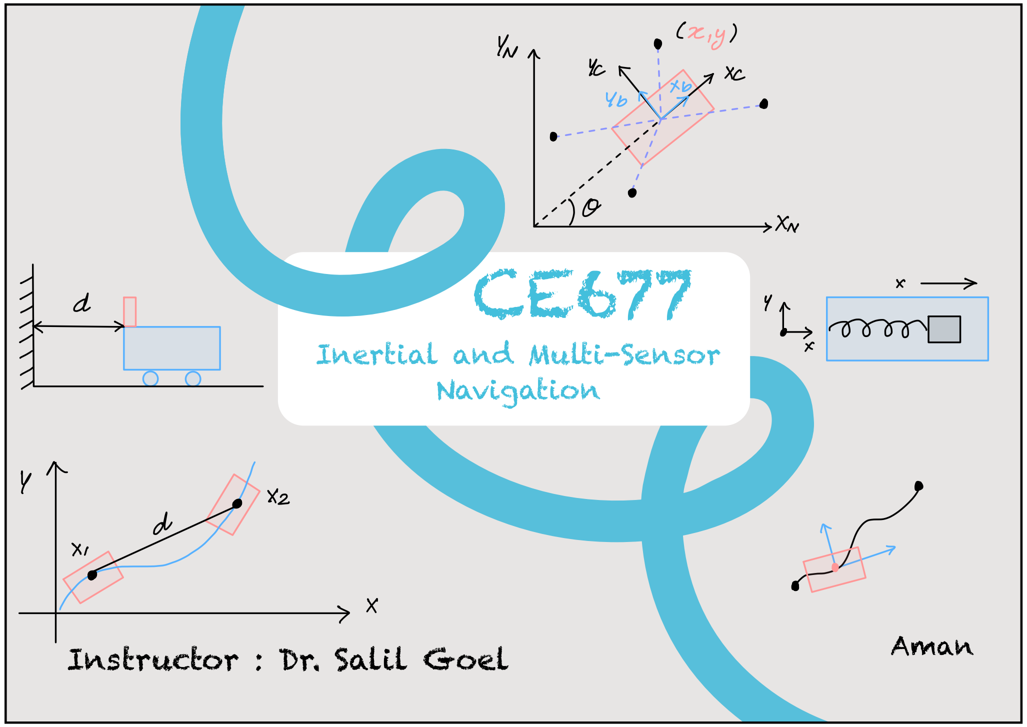


CE677

Inertial and Multi-Sensor Navigation

Instructor : Dr. Salil Goel

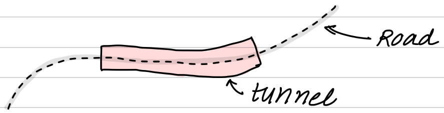
Aman

INERTIAL AND MULTI-SENSOR NAVIGATION

End - sem	40 %	
Labs (Attendance)	25 %	Lab 2PM to 5PM
Quiz (3)	15 %	@ Varun Lab
Home Assignment	20 %	

I — Sensors (2 weeks)
II — Mathematics (3 weeks) ~ use for positioning + navigation
III — Applications (2 weeks)

Sensors which we use when GPS doesn't work.
eg:- travelling in a road, want to track path.
suppose if inside a tunnel ~ GPS won't work
so how to track the path / location in that part.



SENSORS PART

Inertial sensors

- eg: auto-rotate in your smartphone, inertial (sense rotation)
- cost can vary from hundred to crores.

- Inertial sensors have

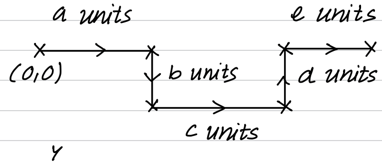
· Gyroscope (3axis) · Accelerometer (3axis) · Magnetometer (3axis)	· Barometer · Processor

} → Depending upon cost

- To define an object in space we need position + orientation
GPS gives only position, doesn't tell you orientation.
We use inertial sensors along with it to get orientation.

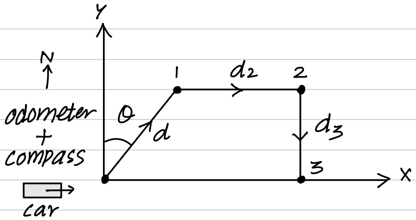
- Let us try to understand how would we do navigation if GPS was not there. GPS became operational in 1990s, What people did to do navigation before GPS. They used to rely on a principle called Dead Reckoning.

Dead Reckoning



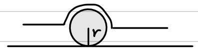
1. find change in position
2. add to prev. position to get current position
3. 2D ~ heading only
3D ~ three orientation (attitude)

can find new location if I know the initial location & can also trace the path back.



