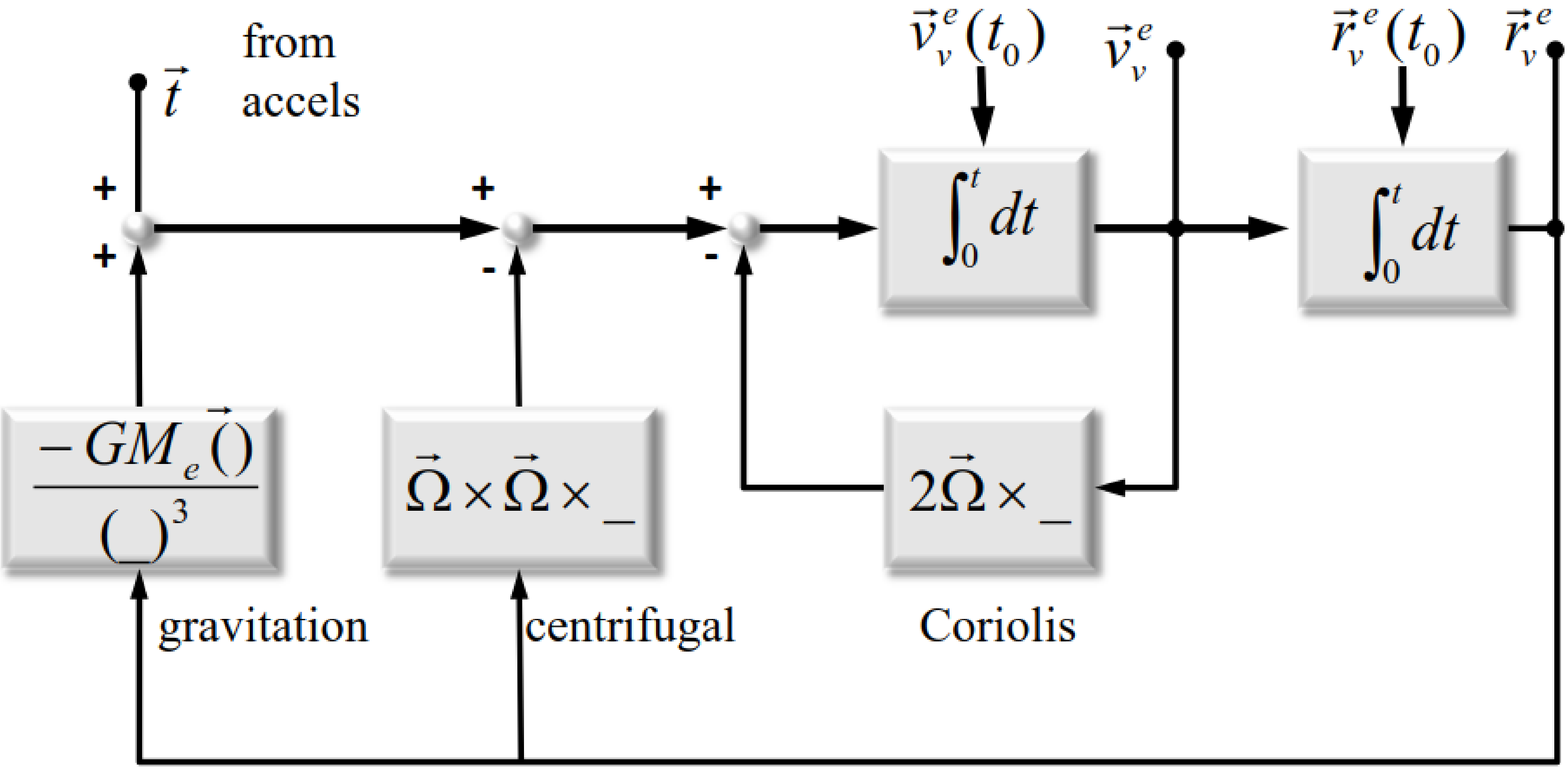
$$\vec{r}_v^e = \int_0^{t} \vec{v}_v^e\,dt\big|_e + \vec{r}_v^e(t_0)$$

You need to know:

- a model of gravity
- earth sidereal rate
- specific forces
- initial position
- initial velocity
- (gyros don't appear in vector form)

These are only valid if you integrate in the earth frame (i.e. in earth-fixed coordinates).

