# CS/EE/ME 75(a)

Nov. 26, 2019

Today:

- Dec. 4 Class: Brett Lopez & “flight control stack”
- Tonight: Brief review of
  - estimation,
  - odometry,
  - inertial navigation,
  - SLAM, pose graph

# State Estimation & Mapping

Some Important questions for robot operation and SubT

- Where am I?  How am I oriented?
- How fast am I moving?
- Where can I move: obstacles, or unobstructed directions of motion?
- What is the local geometry of my environment?
- What is the *global* geometry of my environment?
- How do I store information about the environment that I'm working in?
- Have I been here before?
- How do I best localize the "targets"?

State Estimation & Mapping subsystems address these questions, but not always perfectly.

# Estimation & Optimal (Kalman) Filtering

## Observer

Process Dynamics — Measurement Equation

- Given $\dot{x} = f(x, u)$ $\quad y = h(x)$
- Calculate, infer, deduce the state $x$ from measurements $y$
- E.g. the *Luenberger Observer* $\dot{x} = Ax + Bu + L(y - Cx)$

## Estimator

- Given $\dot{x} = f(x, u) + \xi$ $\quad y = h(x) + \omega$
  - $\xi$ represents *process noise/uncertainty* (e.g., gust or unmodeled effects)
  - $\omega$ represents *measurement noise/uncertainty*
- *Estimate* (in an *optimal)* way the state $x$ based on
  - measurements $y$
  - dynamic and measurement models
  - noise model(s).

# Estimation Overview (continued)

## Noise & Uncertainty Models for Estimation

$$\dot{x} = f(x,u) + \xi \qquad y = h(x) + \omega(t)$$

- Set-based : $\xi \in \Xi \qquad \omega \in \Omega$
- Stochastic $\xi$ and $\omega$ are random processes governed by $p(\xi)$ and $p(\omega)$

## Why Estimation ?

- ***MANY*** important problems can be posed as estimation problems. *E.g.:*
  - Inertial Navigation
  - Localization, Mapping, SLAM
  - Sensor Processing and Fusion
  - Tracking and Prediction
  - Parameter Estimation

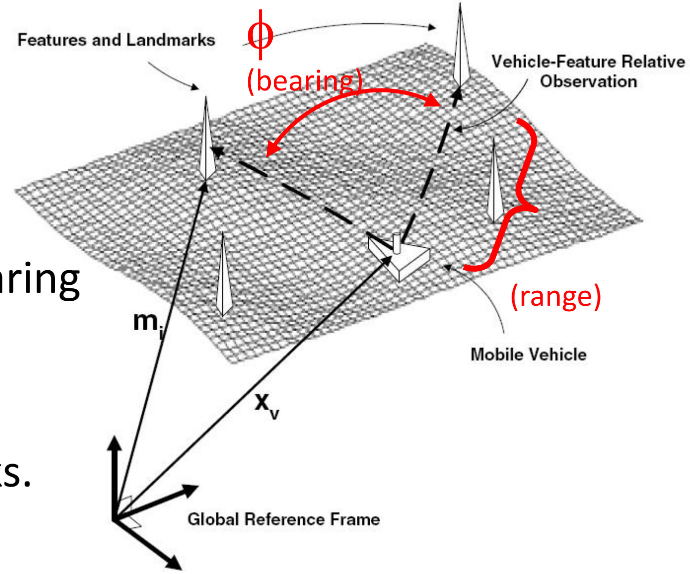
} Specialized estimation techniques & literatures

# Landmark–based Localization & Mapping

**Localization:** A robot explores an static environment where there are known *landmarks*.

- radio beacons (Lojack)
- UWB ranging
- bar-code decals

**Goal:** Estimate the robot's position given range/bearing measurements




**Mapping:** A robot explores an  unknown static environment where there are identifiable landmarks.

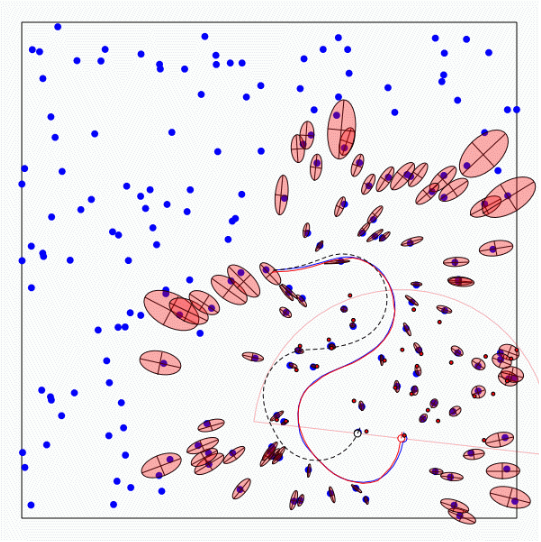
- doors, windows, light fixtures
- linoleum floor patterns

Build a *map* (estimate all landmark positions), assuming robot has GPS-like localization

# ***SLAM***: Simultaneous Localization & Mapping

## Given:

- Robot motion model: $\dot{x} = f(x,u) + \xi$
- The robot's controls, *u*
- Measurements (e.g., range, bearing ) of nearby features: $y = h(x) + \omega$



## Estimate:

- Map of landmarks ($x_L$)
- Robot's current *pose*, $x_R$, & its path
- Uncertainties in estimated quantities

$$\vec{x} = \begin{bmatrix} \vec{x}_R \\ \vec{x}_L \end{bmatrix} = \begin{bmatrix} \vec{x}_R \\ x_{L_1} \\ \vdots \\ x_{L_N} \end{bmatrix}$$

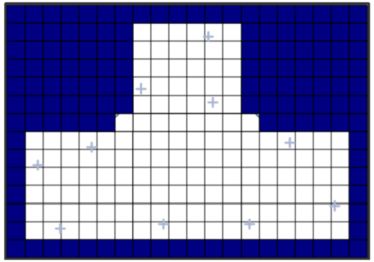
Robot state

Landmark positions

# Three Major Map Models

## Grid-Based:

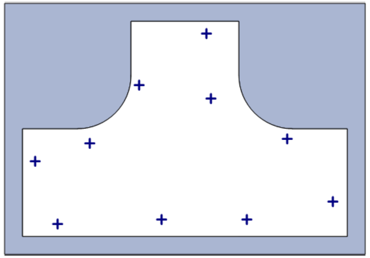
Collection of discretized obstacle/free-space pixels



Elfes, Moravec, Thrun, Burgard, Fox, Simmons, Koenig, Konolige, etc.

