Autonomous Micro Aerial Vehicles: Design, Perception, and Control

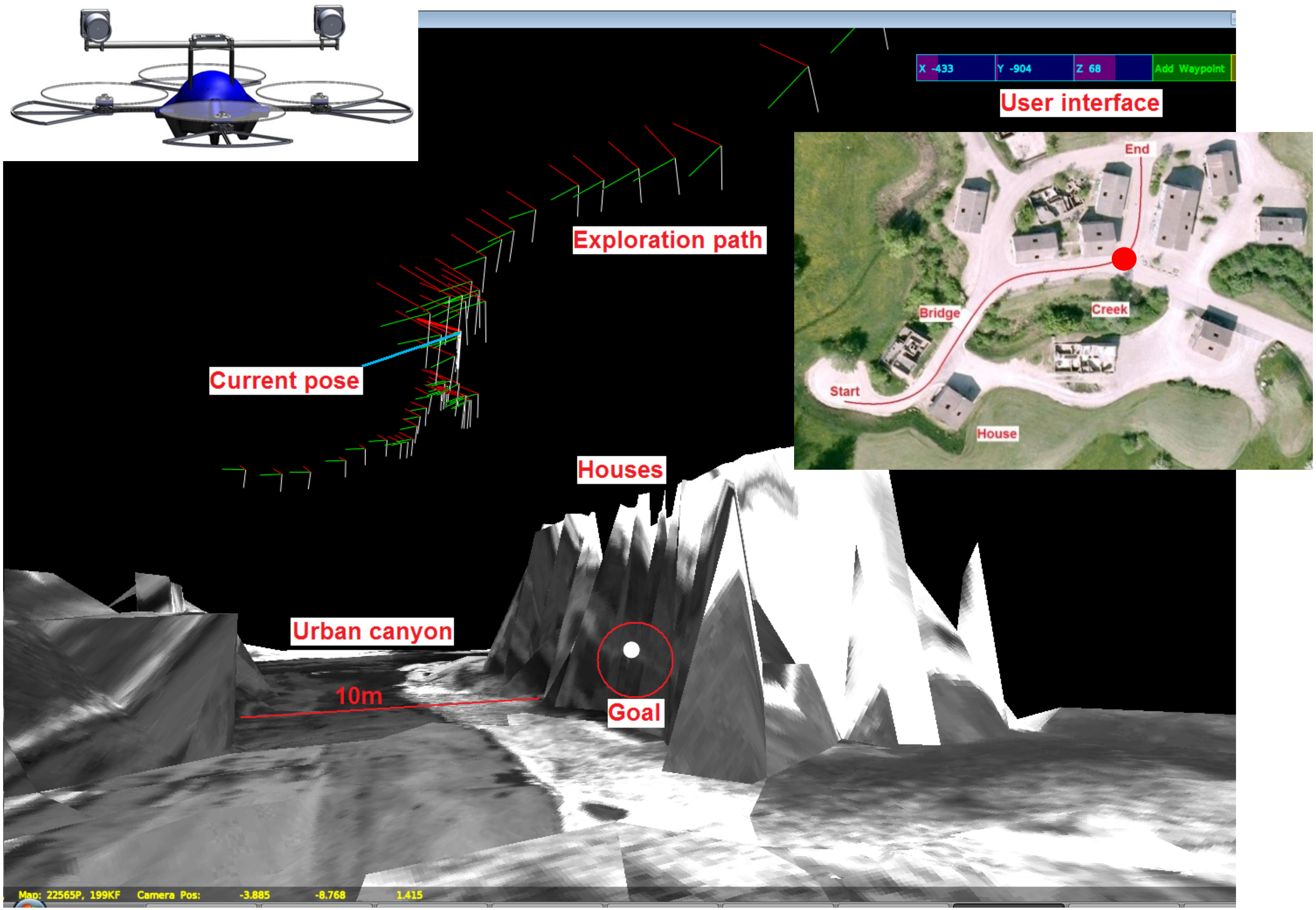
Robot Sensors

Sensor Fusion

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Autonomous Systems Lab

Markus Achtelik, Laurent Kneip, Stephan Weiss

Perception & Navigation



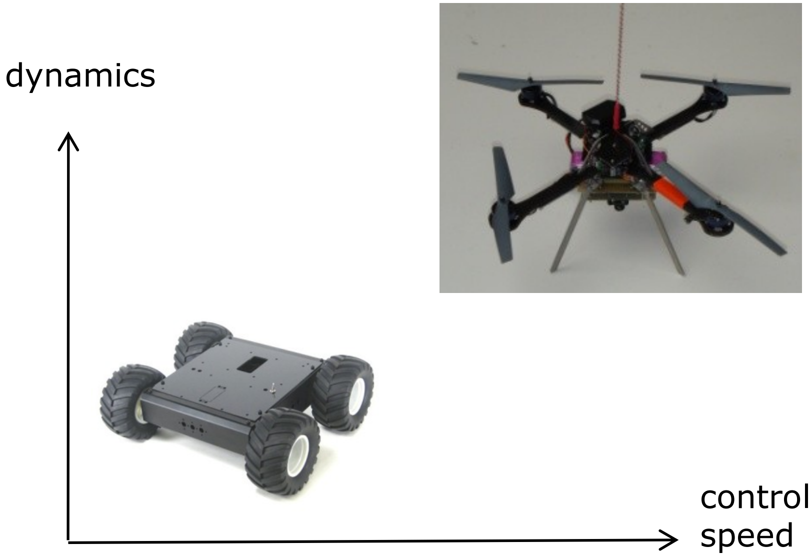
Robot Sensors: The Need of a Pose Estimate

Efficiency

Limited calculation power on MAVs

Every 10 grams need 1W to lift

Sensors with high information density



Real-Time State Estimation

There is no "stop" position like on ground vehicles

Agile and unstable platforms such as quadcopters need high-speed control

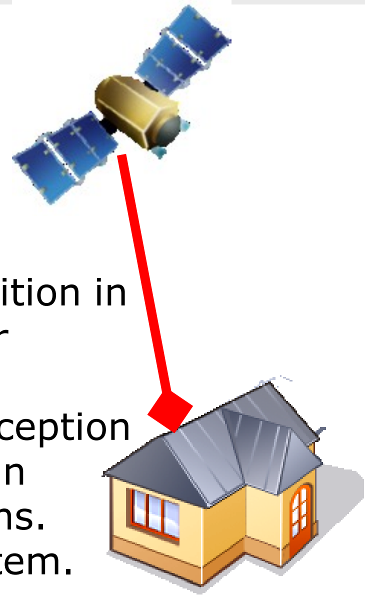
Contents

a) Overview
- Sensors and their usage

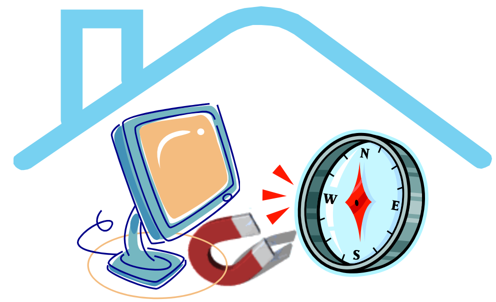
b) Sensor Types
- Global Positioning System (GPS)
- Magnetic-field Sensors
- Accelerometers
- Gyroscopes
- Barometric
- Airspeed

c) Quaternions
- Overview and basics
- Calculation rules
- Conversions
- Derivatives & Integration

Sensors and Their Usage

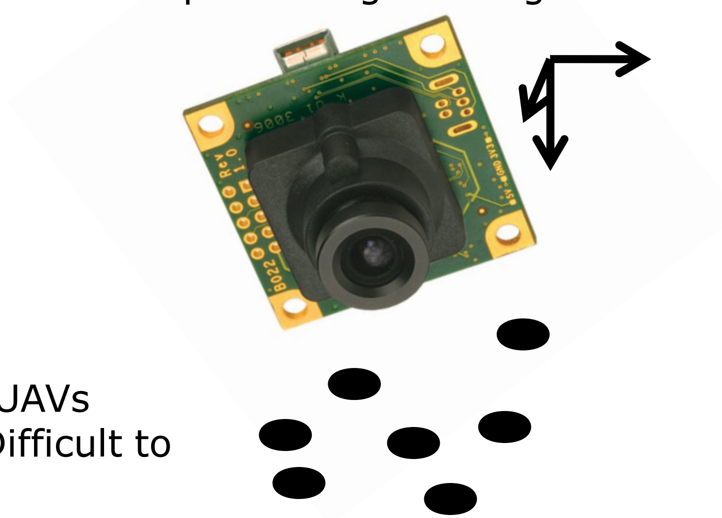


GPS:
Absolute position in clear outdoor areas.
No or bad reception indoors and in urban canyons.
External system.