1 $(d\sin\theta, d\cos\theta)$
2 $(d\sin\theta + d_2, d\cos\theta)$
3 $(d\sin\theta + d_2, d\cos\theta - d_3)$

- angles ~ measured by compass
- distances ~ measured by odometer, it measures distances by counting the rotations of a wheel



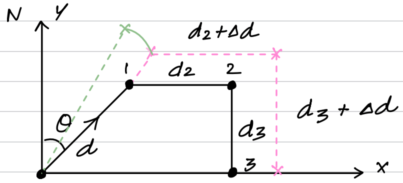
1 rotation distance travelled = $2\pi r$ (assuming no slippage)

- Everything is happening in a Local coordinate system.
- In 3D world (undulating terrain), just heading (yaw) information won't work. I need roll + pitch also.
- All these are good if the sensors are working well. But distance have errors.

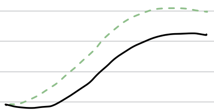
Δd — constant or distance dependent or random error

$d = d + \underbrace{\Delta d}_{\text{error}}$ (distance)

As you go further, errors add on ~ errors start to grow
~ sensors become useless
~ trajectory also get changed



odometer + compass



- Dead Reckoning ~ Relative system for positioning
- GPS ~ Absolute system for positioning
- Disadvantage of Dead Reckoning ~ accumulation of error

1. Dead Reckoning can be used for positioning
2. Errors will grow over time. ~ Need something to stop this accumulation of error. ~ GNSS/INS Integration

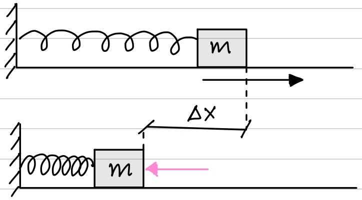
GNSS/INS Integration

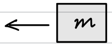
INS — rely on dead reckoning ~ relative positioning }
GNSS — absolute positioning }

This integration is done to avoid errors to grow. GNSS used to correct the errors and keep on using it even if GPS is not available for sometime. There is research being conducted on different ways to stop these errors in INS from growing.

Accelerometer

1. Accelerometer is a spring-mass system





$-kx = ma$

$a = -\frac{kx}{m}$

using accⁿ ~ find distances
~ do dead reck.

2. Accelerometer — accelerations Δt (epoch)

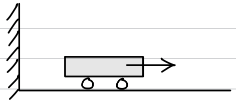
$a \longrightarrow a + \Delta a$ (error in acceleration)

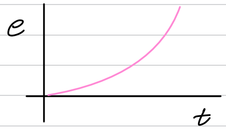
$d = \iint a\,dt$

$v \to a\Delta t \rightsquigarrow a\Delta t + \Delta a \Delta t$

$d \to \frac{1}{2} a \Delta t^2 \rightsquigarrow$

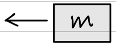
} If a has error, velocity and distance will also have error.





error grow as time ↑
(with time it propogates)

3. Whenever you buy accelerometer from a vendor, he will do some kind of calibration and give you $\Delta a = ?$
   - For cheaper $\Delta a =$ large eg. phones
   - For expensive $\Delta a =$ small eg. fighter jets

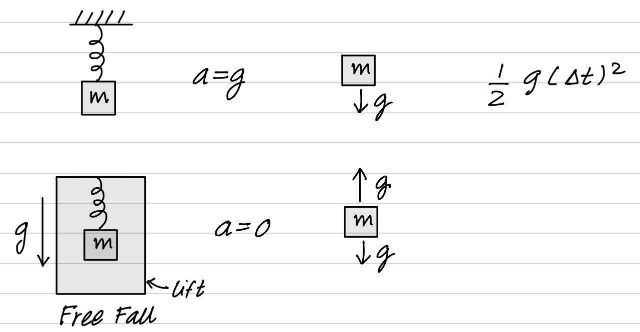


$-kx = ma$

$a = -\frac{kx}{m}$

usually dampers to ensure no damping (ignored in this eqⁿ)

k,m ~ known } → find a → find d (distance)
find x }



4. Accelerometer tells you total acceleration acting on the body that includes acceleration due to gravity. To get actual acc^n we need to subtract gravity.

$$\boxed{a_z = a_s - g}$$

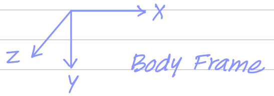
actual acc^n

sensor acc^n

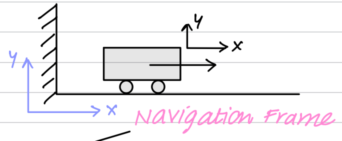
acc^n due to gravity

this value must be known to you beforehand
— NASA — Marsian Rover uses accelerometer need to know 'g' first

5. On every sensor, there is a coordinate system marked called Body Frame (x, y, z)



Accelerometer is aligned in there directions and will measure everything w.r.t. body frame (it doesn't know ECEF and ECI)



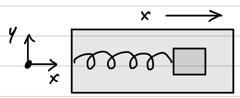
It can be (NED, ENV) ECEF in which I want the position but I am measuring everything in body frame.