Vector Formulation

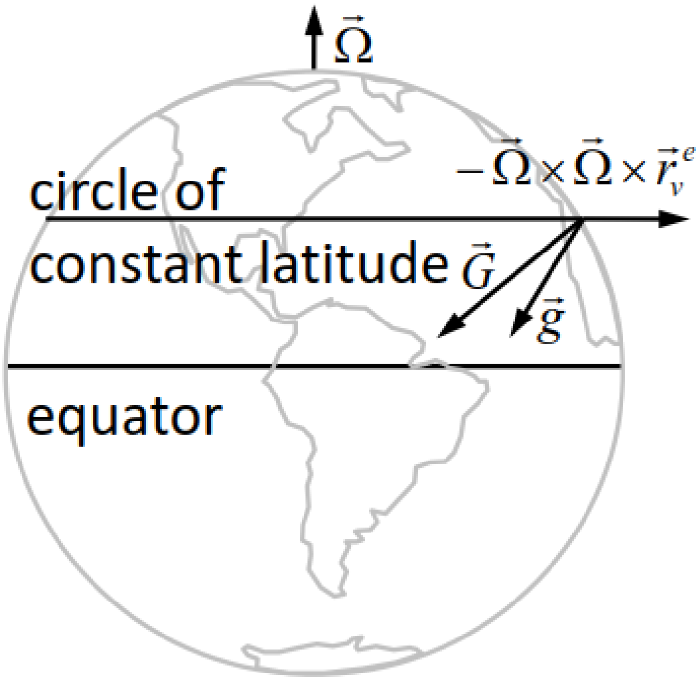


Gravity and Gravitation

Gravity is the force per unit mass required to fix an object wrt the Earth. It includes centrifugal force.

Gravitation is the force described in Newton's law of gravitation.

$$\vec{W} = \frac{\vec{w}}{m} = -\left[\frac{GM_e \vec{r}_v^{\,e}}{\left|\vec{r}_v^{\,e}\right|^3}\right]$$



Only at the equator and at the poles does gravity point toward the center of the earth.

Third Fix: Adopt a Coordinate System

- The heart of the INS is the inertial measurement unit (IMU) containing 3 accelerometers and 3 gyros.
- The gyros are used to track the orientation of the vehicle wrt the earth.
- You need orientation because:
  - $\hat{g}$ and $\vec{\Omega}$ are known in earth coordinates, whereas....
  - $\hat{t}$ and $\vec{\omega}$ are measured in body coordinates in a modern strapdown system.
- Can’t add em up unless they are in the same coordinate system.

Third Fix: Euler Angles

- Step 1: Integrate the gyros:

$$\begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix} = \int_0^t \cdot \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} dt + \begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix}_0 = \int_0^t \cdot \begin{bmatrix} c\phi & 0 & s\phi \\ t\theta s\phi & 1 & -t\theta c\phi \\ -\frac{s\phi}{c\theta} & 0 & \frac{c\phi}{c\theta} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} dt + \begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix}_0$$

Third Fix: Direction Cosines

- Step 1: Or, use direction cosine form (better):

$$\delta\underline{\Theta} = \underline{\omega}\delta t \qquad \delta\Theta = \left|\delta\underline{\Theta}\right|$$

$$f_1(\delta\Theta) = \frac{\sin\delta\Theta}{\delta\Theta} \qquad f_2(\delta\Theta) = \frac{(1-\cos\delta\Theta)}{\delta\Theta^2}$$

$$R_{k+1}^{k} = I + f_1(\delta\Theta)[\delta\underline{\Theta}]^{X} + f_1(\delta\Theta)([\delta\underline{\Theta}]^{X})^2$$