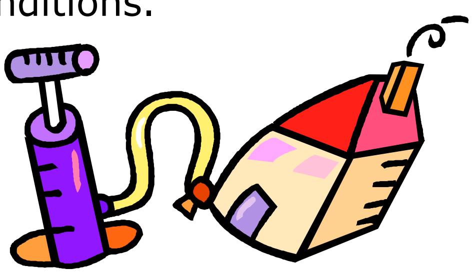


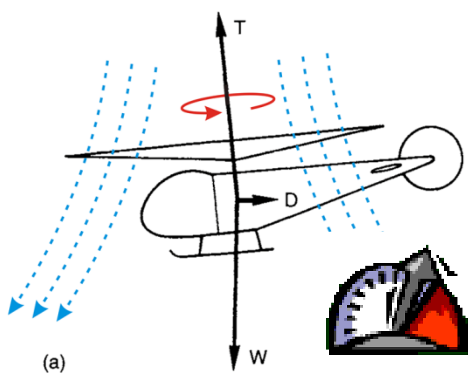
Magnetic field sensors:
2 Absolute angles in open areas.
Disturbances from (electric) objects.

Cameras:
Vast information (angular and intensities/color).
Salient patterns must be visible.
Susceptive to light changes.



Barometric / Pressure sensors:
Absolute altitude measurement.
Sudden pressure changes in indoor environments. Changing weather conditions.





Airspeed sensors:
Reliable for fix wing UAVs
Application critical. Difficult to use on helicopters.

Sensor Types: Global Positioning System

"The Global Positioning System (GPS) is a space-based global navigation satellite system that provides reliable location and time information in all weather and at all times and anywhere on or near the Earth when and where there is an unobstructed line of sight to four or more GPS satellites." [Wikipedia]

Full Operational Capability in April 1995 (24 Satellites)
Since March 2008: 31 actively broadcasting satellites (nonuniform arrangement).

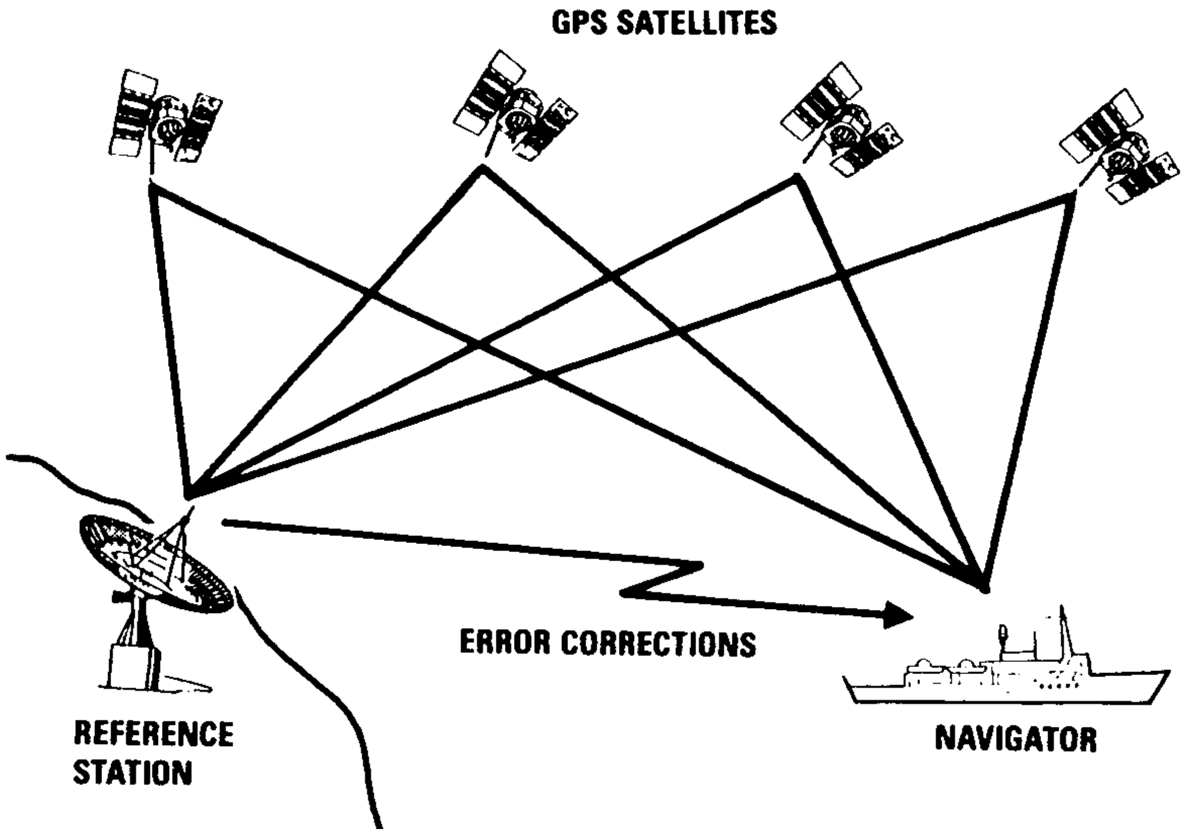
Carrier frequency is about 1.5GHz, clouds are +/- transparent

Accuracy: Difficult to assess. Different factors play a role:

1. Signal quality (atmosphere conditions, # of visible satellites)
2. Support method: for ex. differential GPS (DGPS)
3. Observation time

DGPS: differential GPS:

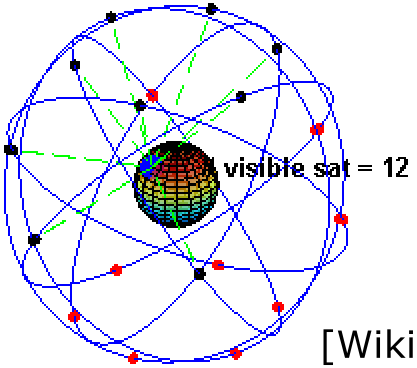
Ground station at known location sends error corrections to the mobile user



Sensor Types: Global Positioning System

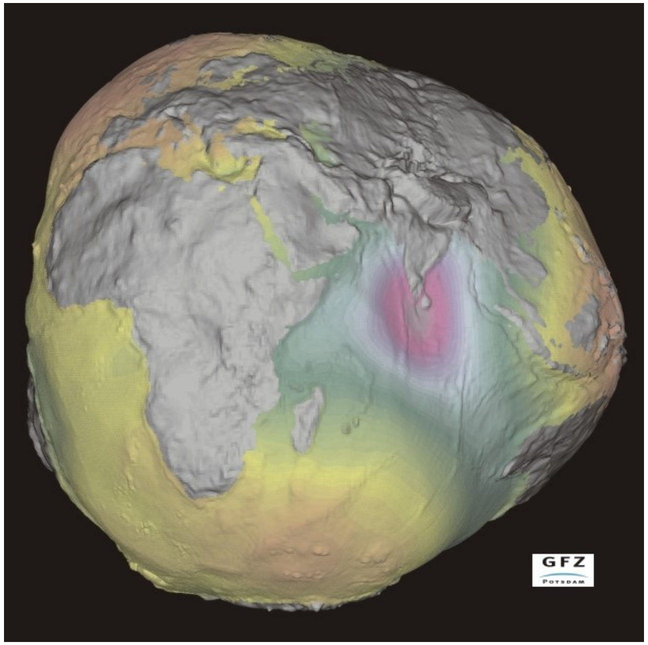
GPS Segments:

- Space Segment:
  - 31 orbiting satellites with atomic clocks
  - About 8 always visible
- Control Segment
  - Master control for satellites
  - Update satellite position (Kalman Filter) and atomic clocks
- User Segment
  - General GPS receiver with antenna and precise clock (crystal, less precise than atomic clock!)
  - Geoid model for height estimation



[Wikipedia]

[GFZ]