## Feature-Based:

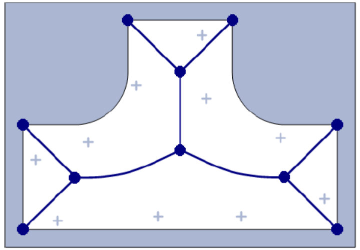
Collection of landmark locations and correlated uncertainty



Smith/Self/Cheeseman, Durrant–Whyte, Leonard, Nebot, Christensen, etc.

## Topological:

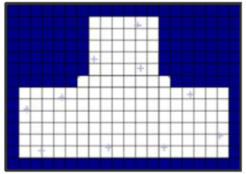
Collection of nodes and their interconnections

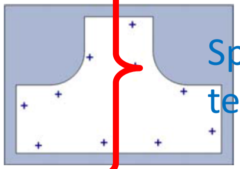


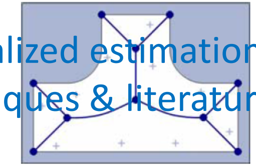
Kuipers/Byun, Chong/Kleeman, Dudek, Choset, Howard, Mataric, etc.

# Three Major Map Models

| | Grid-Based | Feature-Based | Topological |
|---|---|---|---|
| **Resolution vs. Scale** | Discrete localization | Arbitrary localization | Localize to nodes |
| **Computational Complexity** | Grid size and resolution | Landmark covariance ($N^2$) | Minimal complexity |
| **Exploration Strategies** | Frontier-based exploration | No inherent exploration | Graph exploration |





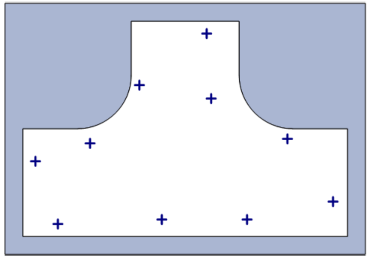


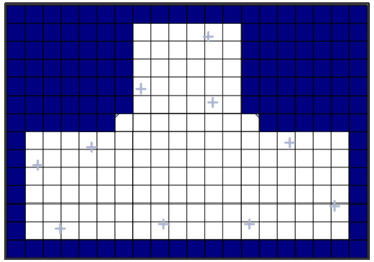
Specialized estimation techniques & literatures

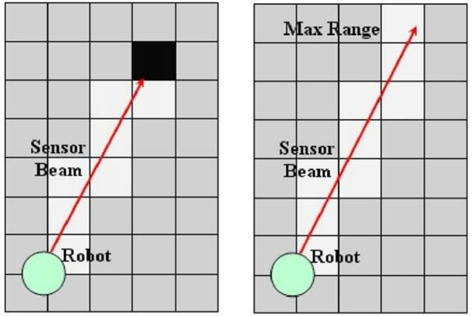
# Gmapping

**Occupancy Grid:** “map” is a grid of “cells”: $\{x_{i,j}^{m}\}$

- $x_{i,j}^{m} = 0$ if cell (i,j) is empty; $x_{i,j}^{m} = 1$ if cell (i,j) is occupied
- $p\left(x_{k+1}^{r}, \{\mathrm{x}_{i,j}^{\mathrm{m}}\}_{k+1} \middle| \mathrm{x}_{1:k}^{\mathrm{r}}, \{\mathrm{x}_{\mathrm{i,j}}^{\mathrm{m}}\}_{k}, \mathrm{y}_{1:k+1}\right)$ (estimate cell occupancy probability)





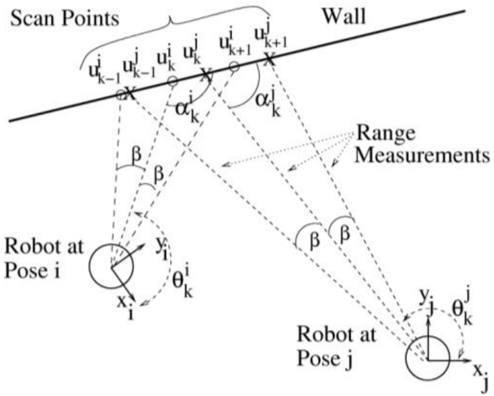


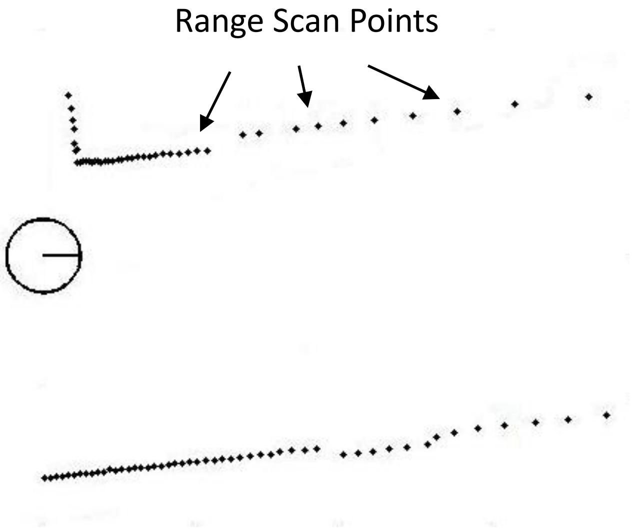
(a) (b)