$$R_{k+1}^{n} = R_k^n R_{k+1}^{k}$$

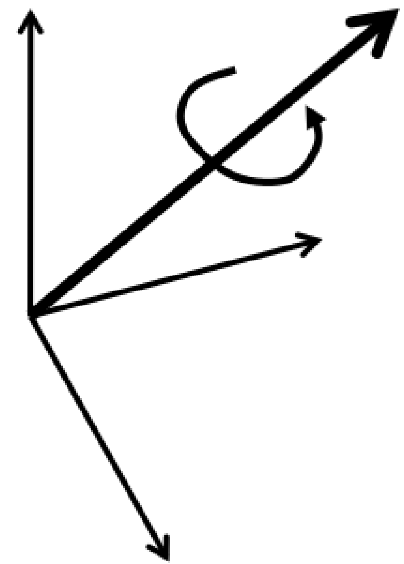
Direction Cosines from Angular Velocity

- Hence, the direct transformation from angular velocity to direction cosines is the recursion:

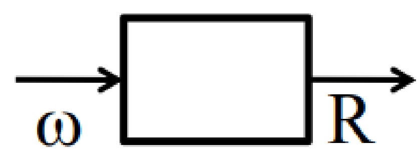
$$R^n_{k+1} = R^n_k R^k_{k+1}$$

$$R^k_{k+1} = I + f_1(\delta\Theta)[\delta\underline{\Theta}]^X + f_2(\delta\Theta)([\delta\underline{\Theta}]^X)^2$$

$$f_1(\delta\Theta) = \frac{\sin\delta\Theta}{\delta\Theta} \qquad f_2(\delta\Theta) = \frac{(1-\cos\delta\Theta)}{\delta\Theta^2}$$



- Where dt is the time step, $d\Theta = |d\underline{\Theta}|$ and $d\underline{\Theta} = \underline{\omega}\, dt$.
- Advantage: You don't need to solve for the Euler angles.



Third Fix: Quaternions

- Step 1: Or, use the quaternion form (best):

$$\underline{\delta\Theta} = \underline{\omega}\delta t \qquad \delta\Theta = \left|\underline{\delta\Theta}\right|$$

$$\tilde{q}^{k}_{k+1} = \cos\delta\Theta[I] + \sin\delta\Theta\left[\left({}^{\times}[\tilde{\omega}_b]\right)/\left|\vec{\omega}_b\right|\right]$$

$$\tilde{q}^{n}_{k+1} = \tilde{q}^{k}_{k+1}\tilde{q}^{n}_{k}$$

Notations

- 4-tuples (q_0, q_1, q_2, q_3)
- Hypercomplex numbers $q = q_0 + q_1 i + q_2 j + q_3 k$
- Sum of real and imaginary parts $\tilde{q} = q + \vec{q}$
- Ordered doublet $(q, \vec{q})$
- Exponential $q = e^{\frac{1}{2}\theta \vec{w}}$

Manipulate like Polynomials

I will use these two

Rotations as Quaternions

- The unit quaternion: $$\tilde{q} = \cos\frac{\theta}{2} + \hat{w}\sin\frac{\theta}{2}$$

- Represents the operator which rotates by the angle θ around the axis whose unit vector is $\hat{w}$.

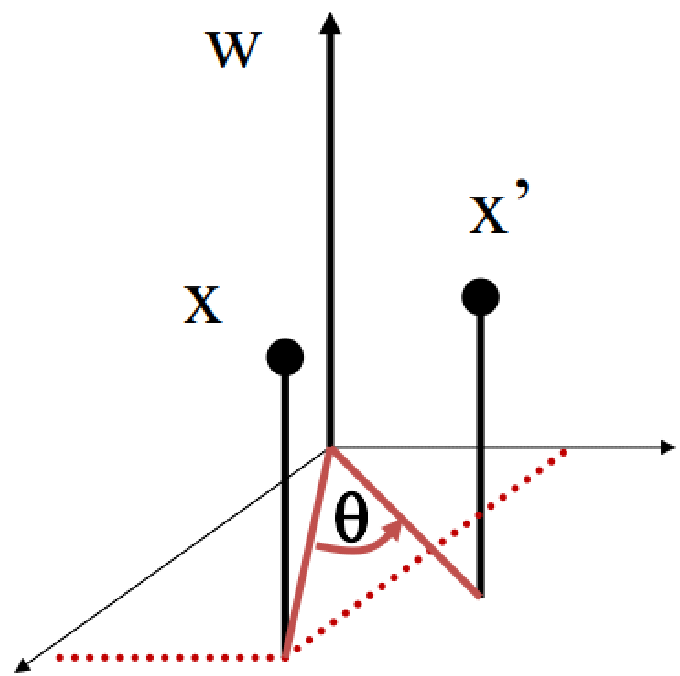
- The inverse is clearly: $$\hat{w} = \vec{q} / |\vec{q}|$$ $$\theta = 2\,\mathrm{atan2}(|\vec{q}|, q)$$