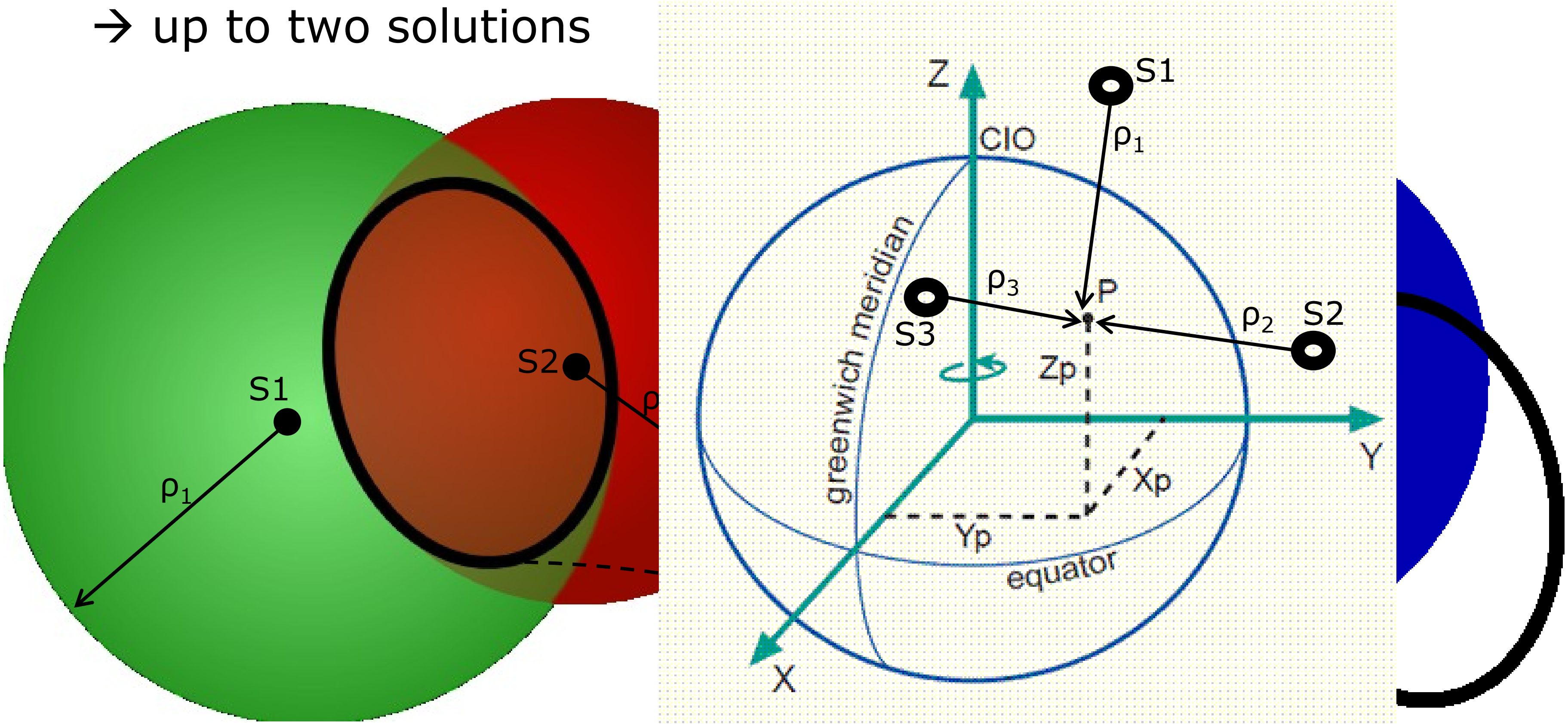
Navigation Equations: Intuition

Basic Principle:

Distance to 3 satellites is known as ρ_i.

Intersect 3 spheres with radii ρ_i and satellite position as center

→ up to two solutions



Navigation Equations: Mathematics 1

Distance to satellite $\rho_i = v \cdot t = c \cdot (t_r - t_i)$

- c: speed of light
- t_r: time signal received
- t_i: time signal sent (satellite i)

Navigation equation:

$$\rho_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} + c \cdot \Delta t = c \cdot (t_r - t_i)$$

- (x_i, y_i, z_i): position of satellite i
- (x, y, z): receiver position
- Δt: receiver time offset to GPS time

➔ 4 unknowns require 4 equations (i.e. satellites)

Navigation Equations: Mathematics 2

Solution of the Navigation Equation System (with $b = c \cdot \Delta t$)

$$\rho_1 = \sqrt{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2} + b = c \cdot (t_{r1} - t_1)$$

$$\rho_2 = \sqrt{(x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2} + b = c \cdot (t_{r2} - t_2)$$

$$\rho_3 = \sqrt{(x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2} + b = c \cdot (t_{r3} - t_3)$$

$$\rho_4 = \sqrt{(x_4 - x)^2 + (y_4 - y)^2 + (z_4 - z)^2} + b = c \cdot (t_{r4} - t_4)$$

Least squares solution using Bancroft's method

Sensor Types: Global Positioning System

Error Sources

Influence of receiver time offset $b = c \cdot \Delta t$

c = 299792458 m/s = 0.299792458 m/nsec

➔ 1ns offset reflects in 30cm distance error (1ms in 300km!)

Atmospheric Distortions and Satellite Errors

Satellite signal is delayed as a function of the atmosphere state
Satellite information is inaccurate (position, time sync)

➔ Phase measurements on different receivers eliminate errors

Sensor Model

Recall:

$$\rho_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} + c \cdot \Delta t = c \cdot (t_r - t_i)$$

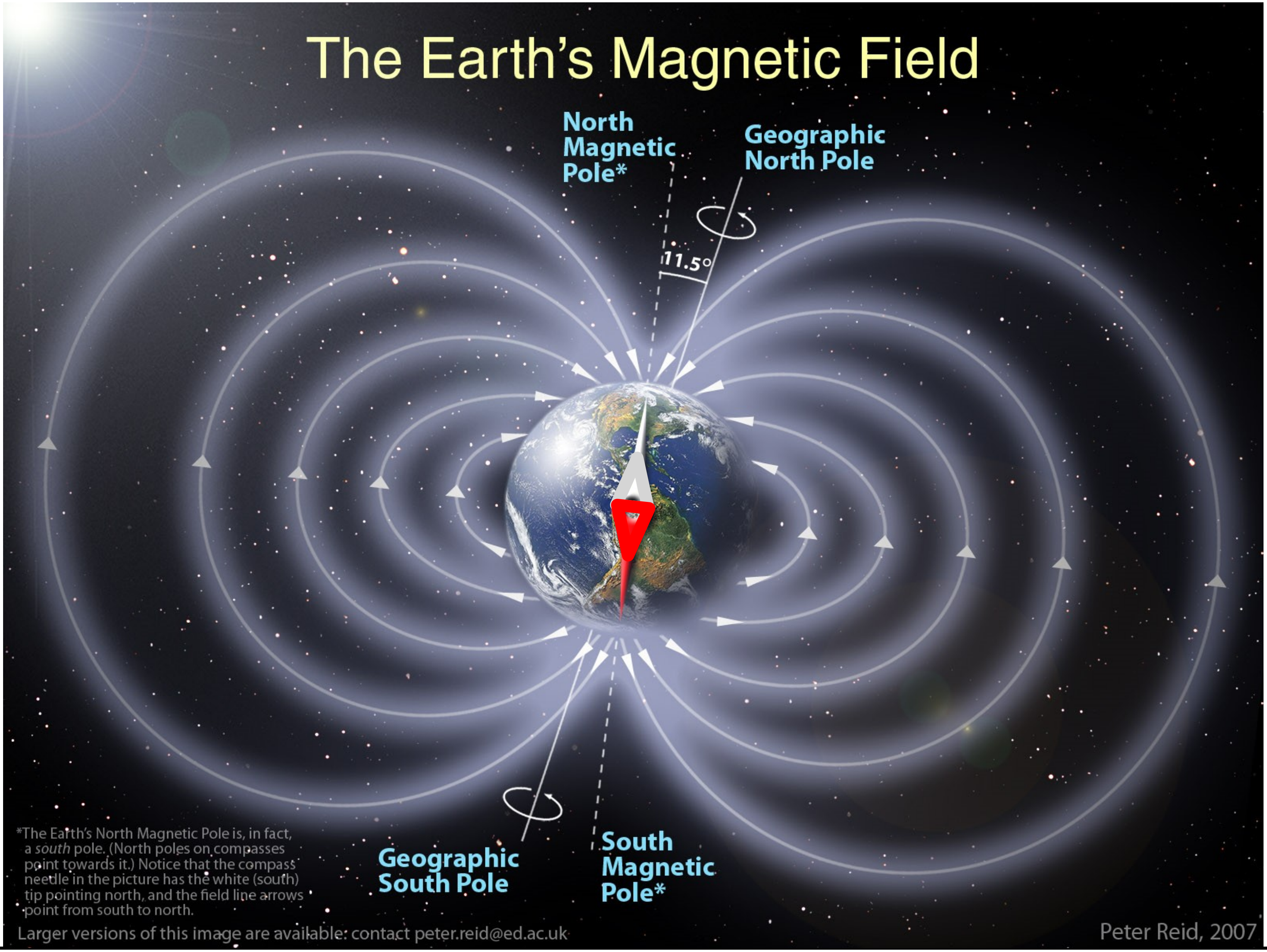
Clock Noise: $t^*_r = t_r + b_{tr} + n_{tr}$, b_{tr}, b_{ti} as random walk, n_{tr}, n_{ti} as white noise
$t^*_i = t_i + b_{ti} + n_{ti}$

Satellite Position Noise: (x_i, y_i, z_i) is a result of the GPS satellite update.
Precise model requires error propagation from this algorithm!