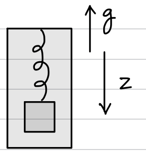
transformation:
Body Frame → Navigation Frame

Accelerometer

$a = -\frac{R}{m} x$

"Body Frame"



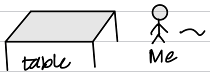


a=g

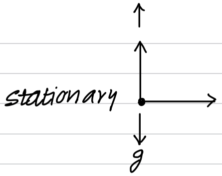
"Inertial Frame"
It measures everything w.r.t. an Inertial Frame
↓
doesn't rotate & doesn't accelerate

ECI (Earth centered Inertial) ~ an inertial frame ~ do not rotate w.r.t. earth
ECEF ~ non-inertial frame

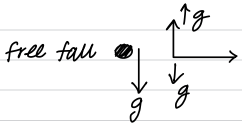
f_{ib}^{b} = acceleration of body frame w.r.t. an inertial frame measured in body frame
→ vector ~ denote the acc^n from accelerometer as f_{ib}^{b}



~ non-inertial frame (rotate with earth)



To make it an inertial frame, apply 'g' in opposite dir^n.
For stationary object, acc^n 'g' in upward direction.
sensor measure everything w.r.t. inertial frame.
↳ some frame not accelerating

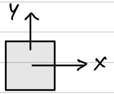


a=0 (accelerometer reading)

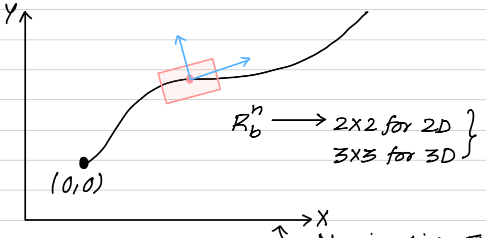
"Inertial" word comes from "Inertia" ~ tendency to resist its change in motion. ~ by Newton the guy who never leaves you.
Inertial sensors measure this inertia, that's why called so.

$f_{ib}^{b} = [x\ y\ z]$

~ acc^n of body frame w.r.t. inertial frame expressed in body frame



[a, 0, 0] [0, a, 0] ~ sensors resolve them in body frame



Navigation Frame (eg. ECEF)

Navigation Frame

Body Frame

Vehicle Frame

$v = v_0 + R_b^n \int a\,dt$
$x = x_0 + \int v\,dt$ } x and v in navigation frame

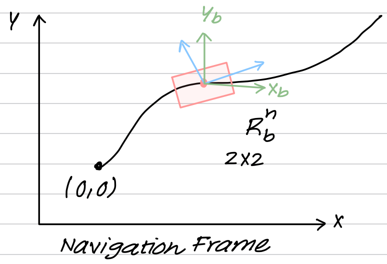
acceleration in body frame

~ have to convert in navigation frame ~ Rotation matrix

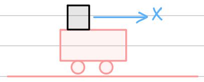
R_b^n = Rotation matrix to convert from body frame to navigation frame

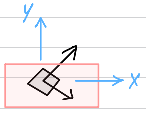
{ $a^b \to v^b$
$x = x_0 + R_b^n v^b \Delta t$ } translation + rotation ~ w.r.t an ECEF frame

strictly speaking ECI is itself not an inertial frame (rotate w.r.t sun) but for our application we assume it inertial bcoz acc^n negligible



$v = v_0 + R_b^n \int a\,dt$
$x = x_0 + \int v\,dt$



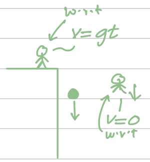


vehicle frame

To keep things simple, we make body frame and vehicle frame parallel. But this may not always be the case.

$R_b^n = R_v^n R_b^v$

$R_b^v = I$ (Identity matrix) if b ∥ v (parallel)



· I need to know — heading in a 2D case
· I need to know — roll, pitch, yaw in a 3D case (eg aircraft)
(heading)

2D case

$R_2 = \begin{bmatrix} \cos\theta\ \sin\alpha & -\ - \\ -\ \ - & -\ - \end{bmatrix}$

heading ~ find using compass
or
using gyroscope

3D case

$R_3 = \begin{bmatrix} - & - & - \\ - & - & - \\ - & - & - \end{bmatrix}_{3\times3}$

$\begin{array}{cl} \begin{bmatrix} \theta \\ \phi \\ \omega \end{bmatrix} & \begin{array}{l} \text{roll} \\ \text{pitch} \\ \text{yaw (heading)} \end{array} \end{array}$

In a 2D case I can't just use accelerometer to find my position using dead reckoning ~ need heading also.