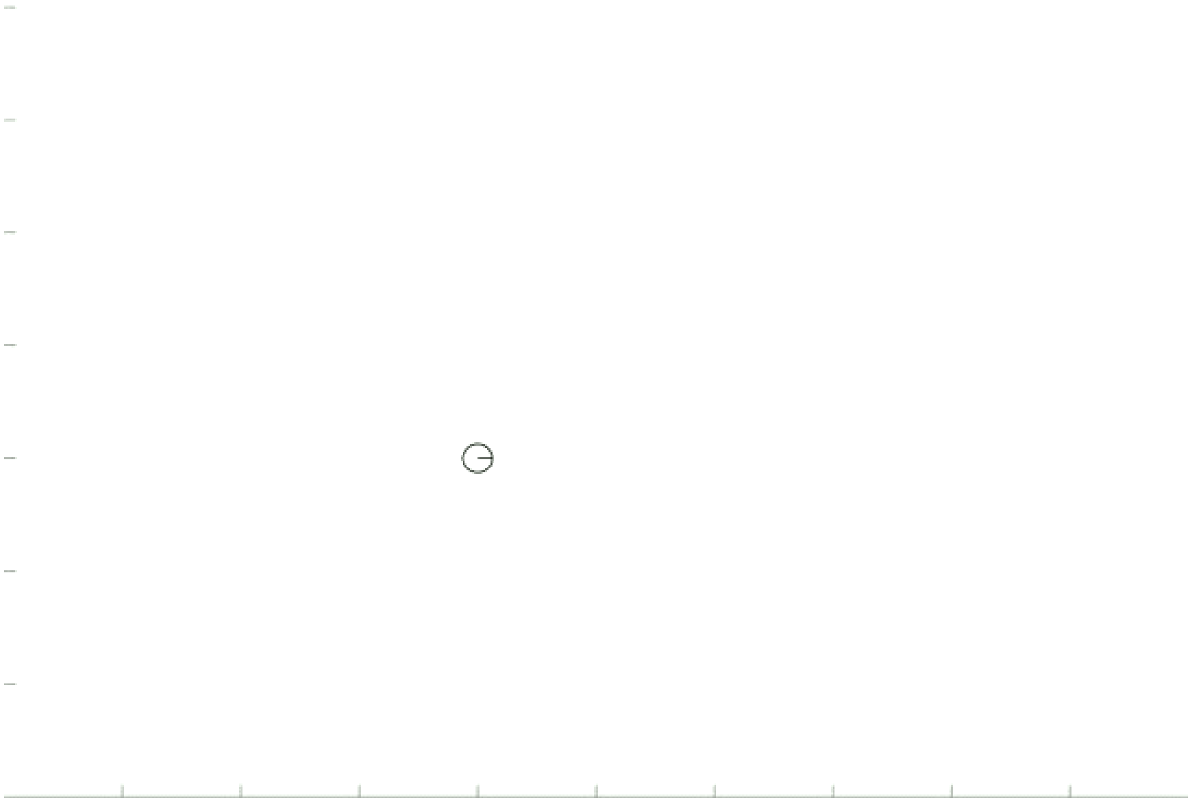
**Gmapping:**

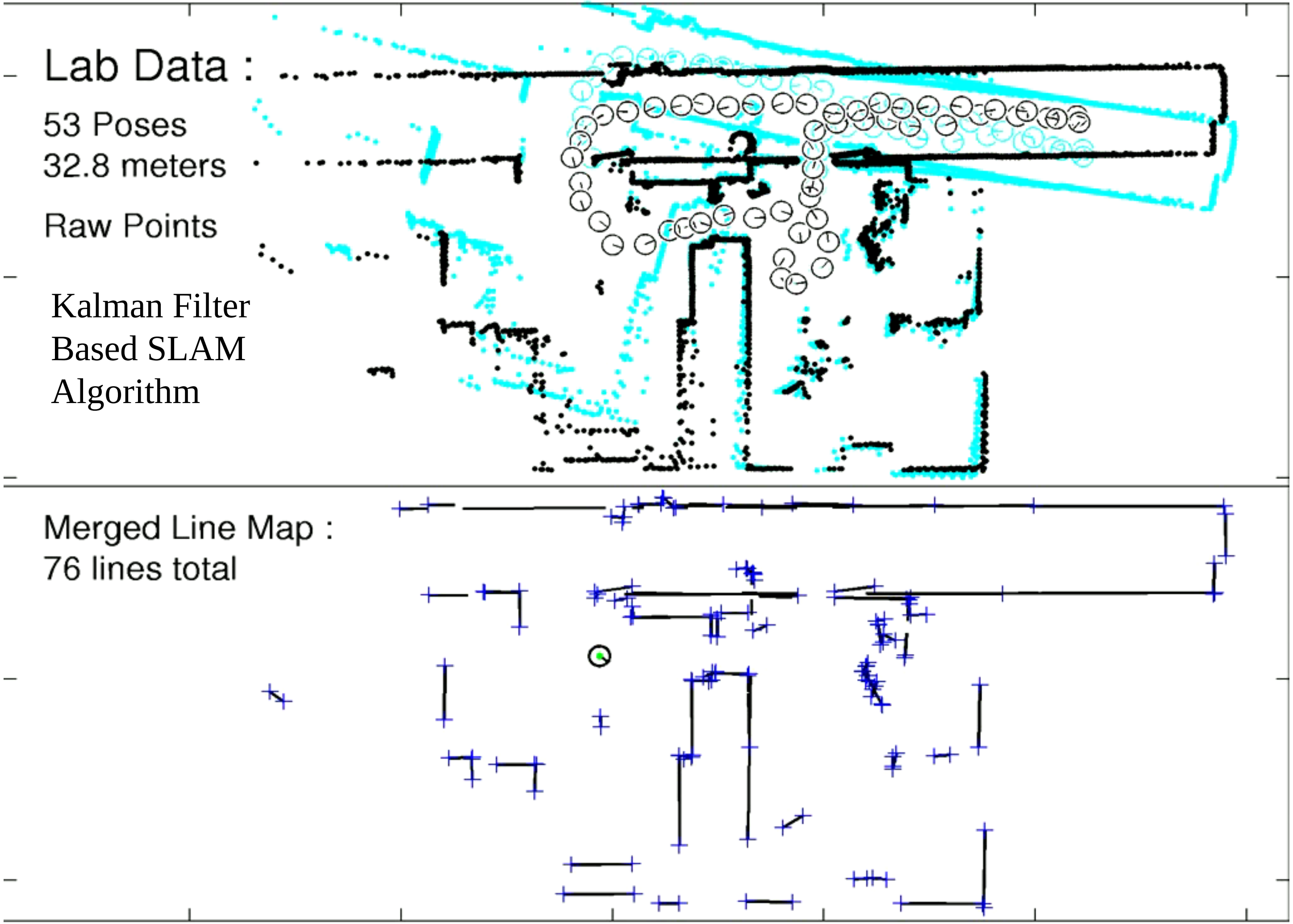
- Uses a *Rao-Blackwellized* particle filter for estimator
- Actually computes $p\left(x_{1:T}^{r}, \{x_{i,j}^{m}\} \middle| \mathrm{x}_{1:k}^{\mathrm{r}}, \mathrm{x}_{\mathrm{k}}^{\mathrm{m}}, \mathrm{y}_{1:k+1}\right)$

# Line-Based SLAM: LADAR





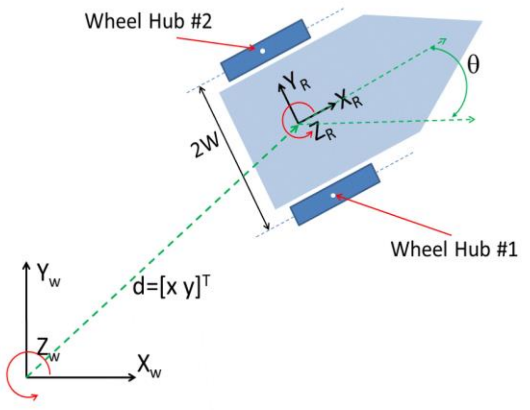

Lab Data :
53 Poses
32.8 meters
Raw Points
Kalman Filter
Based SLAM
Algorithm
Merged Line Map :
76 lines total