➔ GPS noise model requires noise propagation from all measurements!

Simplified model for GPS position measurement:
$\mathbf{p}^* = \mathbf{p} + \mathbf{b}_\mathbf{p} + \mathbf{n}_\mathbf{p}$, $\mathbf{b}_\mathbf{p}$ as a bounded random walk, $\mathbf{n}_\mathbf{p}$ as zmwgn

Sensor Types: Magnetic Field Sensor



Sensor Types: Magnetic Field Sensor

Earth's magnetic field in practice

International **G**eomagnetic **R**eference **F**ield

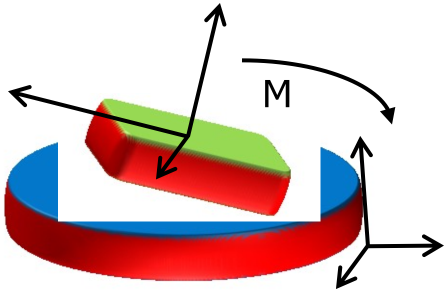
- Highly complex model with several coefficients, download as text files.
- Models the earth’s magnetic field at any location and time between 1900 and 2015.
- IGRF models are standardized for a particular year, reflecting the most accurate measurements available at that time, and indicating a small-scale, slow time variation of the Earth's overall magnetic field.
- Currently at the 11th generation: IGRF-11

Sensor Types: Magnetic Field Sensor

Sensor Model

Define noise model of measurement vector **m*** = (m^*_x, m^*_y, m^*_z)

$$\underbrace{\vec{m}^*}_{\text{measurement}} = \underbrace{\vec{b}_C + s \cdot M}_{\text{calibration}} \cdot \underbrace{\vec{m} + \vec{b}}_{\text{estimation}} + \underbrace{W\vec{n}}_{\text{modelling}} + \vec{o}$$



Where: **m*** = measured magnetic vector

b$_c$ = long term constant bias

s = scale factor

M = misalignement correction matrix

b = short term constant bias

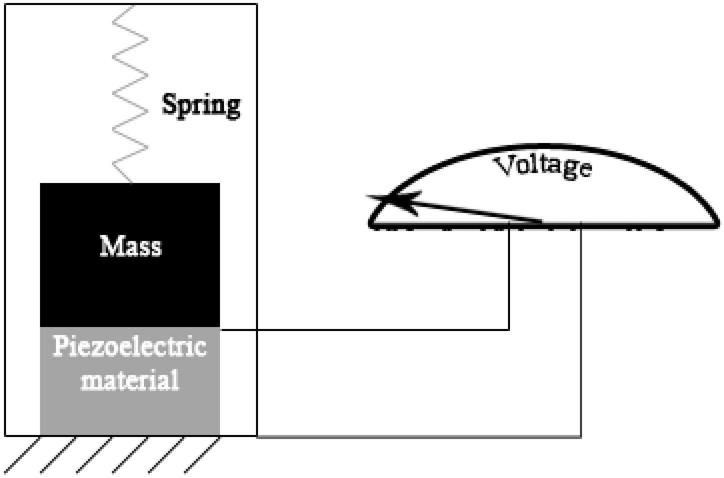
W=crosscorrelation matrix for white noise vector **n**

o = other (environmental) influences

Sensor Types: Accelerometer

„ An accelerometer is a device that measures proper acceleration, the acceleration experienced relative to freefall." [Wikipedia]

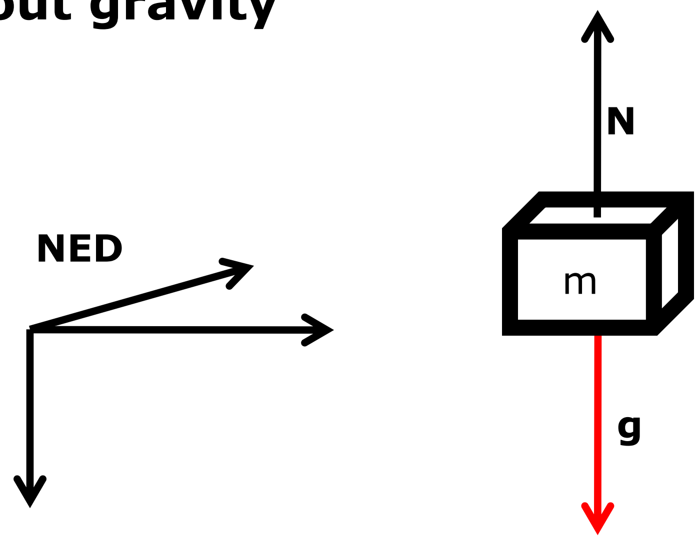
Basic principle:
mass inertia effect



Most widely used types: MEMS devices (Microelectromechanical systems)

Sensor Types: Accelerometer

A word about gravity



Note: $\mathbf{g}$=(0,0,9.81)

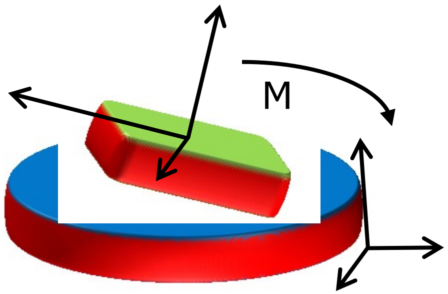
But the measured acceleration <u>in hover mode </u>is
$\mathbf{a}$=(0,0,-9.81)

Sensor Types: Accelerometer

Sensor Model

Define noise model of measurement vector **a*** = (a*$_x$, a*$_y$, a*$_z$)

$$\underbrace{\vec{a}^*}_{\text{measurement}} = \underbrace{\vec{b}_C + s \cdot M}_{\text{calibration}} \cdot \underbrace{\vec{a} + \vec{b}}_{\text{estimation}} + \underbrace{W\vec{n}}_{\text{modelling}} + \vec{o}$$