Accelerometer

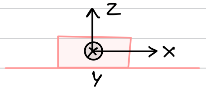
1. The resultant position of the mass w.r.t. case ∝ acceleration applied to the case

2. The exception — acceleration due to gravitational force. Gravitation acts on proof mass directly, not via the springs and applies the same acceleration to all components of the accelerometer, so there is no relative motion of mass w.r.t. case.

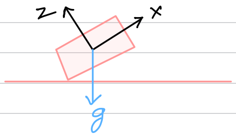
3. All accelerometer sense specific force, the non-gravitational acceleration, not the total acceleration.

specific force = non-gravitational force per unit mass on a body sensed w.r.t. an inertial frame.

Important Question

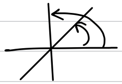


[0,0,g]

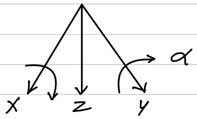


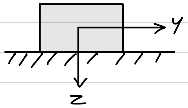
Can I find its orientation just by using accelerometer values?

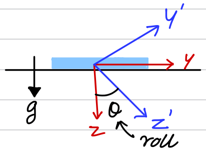
cheaper phones ~ accelerometer only
gyroscope absent
expensive phones ~ gyro + accelerometers both



Can I find orientation (roll, pitch, yaw) just using accelerometer?

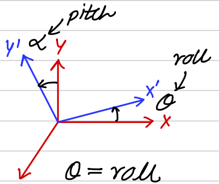






$f_x = 0$
$f_y = -g\sin\theta$
$f_z = g\cos\theta$

$\tan\theta = -\frac{f_y}{f_z}$ → roll (rotation about x-axis)



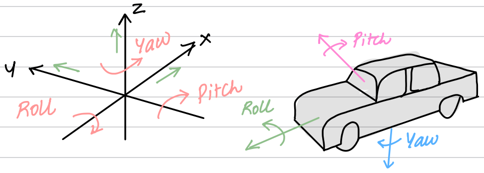
θ = roll
α = pitch

$f_x = g\sin\alpha$
$f_y = g\sin\theta\cos\alpha$
$f_z = g\cos\theta\cos\alpha$

$\tan\alpha = \frac{f_x}{\sqrt{f_y^2 + f_z^2}}$ → pitch (rotation about y-axis)

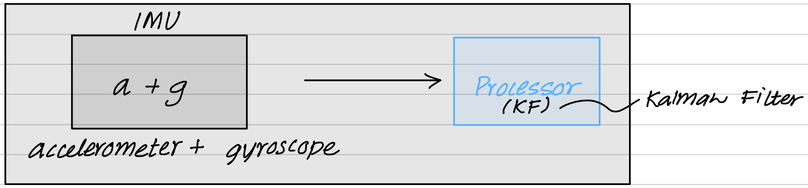
From accelerometer readings (fx, fy, fz) only.
We can find roll + pitch
But we can't find heading (yaw) just from accelerometer.
We need gyroscope for that.

Roll ~ rotation about x-axis
Pitch ~ rotation about y-axis
Yaw ~ rotation about z-axis



gyroscope ← { roll, pitch, yaw (heading) }
roll, pitch } → accelerometer
yaw (heading) } → compass

INS / AHRS



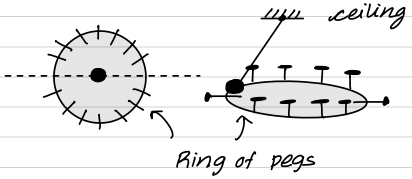
① IMU Inertial Measurement Unit
③ INS Inertial Navigation System ~ pos^n, vel., orientation
② AHRS Attitude Heading Reference System ~ attitude + heading

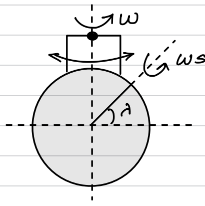
- In terms of cost IMU < AHRS < INS
- IMU and INS are not the same, their cost will vary a lot.

Gyroscope

Foucault Pendulum

1. Experiment by a French physicist Leon Foucault in 1851.
2. A long and heavy pendulum suspended from a high point
3. Used to measure rotation of earth.
4. Later the first gyroscope designed based on this principle.