# Odometry & "Wheel Odometry"

**Definition:** Use of on-board sensors to estimate the change in a moving vehicle's position/orientation over time.

- **Wheel Odometry:** use motion of the robot wheels to estimate robot displacement. Need a "kinematic model"
- **Inertial Odometry:** use sensors that measure inertial data (acceleration and rates of rotation) to estimate robot displacement.
- **Visual odometry:** use vision information to estimate robot displacement
  - Monocular & stereo cameras
  - Lidar
  - RGB-D
- **Visual/Inertial Odometry (VIO):** "Fuse" data from a vision sensor and the inertial sensors to get better estimates of displacement

# Example: differential drive robot

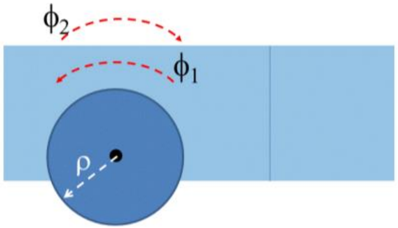



We can derive (see notes on syllabus)

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \frac{\rho}{2}\begin{bmatrix}(\dot{\phi}_2 - \dot{\phi}_1)\cos\theta\\ (\dot{\phi}_2 - \dot{\phi}_1)\sin\theta\\ -(\dot{\phi}_2 + \dot{\phi}_1)/W\end{bmatrix}$$

Which is equivalent to:

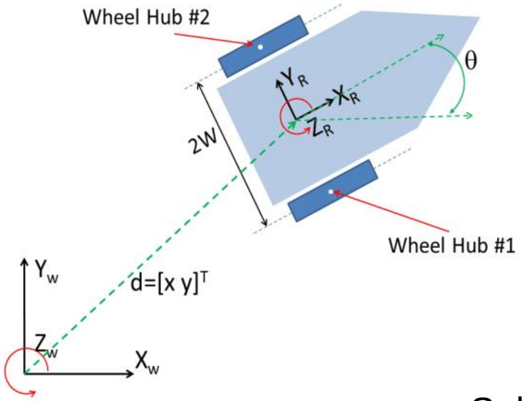
$$\begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \frac{\rho}{2}\begin{bmatrix}(\dot{\phi}_2 - \dot{\phi}_1)\\ 0\\ -(\dot{\phi}_2 + \dot{\phi}_1)/W\end{bmatrix}$$

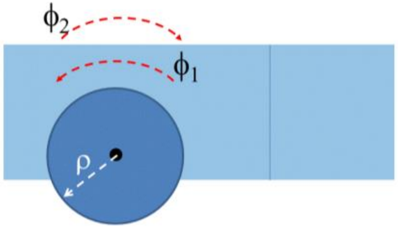



“Integrate” the kinematic equations w.r.t. time:

$$\begin{bmatrix}x(t)\\ y(t)\\ \theta(t)\end{bmatrix} = \begin{bmatrix}x(0)\\ y(0)\\ \theta(0)\end{bmatrix} + \int_0^t \begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix} dt = \begin{bmatrix}x(0)\\ y(0)\\ \theta(0)\end{bmatrix} + \frac{\rho}{2}\int_0^t \begin{bmatrix}(\dot{\phi}_2(t) - \dot{\phi}_1(t))\cos\theta(t)\\ (\dot{\phi}_2(t) - \dot{\phi}_1(t))\sin\theta(t)\\ -(\dot{\phi}_2(t) + \dot{\phi}_1(t))/W\end{bmatrix} dt$$

# Example: differential drive robot




To practically integrate these equations, we can use a *backward difference* approximation scheme:

$$\dot{x}(t_k) \simeq \frac{x(t_k) - x(t_{k-1})}{t_k - t_{k-1}} = \frac{x(t_k) - x(t_{k-1})}{\Delta t}$$

$$\dot{y}(t_k) \simeq \frac{y(t_k) - y(t_{k-1})}{t_k - t_{k-1}} = \frac{y(t_k) - y(t_{k-1})}{\Delta t}$$

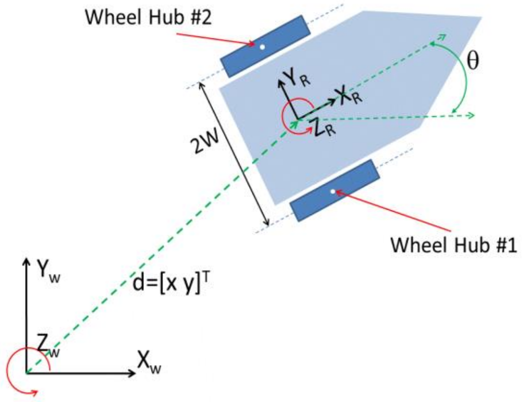
$$\dot{\theta}(t_k) \simeq \frac{\theta(t_k) - \theta(t_{k-1})}{t_k - t_{k-1}} = \frac{\theta(t_k) - \theta(t_{k-1})}{\Delta t}$$

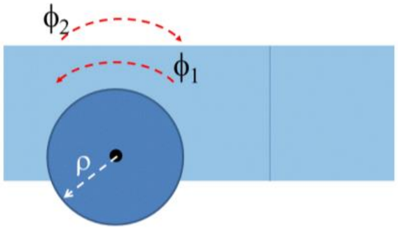
Substituting into the integration and rearranging yields:

$$\begin{bmatrix} x(t_k) \\ y(t_k) \\ \theta(t_k) \end{bmatrix} = \begin{bmatrix} x(t_{k-1}) \\ y(t_{k-1}) \\ \theta(t_{k-1}) \end{bmatrix} + \frac{\rho}{2} \begin{bmatrix} (\Delta\phi_2 - \Delta\phi_1)\cos\theta(t_{k-1}) \\ (\Delta\phi_2 - \Delta\phi_1)\sin\theta(t_{k-1}) \\ -(\Delta\phi_2 + \Delta\phi_1)/W \end{bmatrix}$$