Where: **a*** = measured acceleration vector

b$_c$ = long term constant bias

s = scale factor

M = misalignement correction matrix

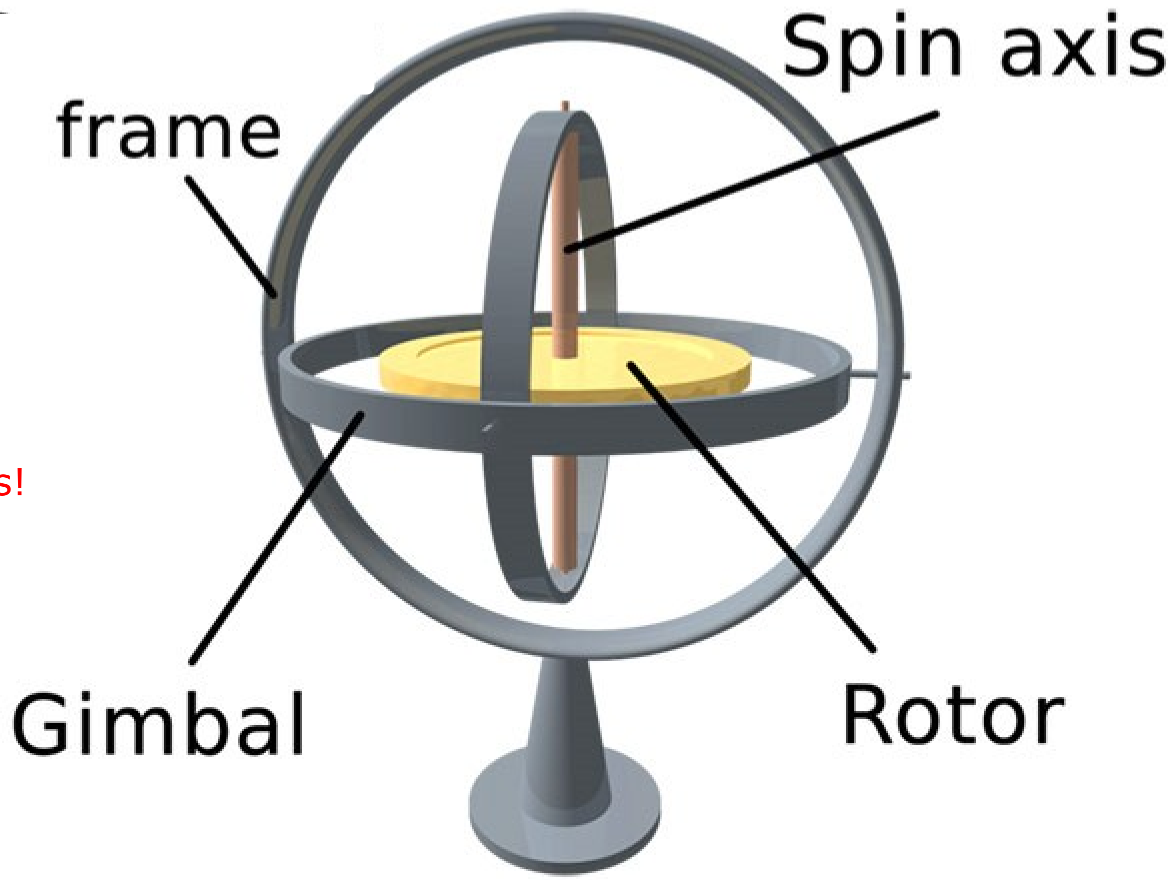
b = short term constant bias

W=crosscorrelation matrix for white noise vector **n**

o = other (environmental) influences

Sensor Types: Gyroscope

„ A spinning wheel is a device for measuring or maintaining orientation, based on the principles of conservation of angular momentum." [Wikipedia]



Mesures
directly
the angles!

Sensor Types: Gyroscope

Gyroscope

Use the Coriolis effect to directly measure angular velocity:

Proof masses vibrate at frequency ω_r giving the position $p = A\sin(\omega_r t)$
The Coriolis effect introduces the acceleration: $a = -2(v \times \omega)$
with the velocity $v = \dot{p} = A\omega_r \cos(\omega_r t)$

We can measure the introduced acceleration with an accelerometer

Note: Angular velocity is directly measurable. Linear velocity is not! (we have to integrate linear acceleration)

Sensor Types

MEMS: Vibrating elements as Micro Electro-Mechanical System

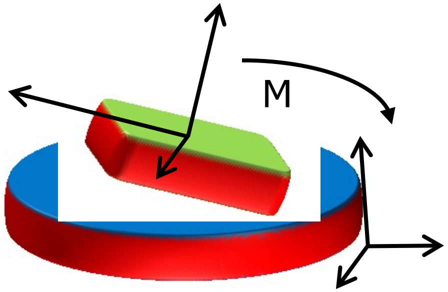
Fiber Optic: Interference of light to detect mechanical rotation (5km fibre!)

Sensor Types: Gyroscope

Sensor Model

Define noise model of measurement vector **ω*** = (ω $^*{}_x$, ω $^*{}_y$, ω $^*{}_z$)

$$\underbrace{\vec{\omega}^*}_{\text{measurement}} = \underbrace{\vec{b}_C + s \cdot M}_{\text{calibration}} \cdot \underbrace{\vec{\omega} + \vec{b}}_{\text{estimation}} + \underbrace{W\vec{n}}_{\text{modelling}} + \vec{o}$$



Where: **ω** * = measured angular velocities

$\mathbf{b_c}$ = long term constant bias

s = scale factor

M = misalignement correction matrix

b = short term constant bias

W=crosscorrelation matrix for white noise vector **n**

o = other (environmental) influences

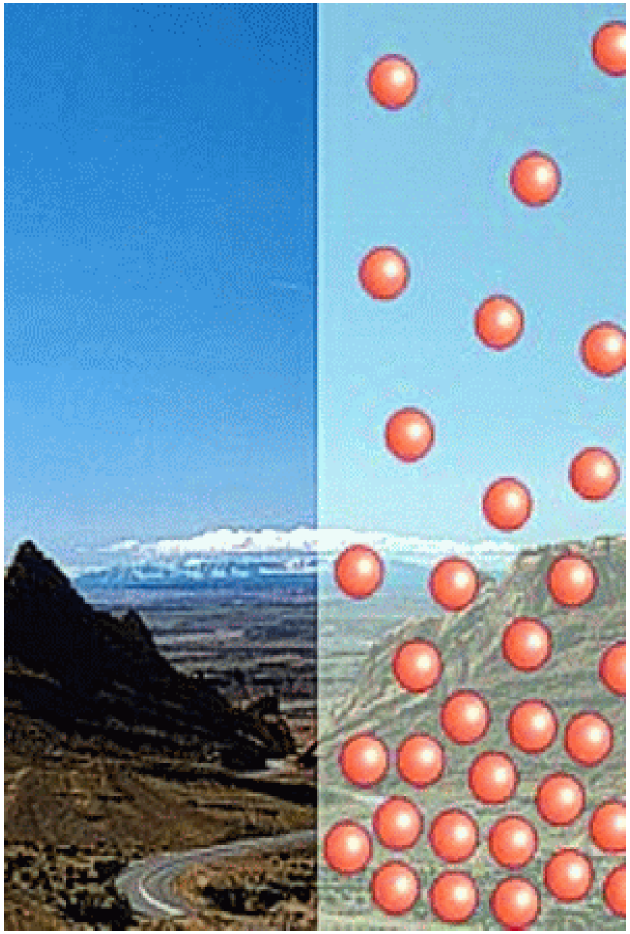
Sensor Types: Air Pressure Sensor

Air pressure as a function of atmospheric height:

$$p = p_0 \cdot e^{-Mgh/RT}$$

$$h = \frac{(\ln(p) - \ln(p_0)) \cdot RT}{Mg}$$

Where: p = pressure at height h
T = temperature at height h in K
R = 8.31432 N·m/(mol·K) universal gas const.
g = gravity
M = 0.0289644 kg/mol Molar mass of Earth's air
p_0 = 1.2250 kg/m^3



This model of the low altitude atmosphere is valid up to an altitude of 11000m.

Sensor Types: Air Pressure Sensor

Error Sources

Outdoor use: highly dependent on weather changes!

Pressure and temperature changes during flight time

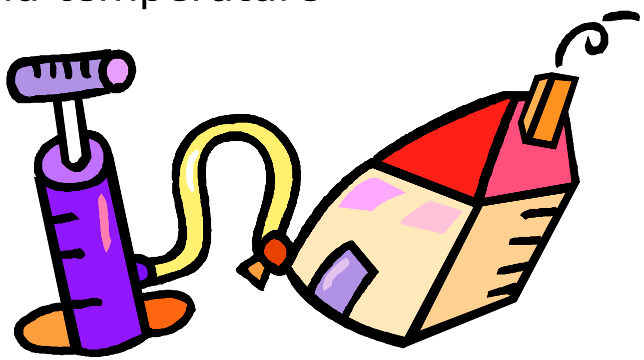
➔ initially measured p_1 and T_1 are not reliable anymore

Indoor use: sudden pressure or/and temperature changes and pressure spikes

Closing doors provoke pressure spikes

Open windows or ventilation changes pressure and temperature

Heating/air conditioning changes temperature



Sensor errors:

sensor model has to account for bias, scaling factor and other noise terms

Sensor Types: Air Pressure Sensor

Sensor Model

Define noise model of measurement p^* and T^*

$$p^* = b_{Cp} + s_p \cdot p + b_p + n_p + o_p$$

$$\underbrace{T^*}_{\text{measurement}} = \underbrace{b_{CT} + s_T}_{\text{calibration}} \cdot \underbrace{T + b_T}_{\text{estimation}} + \underbrace{n_T}_{\text{modelling}} + o_T$$

Where: p^* and T^* = measured pressure and Temperature
b_c = long term constant bias
s = scale factor
b = short term constant bias
n = white noise
o = other (environmental) influences

Ideally, the noise model has to be propagated for Δh!