$\omega \sin\lambda = 11{\cdot}25^\circ$ (Paris)

λ = longitude

24 hours ⟶ 360°

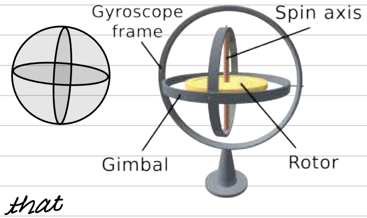
$\Rightarrow$ 15° per hour (at poles)

For foucault (in Paris, France)

$\approx 11{\cdot}25^\circ$ per hour

For equator, $\underset{0}{\underset{\downarrow}{\omega \sin\lambda}} = 0$

Spinning Mass Gyros



- conservation of angular momentum
- Maintains its axis as long as mass is spinning
- Gimbals attached to spin mass to ensure that rotation of outer body is seperate from rotation of this spin mass.
- spin mass gyros ~ not used these days ~ bulky and expensive
  But it was the first gyroscope.

Gyroscope

1. Gyroscope measures angular rate (rate of change of orientation)
2. Given initial coordinates, by successive integration of angular rates, we can compute the platform orientation.

MEMS Based Accelerometers ~ No spring mass system ~ cheap
MEMS Based Gyroscopes ~ No spin mass system ~ bit expensive
accelerometer ~ cheap ~ found in all phones
gyros ~ expensive ~ not found in cheap phones

MEMS Based Gyroscope

$(\omega_x, \omega_y, \omega_z)$

1. No spinning mass system
2. Very very small, cheap and easy to manufacture
3. But they are not very stable ~ errors are varying and have to be corrected with some mathematical model.
4. Original gyros directly give you angular rate. But these modern gyros are vague.
5. They measure rate of change of orientation (angular rate) of body frame w.r.t. an inertial frame in θ per second or rad/sec

$$\theta = \theta_0 + \int \omega dt$$

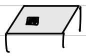
⟶ have errors, on integrating errors propogate

$$v = v_0 + R_b^n \int a dt$$

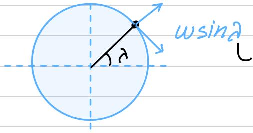
$R_b^n \, W_{ib}^n$ ~ angular rate of body frame w.r.t. inertial frame

---

MEMS based gyroscope ~ put on table ~ output?



It will give you rotation of the earth because body frame is now fixed to the earth, it is measuring rotation of earth w.r.t. inertial frame and w.r.t. inertial frame earth is undergoing rotation.



↙ gyro measure rotation of earth

└ lattitude

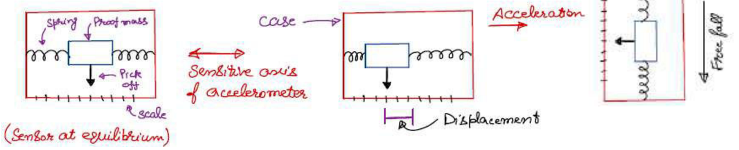
If I was able to take foucault principle gyroscope and put it on a car, the car is also undergoing rotation. That gyroscope measure combined rotation of earth and car (body)

Assume noise is low and air drag is low.
If noise is more than earth's rotation you wont be able to observe 'w' because noise is very high. This is what happens with MEMS based sensors, they have lot of noise and they are not able to measure earth's rotation.

MEMS-based gyroscope ~ consumer grade gyroscope ~ we use
Resonant fiber-optic gyroscope (RFOG)
Ring Laser gyroscope (RLG)
Micro-optic gyroscope (MOG)

Working of inertial sensors

Accelerometer

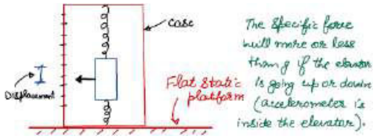


Accelerometers can measure specific forces (non-gravitational force per unit mass) only!

If the platform accelerates along the sensitive axis of the accelerometer, its acceleration is recorded by the sensor.

The displacement of the proof mass is proportional to the acceleration of the platform.

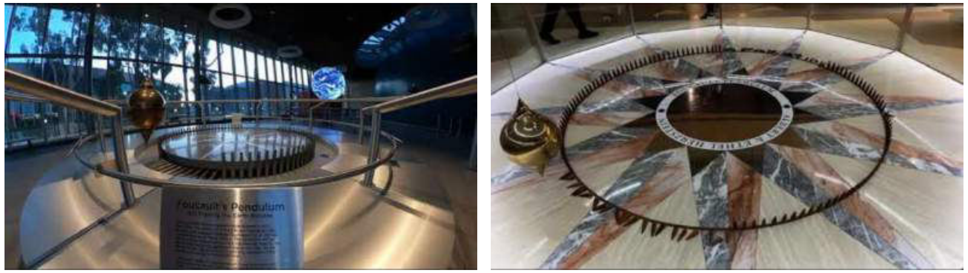
This acceleration is measured w.r.t an inertial frame (non-accelerating frame), in the frame attached to the case.