Real vectors are just quaternions 0+xi+yj+zk

Rotating a Vector (Point)

- Use the quaternion sandwich:

$$\tilde{x}' = \tilde{q}\tilde{x}\tilde{q}^*$$



Quaternion to Rot() Matrix

- For the quaternion:

$$q = q_0 + q_1 i + q_2 j + q_3 k$$

- The equivalent Rot() matrix is:

$$R = \begin{bmatrix} 2[q_0^2 + q_1^2] - 1 & 2[q_1 q_2 - q_0 q_3] & 2[q_1 q_3 + q_0 q_2] \\ 2[q_1 q_2 + q_0 q_3] & 2[q_0^2 + q_2^2] - 1 & 2[q_2 q_3 - q_0 q_1] \\ 2[q_1 q_3 - q_0 q_2] & 2[q_0 q_1 + q_2 q_3] & 2[q_0^2 + q_3^2] - 1 \end{bmatrix}$$

Rot() Matrix to Quaternion

- For the Rot() matrix:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

- The equivalent quaternion is determined from:

$$r_{11} + r_{22} + r_{33} = 4q_0^2 - 1$$
$$r_{11} - r_{22} - r_{33} = 4q_1^2 - 1$$
$$-r_{11} + r_{22} - r_{33} = 4q_2^2 - 1$$
$$-r_{11} - r_{22} + r_{33} = 4q_3^2 - 1$$

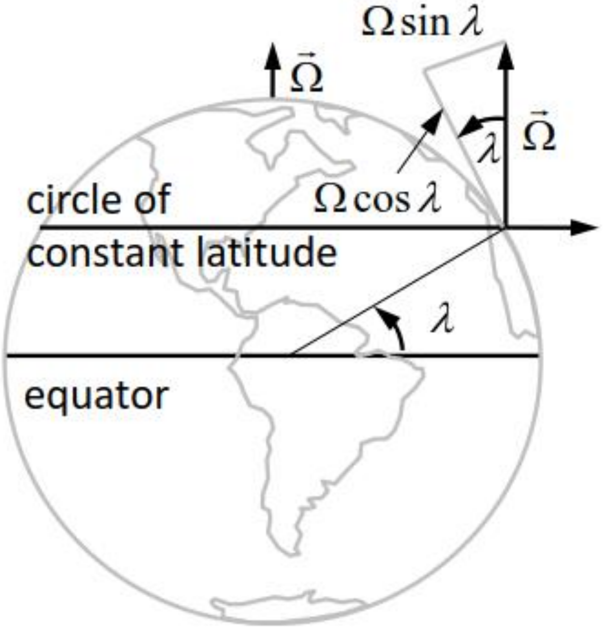
Are quaternions unique for a given rotation?