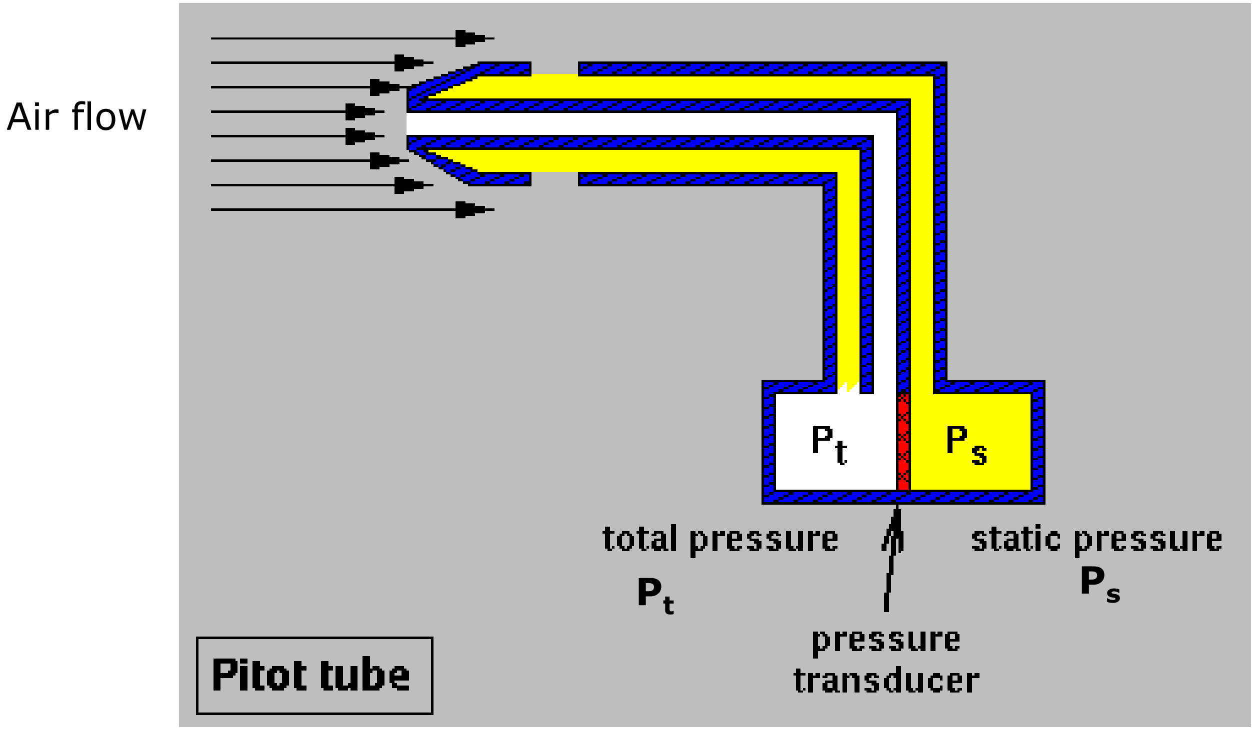
Approximation is
$\Delta h^* = \Delta h + b + n$ for a calibrated sensor with random walk bias b and white noise n

Sensor Types: Air Speed Sensor (Pitot tube)

„ A pitot tube is a pressure measurement instrument used to measure fluid flow velocity” [Wikipedia]

Functioning

Measured air pressure (at stagnation point) is related to the air speed:



Bernoulli`s eqution:

$$p_t = p_s + (0.5 \cdot \rho v^2)$$

With ρ = air density
v = air flow velocity

Attitude Representation

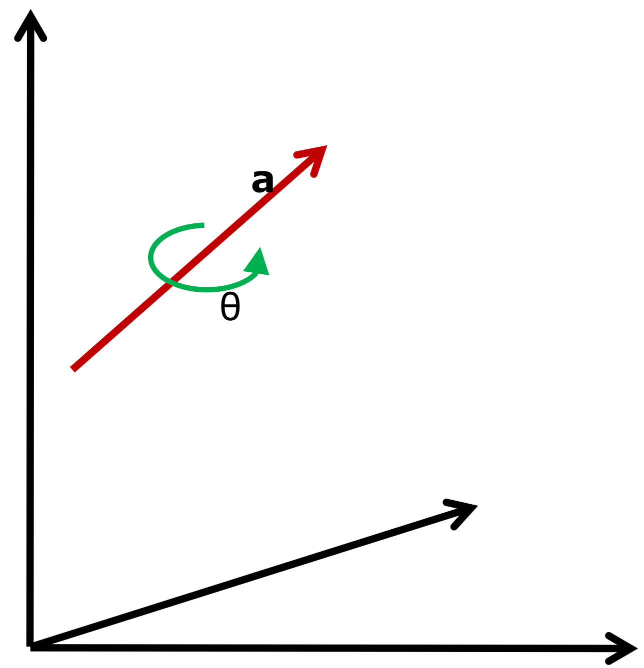
Quaternion as a complex number: $q = q_4 + q_1 i + q_2 j + q_3 k$

with: $i^2=j^2=k^2=-1$

$-ij=ji=k \qquad -jk=kj=i \qquad -ki=ik=j$

Convention as proposed by JPL* (i.e. not Hamilton)

Define: $q = \begin{bmatrix} \vec{r} \\ q_4 \end{bmatrix} = [q_1 \quad q_2 \quad q_3 \quad q_4]^T$



If the following is fulfilled:

$$\vec{r} = \begin{bmatrix} a_x \sin(\theta/2) \\ a_y \sin(\theta/2) \\ a_z \sin(\theta/2) \end{bmatrix} = \vec{a}\sin(\theta/2) \quad ,|\vec{a}| = 1$$

$$q_4 = \cos(\theta/2)$$

Then: q is a “quaternion of rotation”

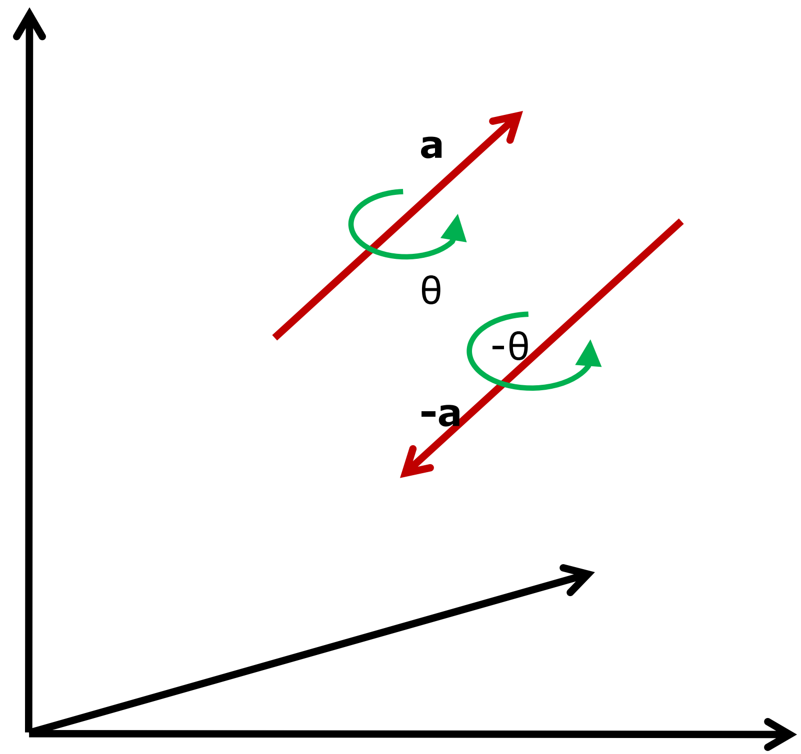
*W.G. Breckenridge, „Quaternions – Proposed Standard Conventions“, JPL Tech. Rep.

Quaternions: Overview and Basics

Quaternion Properties

Recall:

$$q = \begin{bmatrix} a_x \sin(\theta/2) \\ a_y \sin(\theta/2) \\ a_z \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} = \begin{bmatrix} \vec{a}\sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} \quad , |\vec{a}| = 1$$



Hence:

q and –q describe same rotation to final coordinate system position

$$|\vec{q}| = \sqrt{\vec{q}^T \vec{q}} = \sqrt{|\vec{r}|^2 + q_4^2} = 1$$

Other conventions:

$q = q_1 + q_2 i + q_3 j + q_4 k$

Quaternions: Calculation Rules

Multiplication: $q \otimes p = (q_1 i + q_2 j + q_3 k + q_4)(p_1 i + p_2 j + p_3 k + p_4)$

In matrix form: $q \otimes p = M_q \cdot p = M_p q$

also: $a \otimes b \otimes c = M_{ab} \cdot c = M_{ac} \cdot b = M_{bc} \cdot a$ coeff. are linear!