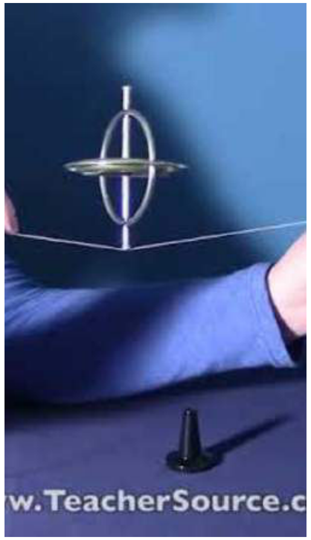
The specific force will more or less than g if the elevator is going up or down (accelerometer is inside the elevator).

Knowledge of acceleration due to gravity is important when working with inertial sensors.

Foucault Pendulum

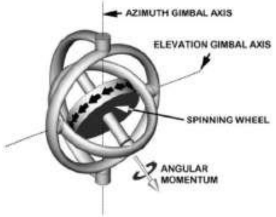


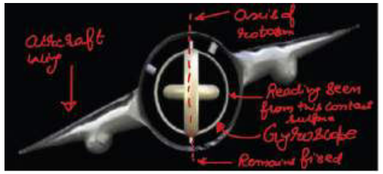
Spinning top as a gyroscope

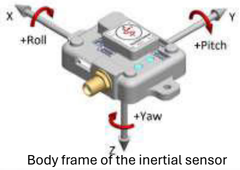


Working of inertial sensors

Gyroscope



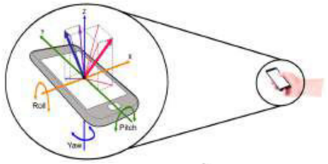


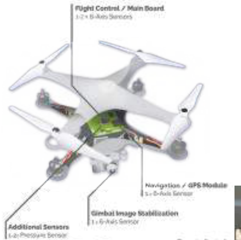


Body frame of the inertial sensor

Gyroscope measures angular rate (rate of change of orientation).
Given initial orientation, by successive integration of angular rates, we can compute the platform orientation.

Inertial sensors



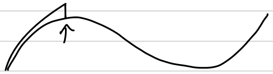


Robotics applications

- Sports
- Clinical diagnostic
- Biomechanical assessment
- Indoor navigation
- Motion capture applications (e.g. sports, movies).

We are surrounded with inertial sensors!

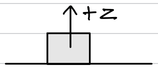
Lab Doubts (Inertial Sensors in Everyday Life)



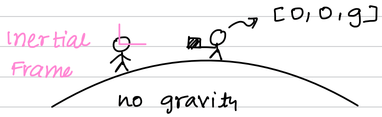
$\frac{800000}{3600} = 222.22$ m/s

[X y z] axis of the smartphone ~ Right hand cs

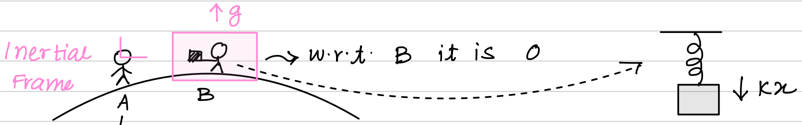
[0, 0, g] ~ g upward ~ why upward?
g is acting downward



Body frame w.r.t. inertial frame



no gravity
(hypothetical planet)

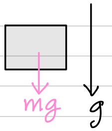


→ w.r.t. B it is 0

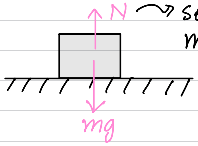
w.r.t. A it is g

This condition is analogous to that.

Accelerometer will only measure the non-gravitational force



Free fall accelerometer reading = 0

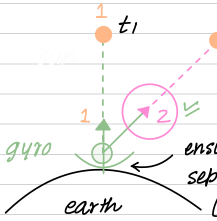


N → sensor will measure this → accelerometer reading = N = +g ↑upward

accelerometer will only measure non-gravitational force (sensor)

{accelerometer ~ spring mass system
gyroscope ~ spinning top system}

What will be axis of rotation of gyroscope after few hours when sun would appear at new position due to rotation of earth? 1 or 2

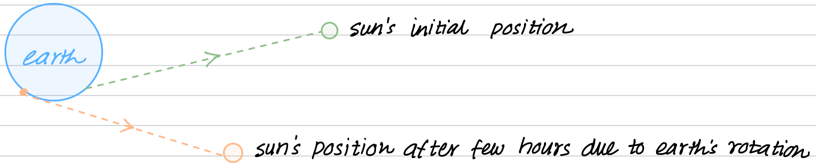


Ans : 2 ✓ It will always point towards sun. Doesn't bother about earth's rotation.

ensure its movement is seperate from earth (using gimbals)

you are standing on earth, gyroscope axis of rotation is along 1 i.e. initial position of sun w.r.t. earth. I ensure rotation of spinning mass is seperate from earth by using Gimbals.