Where $\Delta\phi_i = \phi_i(t_k) - \phi_i(t_{k-1})$

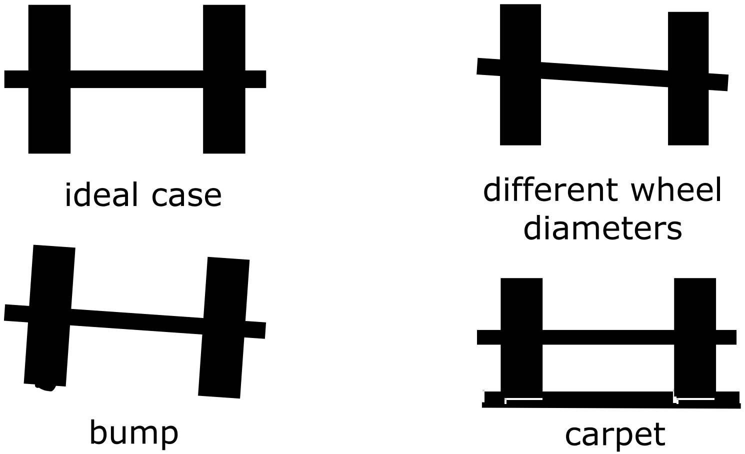
# Example: differential drive robot




Sources of Error:

- Wheels slip on the ground (violating assumption of derivation)
- Equations of motion are an approximation (wheels don't have zero thickness)
- Backwards-difference is an approximation
- Quantization/noise error in wheel sensor measurements



# Inertial Navigation

**Definition:** Given the vehicle's initial position and velocity, estimate the vehicle's current position and velocity relative to the starting point, using only *proprioceptors* that measure rigid body inertial properties (e.g., accelerometers and gyroscopes).

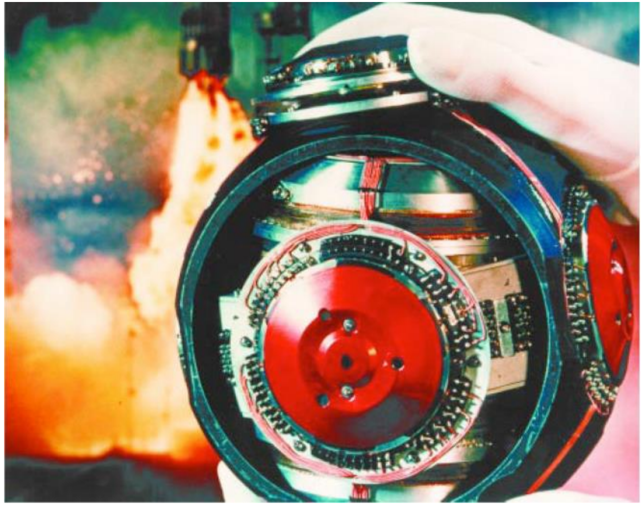
**Inertial Measurement Unit (IMU):** Contains a cluster of sensors attached to a common base.

- accelerometers (typically 3 or more along orthogonal axes) measure accelerations of the sensor's body-fixed reference frame,
- gyroscopes measure the rates of sensor rotation about three orthogonal axes.
- An optional *magnetometer* measure's the earth's magnetic field (i.e., a compass)

**Basic Concept:** Integrate accelerometer twice to obtain velocity and position. Integrate gyro readings to estimate orientation.

# Older types of IMUs

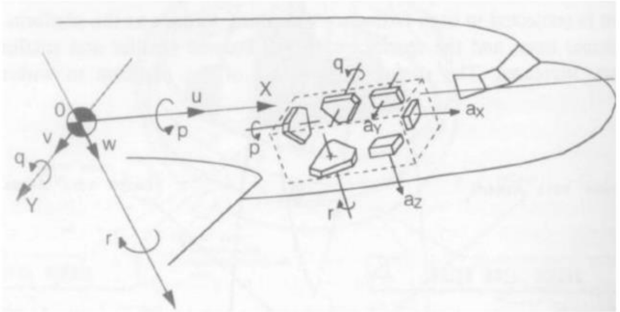
**Gimbaled:** The sensor suite is mounted on a platform in a gimbal.



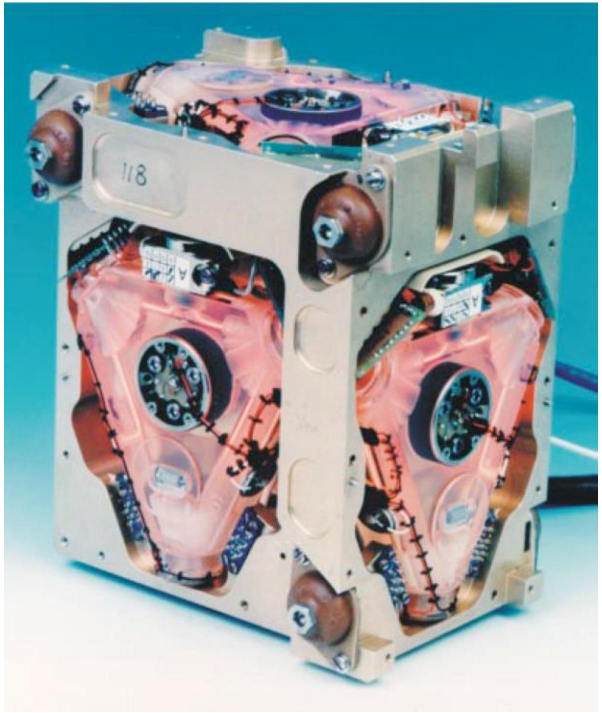
**Mechanical Gyro:** A spinning wheel will resist any change to its angular momentum vector. Maintains constant orientation if in a frictionless gimbal

# Modern IMUs

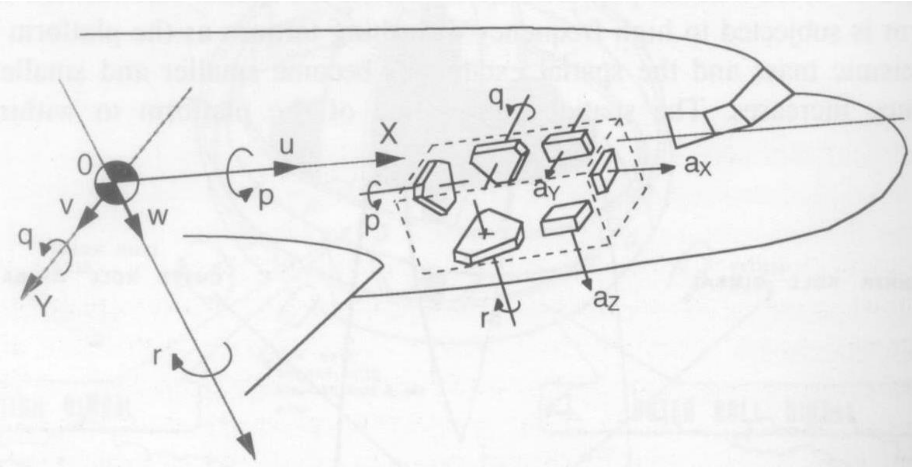
**Strapdown:** Mount sensor suite rigidly to the moving object of interest.