Note: $q \otimes p \neq p \otimes q$

Neutral Element $\mathbf{q_I}$:

Hence: $q \otimes q_I = q_I \otimes q = q$

$$q_I = [0 \quad 0 \quad 0 \quad 1]^T$$

Inverse Element $\mathbf{q^{-1}}$:

Hence: $q \otimes q^{-1} = q^{-1} \otimes q = q_I$ complex conjugate

$$q^{-1} = \begin{bmatrix} -\vec{r} \\ q_4 \end{bmatrix} = \begin{bmatrix} -\vec{a}\sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} = \begin{bmatrix} \vec{a}\sin(-\theta/2) \\ \cos(-\theta/2) \end{bmatrix}$$

Quaternions: Calculation Rules

Vector Rotation

Imaginary quaternion: $v = v_1 i + v_2 j + v_3 k$, $v_4=0$,   Note: $|v| != 1$

Vector relation in different coordinate systems:

$$^{L}v = {}^{L}_{G}q \otimes {}^{G}v \otimes {}^{L}_{G}q^{-1} = {}^{L}_{G}C \cdot {}^{G}v$$

("global vector v expressed in local coordinate system")

Approximations

For very small rotations δq: $q = \delta q \otimes \hat{q}$

$$\delta q = \begin{bmatrix} \delta\vec{r} \\ \delta q_4 \end{bmatrix} = \begin{bmatrix} \vec{a}\sin(\delta\theta/2) \\ \cos(\delta\theta/2) \end{bmatrix} \approx \begin{bmatrix} 0.5\delta\vec{\theta} \\ 1 \end{bmatrix}$$

with $\delta\vec{\theta}$ in the direction of the axis of rotation and the magnitude of the angle of rotation

Quaternions: Conversions

Quaternion to Rotation Matrix

Recall: $^{L}v = {}^{L}_{G}q \otimes {}^{G}v \otimes {}^{L}_{G}q^{-1} = {}^{L}_{G}C \cdot {}^{G}v$

So: $$^{L}v = \begin{bmatrix} (2q_4^2 - 1) \cdot I_{3x3} - 2q_4 \lfloor \vec{r}_X \rfloor + 2\vec{r}\vec{r}^T \\ 0 \end{bmatrix} \cdot \begin{bmatrix} {}^{G}v \\ 0 \end{bmatrix}$$

And thus: $${}^{L}_{G}C({}^{L}_{G}q) = (2q_4^2 - 1) \cdot I_{3x3} - 2q_4 \lfloor \vec{r}_X \rfloor + 2\vec{r}\vec{r}^T$$

Small angle approximation:

From: $$\delta q \approx \begin{bmatrix} 0.5\delta\vec{\theta} \\ 1 \end{bmatrix}$$

We have: $$C(\delta q) \approx I_{3x3} - \lfloor \delta\vec{\theta}_X \rfloor$$

$$\lfloor \vec{v}_X \rfloor = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

skew symetric matrix

Quaternion Derivative

Recall: $\delta q = {}^{L(t+\Delta t)}_{L(t)}q \approx \begin{bmatrix} 0.5\delta\vec{\theta} \\ 1 \end{bmatrix}$

Thus: $\vec{\omega} = \lim_{\Delta t \to 0} \frac{\delta\vec{\theta}}{\Delta t}$

Then: ${}^{L(t)}_{G}\dot{q} = \lim_{\Delta t \to 0} \frac{1}{\Delta t}\left({}^{L(t+\Delta t)}_{G}q - {}^{L(t)}_{G}q\right) = \frac{1}{2}\Omega(\vec{\omega}) \cdot {}^{L(t)}_{G}q$

With: $$\Omega(\vec{\omega}) = \begin{pmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{pmatrix}$$

And $\vec{\omega} = {}^{L(t)}\vec{\omega}$ in the local body frame L(t)

Quaternions: Derivatives & Integration

Quaternion Integration

Find a solution for the following first order differential equation:

$$ {}^{L(t)}_{G}\dot{q} = \frac{1}{2}\Omega(\vec{\omega})\cdot{}^{L(t)}_{G}q $$

Zeroth order quaternion integrator (for small angular velocities):

$$ {}^{L(t+\Delta t)}_{G}q = \left(I_{4x4} + \frac{\Delta t}{2}\Omega(\vec{\omega}) \right)\cdot{}^{L(t)}_{G}q $$

First order quaternion integratior is slightly more complex*

*Trawny N., Roumeliotis S.I., „Indirect Kalman Filter for 3D Attitude Estimation“, Tech. Rep.

Contents

1. Pose Estimation and Sensors
   a) Overview
   b) Sensor Types
   c) Quaternions
2. Sensor Fusion
   a) Merging Strategies
   b) Extended Kalman Filter
   c) Example
3. Camera as a Sensor
   a) Collaborative stereo

a) Merging Strategies
   – Correction, Colligation, Fusion

b) Extended Kalman Filter
   – Review Kalman Filter
   – Extended Kalman Filter
   – Defining the Filter and Noise
   – Propagation
   – Update

c) Example: Attitude
   – Filter Definition
   – Noise Calculations
   – System Equations: EKF with Quaternions
   – Propagation
   – Update

Sensor coupling classification 1
How is the second sensor used?

Correction

Uses information from one sensor to correct or verify another.

> **Ex:** Integrate the IMU accelerations for a position estimate and compensate for the drift with a DGPS measurement.
> (Note: normal GPS is not accurate)
> **Ex:** Estimate the rotation using vision, verify accuracy with gyros

Colligation

Merges different parts of the sensors

> **Ex:** Use gyro information to eliminate rotational effects on optical flow translation estimation

Fusion

Merges sensor information in a (statistically) optimal way

Sensor coupling classification 2
Loose/Tight coupling

Loose coupling

Using separate INS and SFM blocks running at different rates and exchanging information

Tight coupling

Combining the disparate raw data of vision and inertial sensors in a single block

Sensor coupling classification 3
Statistical/Deterministical fusion

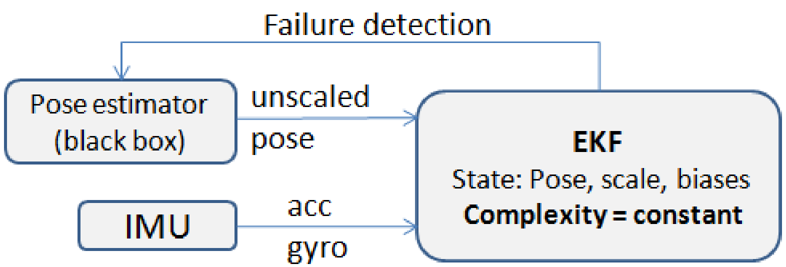
Statistical fusion

Fusion using a filter based strategy
(Kalman filter, EKF, UKF, Particle filter)

Deterministic fusion

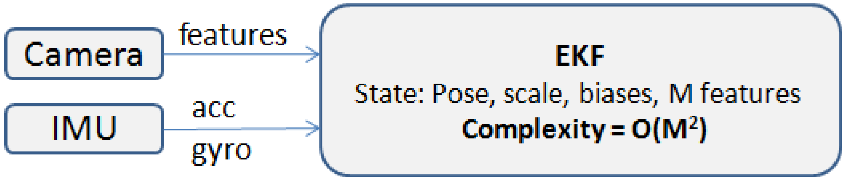
Fusion using a closed-form solution

Four variations



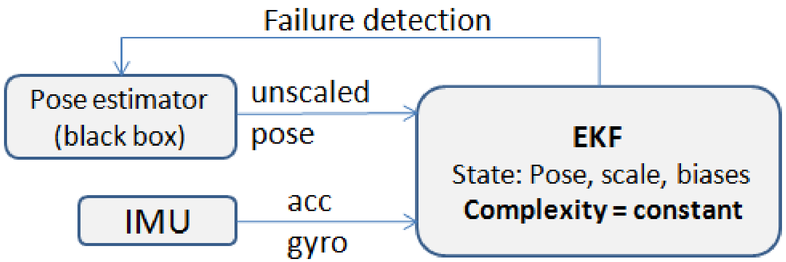
- Loosely coupled
  - EKF-based pose estimation filter with visual pose and IMU measurements (scale as additional state variable) [Weiss, ICRA 2011]
  - Closed-form solution for scale determination using short-term integration of inertial readings (delta observations) and consecutive SLAM poses in relative scale [Kneip, IROS 2011]

Statistical fusion



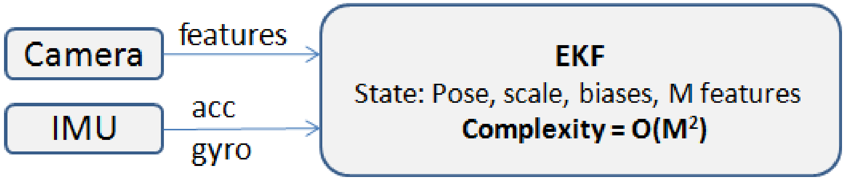
- Tightly coupled
  - EKF-based SLAM with centripetal acceleration motion model (using intertial readings) [Strelow 2004, Jones 2009, Kelly 2009]
  - Closed-form solution for direct speed determination using inertial delta observations and single feature correspondences [Kneip et al., ICRA 2011]