After few hours, earth will undergo rotation so sun will appear at position 2.
Now the axis along which gyroscope sensor will undergo rotation will point towards 1 or 2 ?
It should point towards 2 (pink)



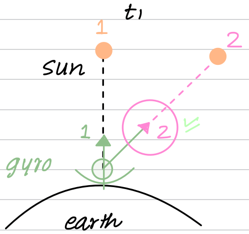


diagram w.r.t. person on ground

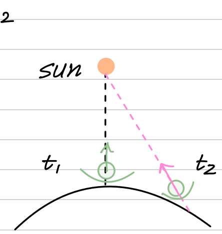
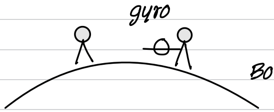


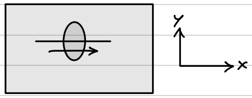
diagram w.r.t. person in space/sun

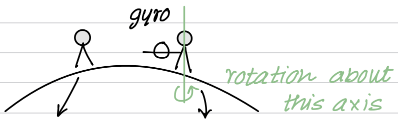
Axis always point towards sun. It does not bother about rotation of earth. So by measuring its change I can measure rotation of earth assuming my sensor have low noise (noise is not so significant)

you seem to be lost ~ let's take an analogy



Both are initial frame

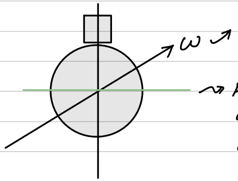




inertial frame
↓
axis will appear to rotate

non-inertial frame
↓
axis will always point towards him

Same idea with sun and earth problem
— to a person in space, you'll undergo rotation
— to a person on ground, you'll not undergo any rotation. That person will say your axis is always pointing towards the sun in the same direction.



↗ whatever you are observing is component of ω and not ω anymore

↝ pendulum says you are not undergoing any rotation at all (bcoz no component along it ↑) at equator

ERRORS IN INERTIAL SENSORS

$a = a_T + s_1 + \epsilon$

$w = w_T + s_2 + \epsilon$

Systematic Errors ⟶ $y = f(x)$

Random Errors ⟶ stochastic

Systematic errors ~ I can write mathematical model $y = f(x)$

Random errors ~ I can write some kind of stochastic model

Bias ~ systematic error

$$a = \underset{\text{true value}}{\underset{\downarrow}{a_T}} + \underbrace{\underset{\text{bias}}{\underset{\downarrow}{b}} + \underset{\text{scale factor}}{\underset{\downarrow}{ka}} + \underset{\text{random noise}}{\underset{\downarrow}{\epsilon}}}_{\text{Error}}$$

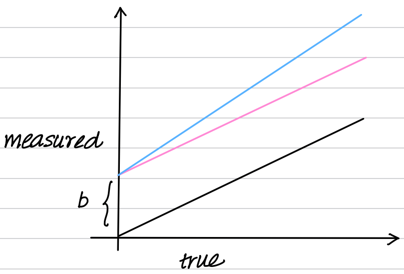
$$\delta a = \underline{\underline{b}} + ka + \underline{\underline{\epsilon}}$$

manufacturer tells you these values

$$\delta v = bt + k\frac{(v-v_0)}{t} \times t$$

ignore this term due to error

$$\delta x = \frac{bt^2}{2} + \underbrace{k(v-v_0)t}$$



Drift – people specify this

After 1 hour, drift should be less than 1 km (some value)

say, $\frac{1}{2}bt^2 < 1000 \Rightarrow b < 10^{-4}$ m/s for $t = 1$ hour

If $b > 10^{-4}$ m/s then drift will be more than 1Km.

For consumer grade accelerometers ~ Drift 3mg

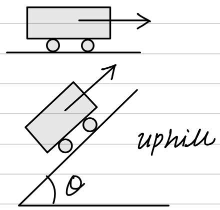
$3\,mg = 3\times10^{-3}\times10 = 3\times10^{-2}\,m/s^2$

Errors ~ 500m just after 5 minutes

Fly for 1hour ~ max error = 1Km ~ for aircraft this much is not a large error

To ↑ accuracy we can combine gyroscope.

How to add effect of gyroscope here?