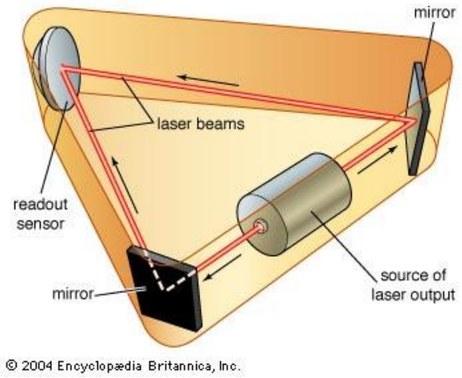






# Inertial Navigation




Laser Ring Gyro

Uses Doppler effect to measure rotation rate

$$\mathrm{x} = \begin{bmatrix} p \\ \phi \\ \dot{p} \\ \dot{\phi} \\ \ddot{p} \\ \ddot{\phi} \end{bmatrix}$$

$$y = \begin{bmatrix} \dot{\phi} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} x = Hx$$

# Strapdown Gyros (1)

**The Sagnac-effect.** The inertial characteristics of light can also be utilized, by letting two beams of light travel in a loop in opposite directions. If the loop rotates clockwise, the clockwise beam must travel a longer distance before finishing the loop. The opposite is true for the counter-clockwise beam. Combining the two rays in a detector, an interference pattern is formed, which will depend on the angular velocity.

The loop can be implemented with 3 or 4 mirrors (*Ring Laser Gyro*), or with optical fibers (*Fiber Optic Gyro*).

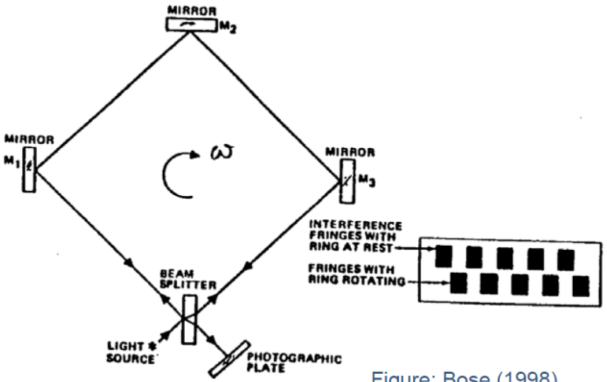
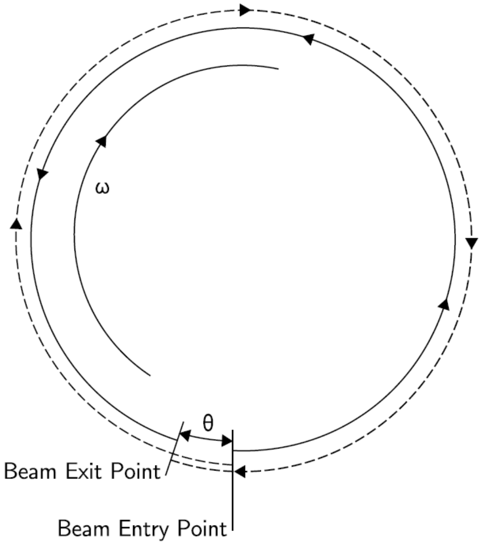



Figure: Bose (1998)

# Strapdown Gyros (2)

**The Sagnac-effect.** The inertial characteristics of light can also be utilized, by letting two beams of light travel in a loop in opposite directions. If the loop rotates clockwise, the clockwise beam must travel a longer distance before finishing the loop. The opposite is true for the counter-clockwise beam. Combining the two rays in a detector, an interference pattern is formed, which will depend on the angular velocity.

The loop can be implemented with 3 or 4 mirrors (*Ring Laser Gyro*), or with optical fibers (*Fiber Optic Gyro*).



# Strapdown Gyros (3)

**Coriolis Effect:** When a mass vibrating in a radial direction is rotated, the Coriolis effect will cause new vibrations perpendicular to the original vibration axis.

- Cheaper MEMS gyros use this principle (“tuning fork” or “wineglass”)
- Less accurate than ring or fiber gyro

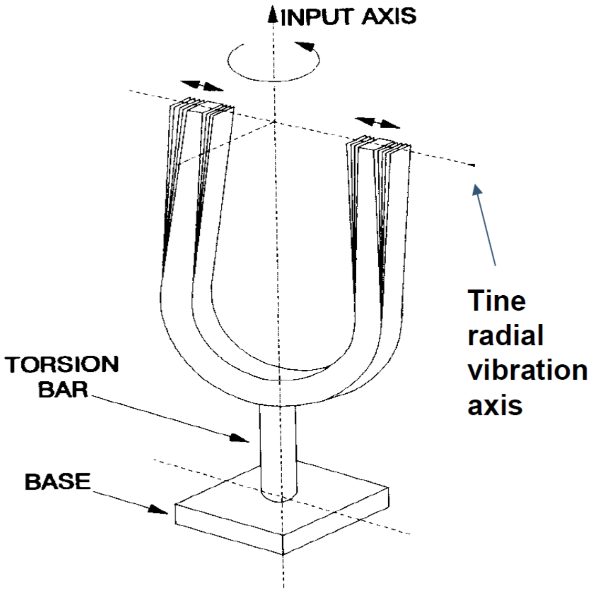
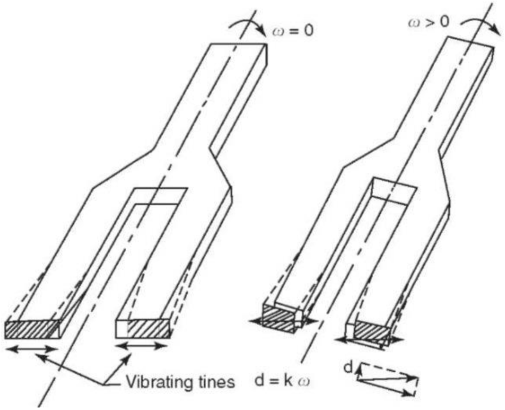
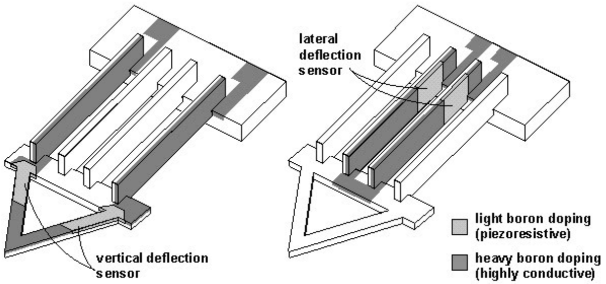