Third Fix: Earth Rate Compensation

- When orientation aiding is rare (yaw aiding is typically rare), it may be useful to remove earth rate from the gyros:

$$ {}^{v}\underline{\omega}_{v}^{e} = {}^{v}\underline{\omega}_{v}^{i} - R_{e}^{v} R_{i}^{e} \underline{\Omega}_{e}^{i} $$

- .. Or its projection onto the yaw axis will be integrated.



$$ {}^{i}\underline{\Omega}_{n}^{i} = \Omega \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{T} $$

$$ {}^{n}\underline{\Omega}_{n}^{i} = \Omega \begin{bmatrix} c\lambda & 0 & -s\lambda \end{bmatrix}^{T} $$

Note: n = e here

Third Fix: Adopt a Coordinate System

- Step 2: Integrate the accels:

$$\underline{v}_v^e = \int_0^t [R_v^e \underline{t} + \underline{g} - 2\underline{\Omega} \times \underline{v}_v^e] dt + \underline{v}_v^e(t_0)$$

- Step 3: Integrate the velocity:

$$\underline{r}_v^e = \int_0^t \underline{v}_v^e dt + \underline{r}_v^e(t_0)$$

Sensitivity in Inertial Navigation

TABLE 3. Term Magnitudes

Term Name	Expression	Nominal Value
Specific Force	$\vec{f}$	0.1 g
Gravitational	$\vec{g}$	1.0 g
Centrifugal	$\vec{\Omega} \times \vec{\Omega} \times \vec{r}_v^e$	3.5x10^{-3} g
Coriolis	$2\vec{\Omega} \times \vec{v}_v^e$	1.5x10^{-4} g

For a vehicle at the equator, moving eastward at a velocity of 10 meters per second, and accelerating at 0.1 g

- Acceleration is multiplied by the square of time.
  - 1 hour2 = 13 million secs^2.
- After 1 hour, the Coriolis (smallest) term accounts for over 9.5 Km of error.

Error Explosion in Inertial Navigation

- For a 10 m/s vehicle at the equator, the Coriolis term is tiny:
  - 1.5x10-4 g
- Consider an error of this magnitude…
- In one hour:
  - $t^2 = (3600)^2 = 13$ million !!
- Position Error:
  - 9.5 Kilometers!!!

$$\vec{a}_{Coriolis} = 2\vec{\Omega} \times \vec{v}_v^e$$

$$\vec{a}_{meas}(t) = \vec{a}_{true}(t) + \vec{a}_{err}(t)$$

$$\vec{r}_{err}(t) = \int_0^t \int_0^t \vec{a}_{err}(t)\,dt = \frac{\vec{a}_{err} t^2}{2}$$

Simple Odometry Aided AHRS

- Attitude and heading reference systems (*AHRS*)
  - With GPS For example https://www.xsens.com/tags/ahrs/
- The AHRS is a degenerate form of inertial navigation system, using much of the same components:
  - indicates orientation only.

Means not stabilized

- Device uses a strapped down IMU today.
  - Accels indicate gravity and acceleration
  - Gyros indicate angular velocity
- Distinguishing acceleration from gravity is still an issue - but less so.

Nav Eqns in Body Frame

- Recall the inertial nav equation

$$\vec{a}_v^e = \left(\frac{d\vec{v}_v^e}{dt}\right)_e = \vec{t} - 2\vec{\Omega} \times \vec{v}_v^e + \vec{g}$$

- Lets express this in the body frame so that it becomes unnecessary to known orientation.
- Use the Coriolis theorem:

$$\left(\frac{d\vec{v}_v^e}{dt}\right)_v = \left(\frac{d\vec{v}_v^e}{dt}\right)_e + \vec{\omega}_v^e \times \vec{v}_v^e$$

This adds another Apparent Coriolis Force.

Nav Eqns in Body Frame

- Define the strapdown angular velocity:

$$\vec{\omega} = \vec{\omega}_v^{\,i} = \vec{\omega}_v^{\,e} + \vec{\omega}_e^{\,i} = \vec{\omega}_v^{\,e} + \vec{\Omega}$$

- Write the inertial navigation equation in the body frame:

$$\left(\frac{d\vec{v}_v^{\,e}}{dt}\right)_v = \vec{t} - (\vec{\omega} + \vec{\Omega}) \times \vec{v}_v^{\,e} + \vec{g}$$

$$\vec{a}_v^{\,e} = \left(\frac{d\vec{v}_v^{\,e}}{dt}\right)_e = \vec{t} - 2\vec{\Omega} \times \vec{v}_v^{\,e} + \vec{g}$$

Recall Earth Frame

- For this purpose, earth rate can be neglected, so:

$$\left(\frac{d\vec{v}_v^{\,e}}{dt}\right)_v = \vec{t} - \vec{\omega} \times \vec{v}_v^{\,e} + \vec{g}$$

Nav Eqns in Body Frame

- Solve for gravity:

$$\vec{g} = \left(\frac{d\vec{v}_v^e}{dt}\right)_v - \vec{t} + \vec{\omega} \times \vec{v}_v^e$$

This vanishes on an Ackerman vehicle during periods of constant speed. Otherwise, differentiate numerically.