Four variations



- Loosely coupled
  - EKF-based pose estimation filter with visual pose and IMU measurements (scale as additional state variable) [Weiss, ICRA 2011]
  - Closed-form solution for scale determination using short-term integration of inertial readings (delta observations) and consecutive SLAM poses in relative scale [Kneip, IROS 2011]

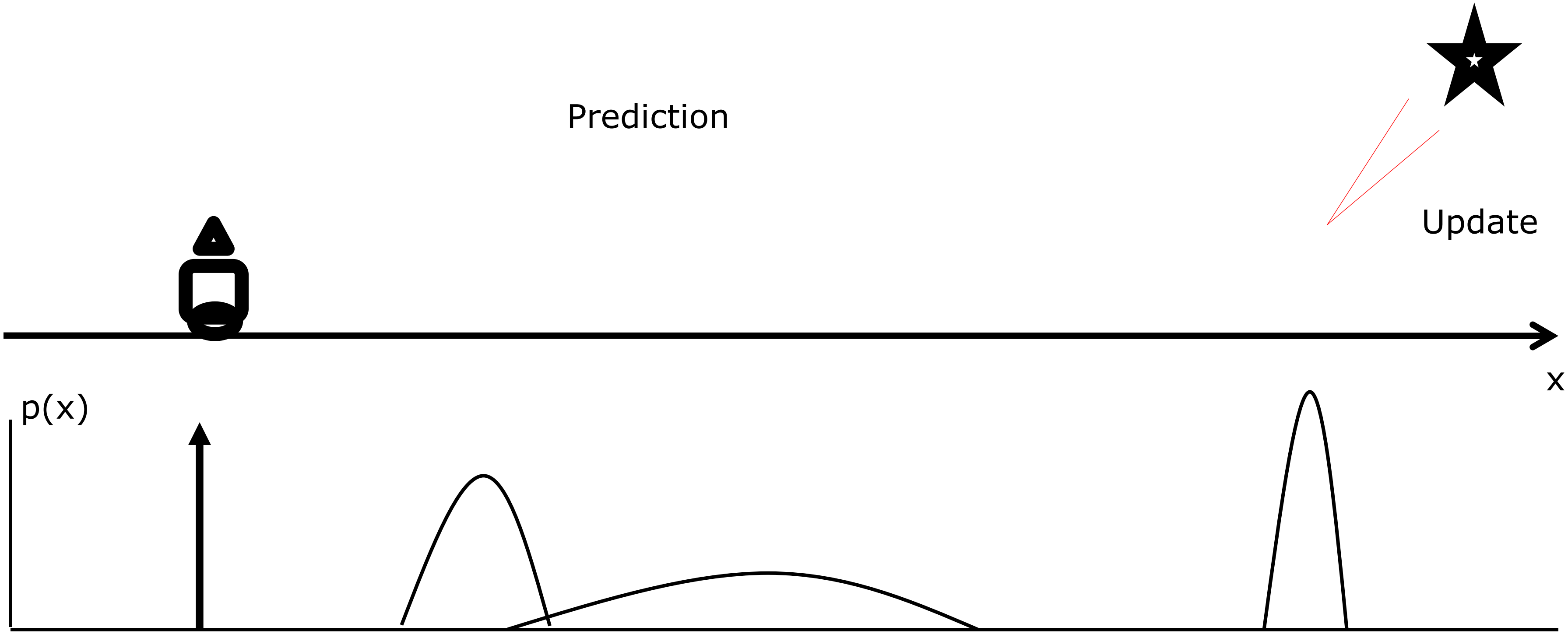
Deterministic fusion



- Tightly coupled
  - EKF-based SLAM with centripetal acceleration motion model (using intertial readings) [Strelow 2004, Jones 2009, Kelly 2009]
  - Closed-form solution for direct speed determination using inertial delta observations and single feature correspondences [Kneip et al., ICRA 2011]

Extended Kalman Filter*

Review Linear Kalman Filter



* *Peter Maybeck "Stochastic Models, Estimation, and Control" Vol 1-3*

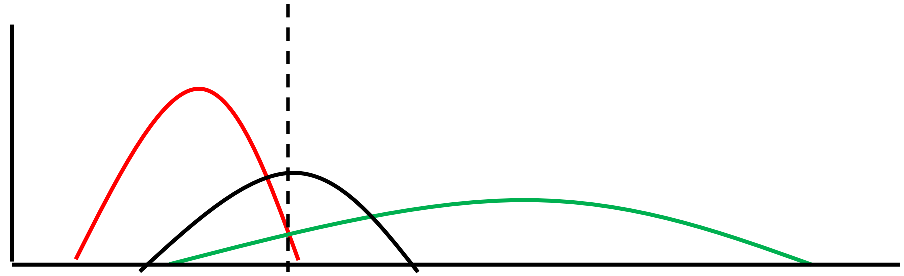
Review Linear Kalman Filter

Prediction step:

$$p(x_{t+1}|u_t) = \int_{x_t} p(x_{t+1}|u_t, x_t)p(x_t)dx_t = p(x_{t+1}|u_t, x_t) \otimes p(x_t)$$

Update step: bayes rule

$$p(x_{t+1}|z_{t+1}) = \frac{p(z_{t+1}|x_{t+1}) \cdot p(x_{t+1})}{p(z_{t+1})} = \eta\, p(z_{t+1}|x_{t+1})\, p(x_{t+1})$$



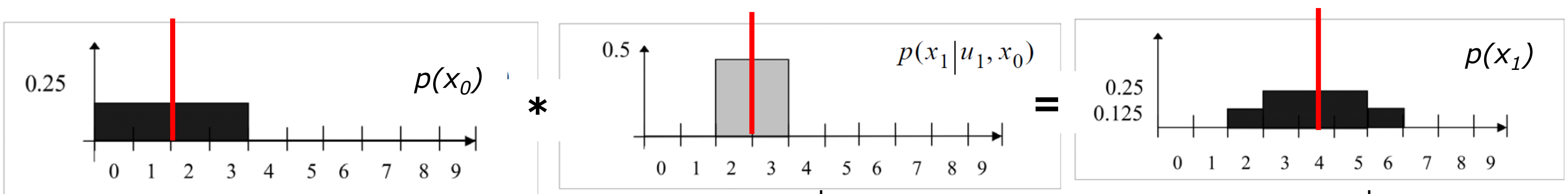
Review Linear Kalman Filter

Linear Filter System:
$$\vec{x}_{k+1} = A\vec{x}_k + B\vec{u}_k + W\vec{n}_x$$
$$\vec{z} = H\vec{x} + V\vec{n}_z$$

Prediction step:

propagating the mean and variance of the gaussian distribution p(x)

mean: $\vec{x}_{k+1} = A\vec{x}_k + B\vec{u}_k$



$$p(x_t) \otimes p(x_{t+1} | u_t, x_t) = p(x_{t+1} | u_t)$$

covariance matrix P: use the error propagation law

$$P_{k+1} = AP_kA^T$$

Review Linear Kalman Filter

Linear Filter System: $\vec{x}_{k+1} = A\vec{x}_k + B\vec{u}_k + W\vec{n}_x$

$$\vec{z} = H\vec{x} + V\vec{n}_z$$

Update step: Minimum mean-square error estimator $\min\left(E\left(\left|\tilde{z}_k - z_k\right|^2\right)\right)$

Error: $y = \tilde{z} - Hx$

Error covariance (innovation): $S = R + HPH^T$ $\quad R = V\operatorname{cov}(n_z)V^T$

Optimal kalman gain: $K = P_k H^T S^{-1}$

State: $x_{k+1} = x_k + K\vec{y}$

Covariance: $P_{k+1} = \operatorname{cov}(\tilde{x}_k - x_k) = (I - KH)P_k(I - KH)^T + KRK^T$

Peter Maybeck "Stochastic Models, Estimation, and Control" Vol 1-3

Extended Kalman Filter

Non-linear Filter System:

$$\dot{x} = f(x,u) + Wn_x$$

$$z = h(x) + Vn_z$$

Use jacobians for prediction and update step:

$$A = J\big(f(x,u)\big)$$

$$H = J\big(h(x)\big)$$

Note:

This is a linear approximation step around the current state. It may yield poor results in highly nonlinear systems. More robust filters are in this case particle or unscented kalman filters. (at the cost of calculation power)