Figure: Titterton & Weston (1997)

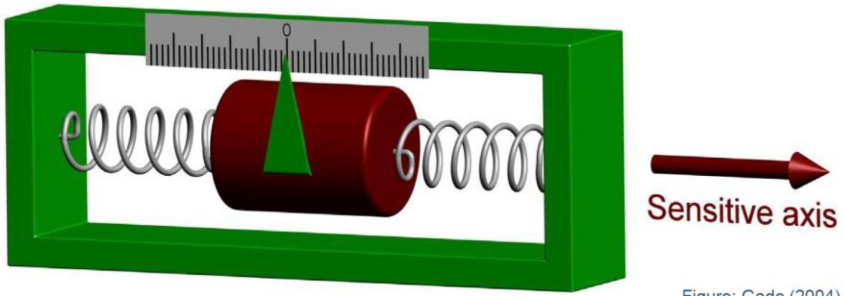


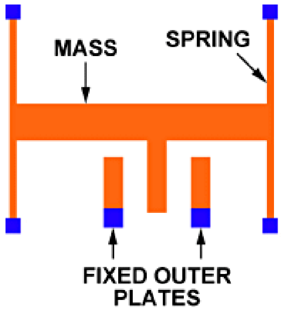



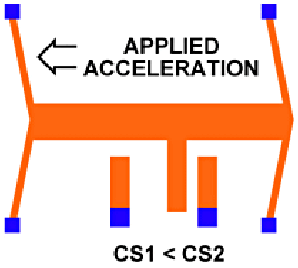
# Accelerometers

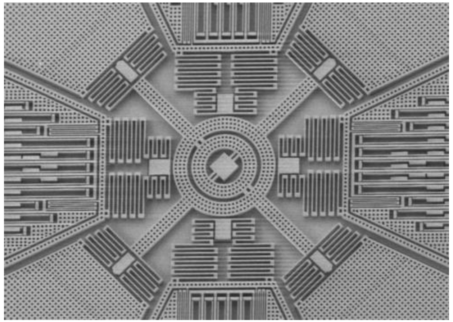
**Vibration:** deflection of a mass on a spring yields simple acceleration signal

- Greater accuracy, dynamic range, and linearity can be realized by keeping mass close to nominal position—control forces are proportional to acceleration.

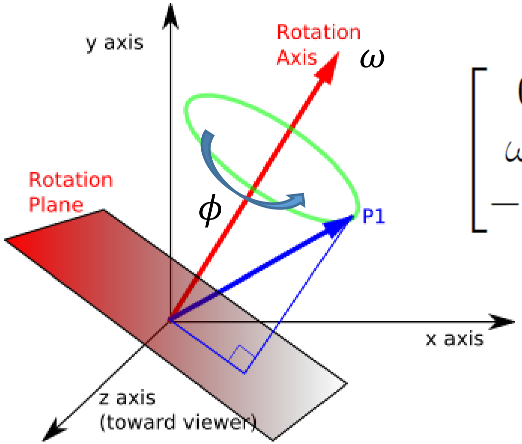


# Summary of Inertial Odometry Equations




$$\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

$$R(\phi, \vec{\omega}) = I + (\sin\phi)\hat{\omega} + (1 - \cos\phi)\hat{\omega}^2.$$

Gyro gives us measurements of angular velocity in *body coordinates*:

$$\vec{\Omega}^1 = \begin{bmatrix} \Omega_x^1 \\ \Omega_y^1 \\ \Omega_z^1 \end{bmatrix}$$

$$\begin{aligned} R_{WB}(t_{k+1}) &= R_{WB}(t_k) e^{\Delta\phi(t_{k+1})\hat{\omega}(t_{k+1})} \\ &= R_{WB}(t_k)\left[I + \sin(\Delta\phi(t_{k+1}))\hat{\omega}(t_{k+1}) + (1 - \cos(\Delta\phi(t_{k+1})))\hat{\omega}^2(t_{k+1})\right] \end{aligned}$$

$$\begin{aligned} \Delta\phi(t_{k+1}) &= ||\vec{\Omega}(t_{k+1}||\Delta t \\ \hat{\omega}(t_{k+1}) &= ||\vec{\Omega}(t_{k+1})||^{-1}\, \vec{\Omega}(t_{k+1}) \end{aligned}$$

# Summary of Inertial Odometry Equations

Convert acceleration measured in body frame to “world” frame

$$\vec{a}^W(t) = R_{WB}(t)\,\vec{a}^B(t)$$

Calculate velocity of body holding the accelerometer (theory, practice)

$$\vec{v}_{WB}^W = \vec{v}_{WB}^W(t=0) + \int_0^t \vec{a}^W(\tau)d\tau \qquad \vec{v}_{WB}^W(t_{k+1}) = \vec{v}_{WB}^W(t_k) + R_{WB}(t_{k+1})\vec{a}^B(t_{k+1})\Delta t$$

Calculate the position of the body holding the accelerometer (theory, practice)

$$\vec{p}_{WB}^W = \vec{p}_{WB}^W(t=0) + \int_0^t \vec{v}_{WB}^W(\tau)d\tau \qquad \vec{p}_{WB}^W(t_{k+1}) = \vec{p}_{WB}^W(t_k) + \vec{v}_{WB}^W(t_{k+1})\,\Delta t$$

# Terrestrial Navigation

**Corrections** need to be added to inertial navigation when the vehicle is operating or navigation on earth, since *earth is not an inertial system*