Simply remove Coriolis term from the accel readings.

- Everything on right is known from measurements. g on left is known in world coordinates.

Nav Eqns in Body Frame

- Write this in body coordinates:

$$R_w^v \underline{g} = \frac{d({}^v\underline{v}_v^e)}{dt} - \underline{t} + {}^v\underline{\omega} \times {}^v\underline{v}_v^e = {}^v\underline{g}_{meas}$$

- Can solve this for attitude (not yaw) in the rotation matrix using inverse kinematics.
  - Rotation around g is not observable.

Solving for Attitude

- To get the attitude, express in body frame:

$$R_w^v \underline{g} = \frac{\mathrm{d}(\underline{{}^v v_v^e})}{\mathrm{d}t} - \underline{t} + \underline{{}^v\omega} \times \underline{{}^v v_v^e} = {}^v\underline{g}_{meas}$$

- Where:

$$Roty(\theta)Rotx(\phi)\underline{g} = \begin{bmatrix} c\theta & s\theta s\phi & s\theta c\phi \\ 0 & c\phi & -s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}$$

- The transpose converts from world to body, thus:

$${}^v\underline{g}_{meas} = R_w^v {}^w\underline{g} = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -s\theta \\ s\theta s\phi & c\phi & c\theta s\phi \\ s\theta c\phi & -s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = g \begin{bmatrix} -s\theta \\ c\theta s\phi \\ c\theta c\phi \end{bmatrix}$$

Solving for Attitude

- The solution is:

$$\tan\theta = s\theta / c\theta = -g_x / (\sqrt{g_y^2 + g_z^2})$$

$$\tan\phi = s\phi / c\phi = g_y / g_z$$

- To get the yawrate, solve:

$$\dot{\psi} = \frac{s\phi}{c\theta}\omega_y + \frac{c\phi}{c\theta}\omega_z$$

Summary of INS (and IMU)

- Black magic ?
- Hard to do well.
– Costs big bucks.
- Most accurate dead reckoning available.
– Cruise: 0.2 nautical miles of error per hour of operation.
- Indispensable on outdoor mobile robots.
- Complementary technology to GPS.

Summary of INS (and IMU)

- Inertial navigation is based on Newton's laws
– Works everywhere that gravity is known.
– It is stealthy and jamproof.
- Modern "strapdown" systems
– "computationally stabilized".
– no stabilized platform
- Naive approaches are seriously flawed. Must compensate for
– Gravity
– inertial forces
– body fixed coordinates.

Summary of INS (and IMU)

- Free inertial performs miserably…

– 1 part in 10,000 acceleration error causes kilometers of position error after 1 hour of operation.

- Interesting Error Dynamics

– Horizontal errors bounded, oscillate every 84 minutes

– Vertical position is unstable without damping devices

- An AHRS unit can find attitude from accelerometers and gyros and odometry.

Example 2: A DCM Based Attitude Estimation Algorithm for Low-Cost MEMS IMUs

- MEMS gyroscopes and accelerometers are relatively cheap. However, they give more inaccurate measurements than conventional high-quality sensors.
- An extended Kalman filter is implemented to estimate attitude in direction cosine matrix (DCM) formation and to calibrate gyroscope biases online.
- Particularly Biases of the MEMS gyros have to be estimated
- MEMS sensors are sensitive to temperature variations, which must be modelled and compensated
- See the details in the journal paper by Hyyti& Visala

DCM Based Partial Attitude Estimation.

- Direction Cosine Matrix (DCM) Euler angles in the ZYX convention have the following relation to the DCM:

$$
{}^{n}_{b}\mathbf{C} = \begin{bmatrix} \theta_c\psi_c & -\phi_c\psi_s + \phi_s\theta_s\psi_c & \phi_s\psi_s + \phi_c\theta_s\psi_c \\ \theta_c\psi_s & \phi_c\psi_c + \phi_s\theta_s\psi_s & -\phi_s\psi_c + \phi_c\theta_s\psi_s \\ -\theta_s & \phi_s\theta_c & \phi_c\theta_c \end{bmatrix}
$$

- Frames: b body, n navigation

$$
{}^{n}_{b}\dot{\mathbf{C}} = {}^{n}_{b}\mathbf{C}\left[{}^{b}\boldsymbol{\omega}\times\right]
$$

$$
\left[{}^{b}\boldsymbol{\omega}\times\right] = \begin{bmatrix} 0 & -{}^{b}\omega_z & {}^{b}\omega_y \\ {}^{b}\omega_z & 0 & -{}^{b}\omega_x \\ -{}^{b}\omega_y & {}^{b}\omega_x & 0 \end{bmatrix}
$$

DCM Based Partial Attitude Estimation.

- Bottom-row elements of DCM are collected into a row vector which can be updated using

$$
{}^{n}_{b}\dot{\mathbf{C}}_3 = \begin{bmatrix} {}^{n}_{b}\dot{C}_{31} \\ {}^{n}_{b}\dot{C}_{32} \\ {}^{n}_{b}\dot{C}_{33} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -{}^{n}_{b}C_{33} & {}^{n}_{b}C_{32} \\ {}^{n}_{b}C_{33} & 0 & -{}^{n}_{b}C_{31} \\ -{}^{n}_{b}C_{32} & {}^{n}_{b}C_{31} & 0 \end{bmatrix}}_{[\mathbf{C}_3\times]} \underbrace{\begin{bmatrix} {}^{b}\omega_x \\ {}^{b}\omega_y \\ {}^{b}\omega_z \end{bmatrix}}_{\mathbf{u}}
$$

- Rotates the current DCM vector according to the measured angular velocities, used as inputs

DCM Based Partial Attitude Estimation.

- The observation model is constructed using accelerometer measurements f, which are compared to the current estimate of the direction of gravity

$$ {}^{b}\mathbf{f} = \begin{bmatrix} {}^{b}f_x \\ {}^{b}f_y \\ {}^{b}f_z \end{bmatrix} = {}^{n}_{b}\mathbf{C}^{T} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} {}^{n}_{b}C_{31} \\ {}^{n}_{b}C_{32} \\ {}^{n}_{b}C_{33} \end{bmatrix} g $$

- State of EKF, b bias vector of MEMS gyros

$$ \mathbf{x} = \begin{bmatrix} {}^{n}_{b}\mathbf{C}_{3} \\ {}^{b}\mathbf{b}^{\omega} \end{bmatrix} = \begin{bmatrix} {}^{n}_{b}C_{31} & {}^{n}_{b}C_{32} & {}^{n}_{b}C_{33} & {}^{b}b^{\omega}_{x} & {}^{b}b^{\omega}_{y} & {}^{b}b^{\omega}_{z} \end{bmatrix}^{T} $$

DCM Based Partial Attitude Estimation.

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{\Gamma}_k \mathbf{w}_k$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k,$$

$$f_k(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \mathbf{I}_3 & -\Delta t\,[\mathbf{C}_3\times]_k \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \Delta t\,[\mathbf{C}_3\times]_k \\ \mathbf{0}_{3\times3} \end{bmatrix} \mathbf{u}_k,$$

$$\mathbf{H} = [g\mathbf{I}_3 \quad \mathbf{0}_{3\times3}].$$

- Covariance of process noise

$$\mathbf{Q}_k = \mathbf{\Gamma} \begin{bmatrix} \sigma_{C_3}^2 \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & (\sigma_b^\omega)^2 \mathbf{I}_3 \end{bmatrix} \mathbf{\Gamma}^T$$

$$= \Delta t^2 \begin{bmatrix} \sigma_{C_3}^2 \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & (\sigma_b^\omega)^2 \mathbf{I}_3 \end{bmatrix}.$$

- Covariance of measurement noise

$${}^b\mathbf{a}_k = {}^b\mathbf{f}_k - g\,{}^n_b\widehat{\mathbf{C}}_{3,k}$$

$$\mathbf{R}_k = \left(\left\|{}^b\mathbf{a}_k\right\| \sigma_a^2 + \sigma_f^2\right) \mathbf{I}_3$$

DCM Based Partial Attitude Estimation.