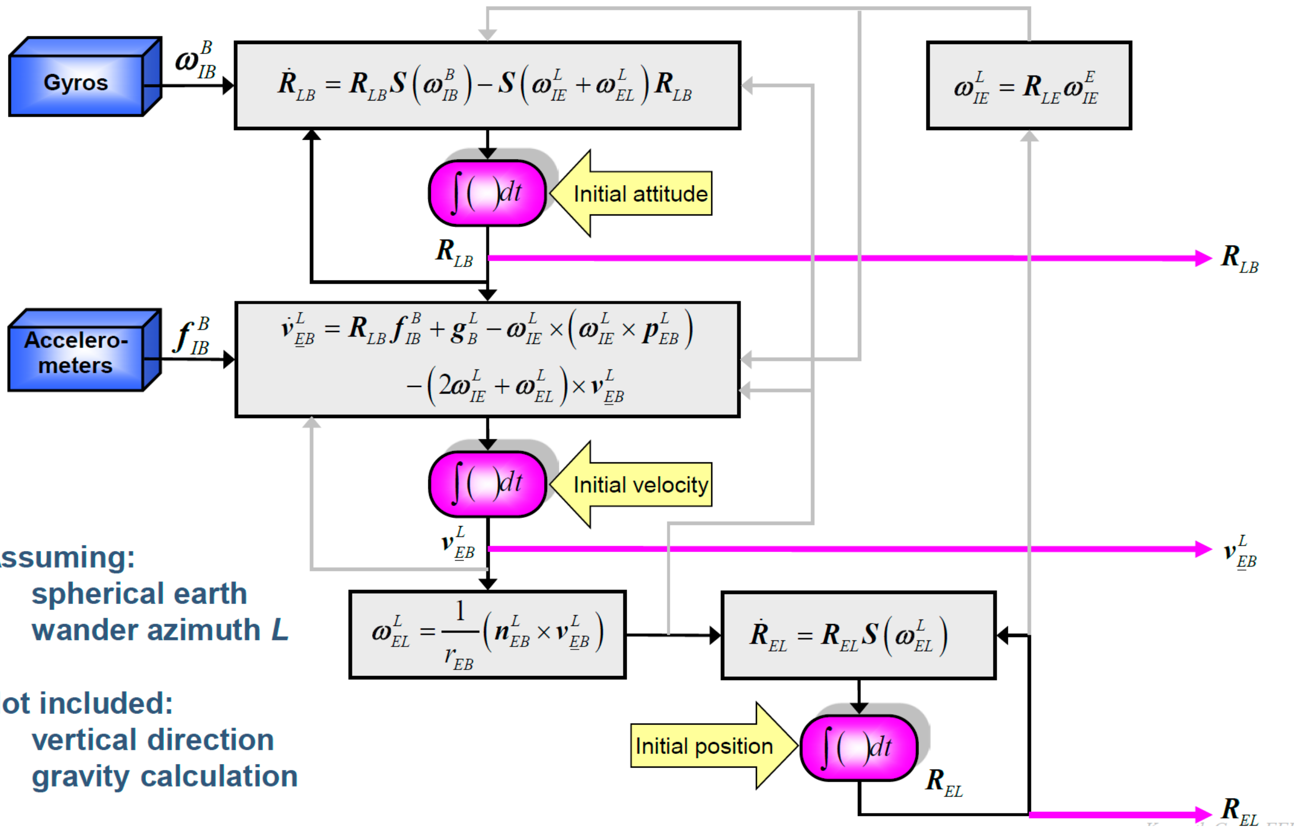
**Gyros:**

- The earth rotates: $\omega_{EB} = \omega_{IB} - \omega_{IE}$ ($\omega_{IE} =$ earth's rotation in space)

**Accelerometers**

- also measure *gravitational acceleration*
  - Let $a_{IB}$ denote the acceleration of the body measured relative to an inertial frame.
  - Let $g_B$ denote the gravitational acceleration
  - The *specific force,* $f_{IB}$, measures the acceleration in an inertial frame: $f_{IB} = a_{IB} - g$
- Also measure centrifugal and Coriolis forces due to movement in earth's rotating frame.

# Terrestrial Navigation




**Assuming:**
- spherical earth
- wander azimuth *L*

**Not included:**
- vertical direction
- gravity calculation

# Important coordinate frames

| Frame symbol | Description |
| --- | --- |
| *I* | Inertial |
| *E* | Earth-fixed |
| *B* | Body-fixed |
| *N* | North-East-Down (local level) |
| *L* | Local level, wander azimuth (as *N*, but not north-aligned => nonsingular) |

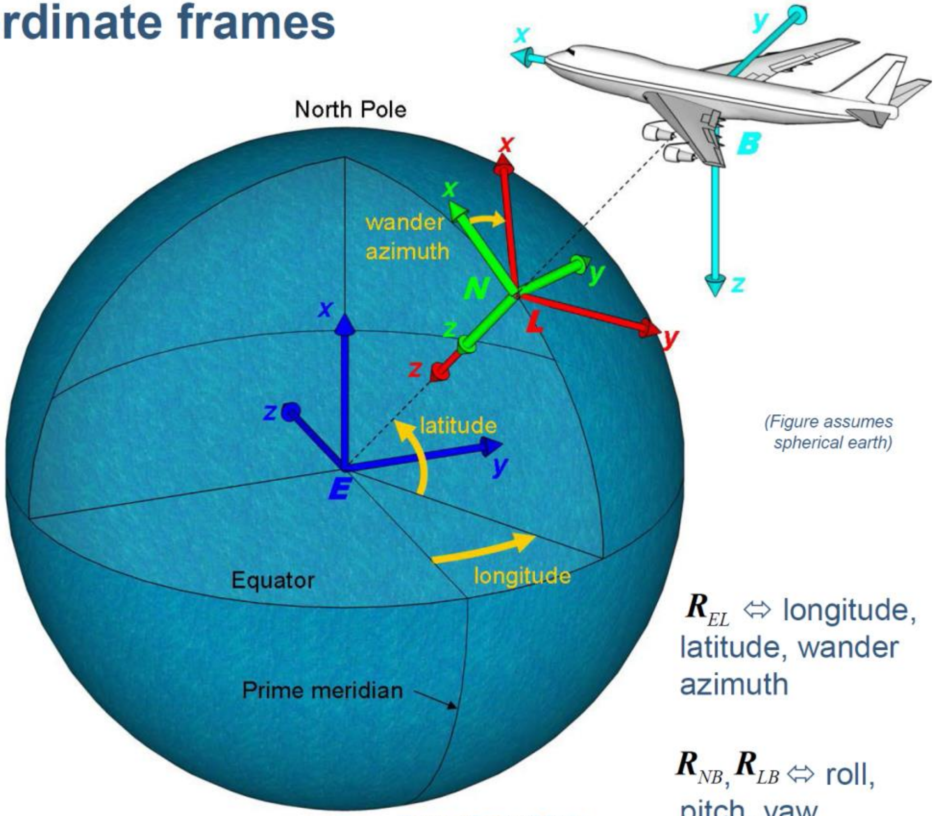



Figure: Gade (2008)

(Figure assumes spherical earth)

$\boldsymbol{R}_{EL}$ ⇔ longitude, latitude, wander azimuth

$\boldsymbol{R}_{NB}, \boldsymbol{R}_{LB}$ ⇔ roll, pitch, yaw

# Homework

**Team Tasks:** (all unit levels)

- Skim through the notes on the course website