- The proposed attitude filter only estimates the partial attitude, corresponding to two out of the three Euler angles present in the bottom row of the rotation matrix
- Pitch θ and roll φ angles can be estimated accurately using

$$\theta_k = \arcsin\left(-C_{31,k}\right),$$

$$\phi_k = \operatorname{atan2}\left(C_{32,k}, C_{33,k}\right)$$

- Yaw angle, ψ, can be integrated from bias-corrected angular velocities using previously computed roll, pitch, and yaw angles as a starting point. Error accumulates

$$\psi_k = \operatorname{atan2}\left(C_{21,k}, C_{11,k}\right)$$

DCM Based Partial Attitude Estimation, Temperature Calibration Method

- The gains and biases of the MEMS gyroscope and accelerometer vary over time, a large part of which can be explained by temperature changes in the physical instrument.
- As our proposed IMU uses accelerometers to calibrate gyroscope biases online, accelerometer calibration is essential for reaching accurate measurements. Therefore, temperature- dependent calibration is needed at least for the accelerometers in order to estimate temperature-dependent bias and gain terms for all the sensor axes if there is any possibility of temperature changes in the environment.

DCM Based Partial Attitude Estimation Temperature Calibration Method.

- The used measurement model for each sensor axis $i \in \{x, y, z\}$

$$^{b}f_{\text{meas},i} = p_{\text{gain}}^{f_i}(T)\,^{b}f_i + p_{\text{bias}}^{f_i}(T),$$

$$^{b}\omega_{\text{meas},i} = p_{\text{gain}}^{\omega_i}(T)\,^{b}\omega_i + p_{\text{bias}}^{\omega_i}(T),$$

$$p_{\text{gain/bias}}^{f_i/\omega_i}(T) = a_{\text{gain/bias}}^{f_i/\omega_i}T + b_{\text{gain/bias}}^{f_i/\omega_i}.$$

DCM Based Partial Attitude Estimation, Summary.

- A partial attitude estimation algorithm for low-cost MEMS IMUs using a direction cosine matrix (DCM) to represent orientation.
- The attitude estimate is partial, as only the orientation towards the gravity vector is estimated. The sensor fusion of triaxial gyroscopes and accelerometers was accomplished using an adaptive extended Kalman filter. The filter accurately estimates gyroscope biases online, thus enabling the filter to perform effectively even if the calibration is inaccurate or some unknown slowly drifting bias exists in the gyroscope measurements.
- DCM IMU is made more robust against temporary contact forces by using adaptive measurement covariance in the EKF algorithm.

DCM Based Partial Attitude Estimation.

- The proposed DCM-IMU algorithm should be used with low-cost MEMS sensors when at least one of the following is true: (a) only accelerometer and gyroscope measurements are available, (b) there exist large temporary accelerations, (c) there exist unknown or drifting gyroscope biases, or (d) measurements are collected using a variable sampling rate.
- However, our proposed method should not be used if constant nongravitational accelerations are present, nor would it be needed if gyroscopes are accurately calibrated and no drifting biases are present in the measurements (i.e., an error-free system). Long-term or constant nongravitational accelerations will be mixed with the estimated gravitational force, as there are no external measurements to separate them.

DCM Based Partial Attitude Estimation.

- Finally, our method is best suited for low-cost MEMS sensors with drifting biases and erroneous measurements. It also eliminates the need for commonly used magnetometers while estimating biases for gyroscope measurements.
- The method, however, is not designed to give an absolute heading; instead, it is best suited for measuring absolute roll and pitch angles and a minimally drifting relative yaw angle.