$a = \frac{kx}{m}$ $\theta = 0$ ~ vanishes

uphill $a = \frac{kx}{m} + g\sin\theta$

θ ~ get from gyroscope

$\dot{\theta} = \dot{\theta}_T + b$ (gyroscope)

(bias)

$\delta_\theta = bt$ (error in θ)

$a = a_T + b + Ka + \epsilon + g\sin\theta$

$\delta a = b + Ka + \epsilon + g b_g t$

$\delta v = bt + K[v - v_0] + \frac{g}{2} b_g t^2$ assume $\sin\theta \approx \theta$ (θ small)

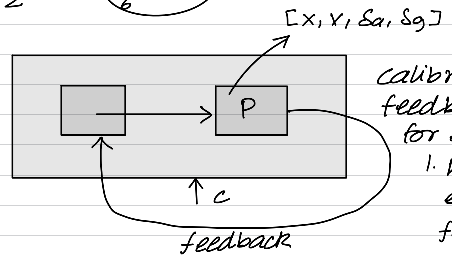
$\delta x = \frac{bt^2}{2} + K[v - v_0]t + \frac{g}{6} b_g t^3$

manufacturer do calibration ~ to correct for errors

---

gravity value also here ~ magnify error + cube power

$\frac{1}{2} b_1 t^2 + \left(\frac{g}{6} b_g t^3\right) < 1000$

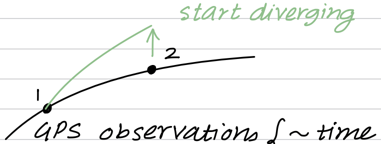


calibration ~ to estimate errors

feedback mechanism to correct for errors.

1. position, velocity, error for accelerometer
error for gyroscope
fed back to correct them!

to find error? need to find absolute system ~ GPS system



errors assumed constant but they vary with time

GPS observations { ~ time b/w 1 & 2 small ~ good estimate
~ " " " " large ~ can't estimate error }

values small ~ drift happens slowly MATLAB code ~ Lab2 ~ error(drift)

Bias 1. fixed
2. turn on bias ~ remains fixed as long as not turn off
~ everytime turn on, I get a new bias
3. in run bias ~ bias is not stuck (bias change)

manufacturer $\angle = \overset{\checkmark}{\times}$

noise ~ temperature dependent ~ not vary randomly ~ models

Some people say ~ Nothing is Random

Whatever we cannot measure we call it random

eg:- no. of red cars while going to z-square mall

If I know all red cars in kanpur and current position and trajectory I can estimate but I don't really know.

systematic random

$$a = a_T + \boxed{(\quad) + \epsilon}$$

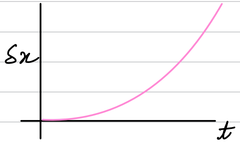
Last class we looked what happens if we don't model these errors

$$S_x = \frac{1}{2} b_a t^2 + \frac{1}{6} g b_g t^3$$

$$\frac{1}{2} b_a t^2 + \frac{1}{6} g b_g t^3 < 1000 \sim 1km \text{ (say)}$$

(some threshold)

Plug $t = 3600$ sec (1hour) ~ upper bound on b



Bias itself is also not constt

Bias also vary w.r.t. time

Bias
- Fixed : remains fix once turn on — turn on bias / fixed bias
- Dynamic : vary with time

Bias ±α ~ fixed part

Bias Instability ±β ~ dynamic part ~ manufacturer can only estimate std. dev. exact bias you have to estimate

	Fixed	
t_1	b_f	b_1
t_2	b_f	b_2
t_3	b_f	b_3

~ turn off instrument & turn on again

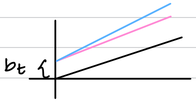
~ every time t_i turn off, turn-on bias get changed but remain fixed throughout.

$$\boxed{b_t = b + b_D}$$

at time t

fixed part

Dynamic part



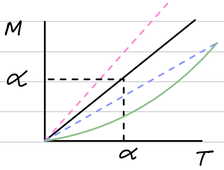
comes from ±β (bias instability)

$$\boxed{b_t = b_{t-1} + \epsilon}$$

random noise $\epsilon \sim N(0, \sigma^2)$

- This is true for both accelerometer and gyroscope
- Initial bias ~ from mean of still observation for 'x' time
- Bias is also along x, y, z ~ three axis

Scale Factor



scale factor ~ either slope > 1 or < 1

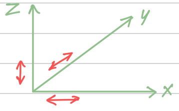
$b_M = K b_{\tilde{M}}$ (people happy with this estimate)

scale can be non-linear as well, then

$$b_M = \sum_i K_i b_{\tilde{M}}^i$$

Cross Coupling

3 axis acc.
3 axis gyro. } sensor placed at three axis



89.95 ~ Angle not exactly 90°

Because of this even at still, values are not zero but some component

$a_x = 0.012$ $a_y = 0.053$ $a_z = 9.83$

(some values)

$f_z = a_z$ if no cross coupling

$f_z = a_1 + \alpha_1 a_2 + \alpha_2 a_3$ 1,2,3 ~ sensor axis (x,y,z)

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 1 & \alpha_1 & \alpha_2 \\ \beta_1 & 1 & \beta_2 \\ \gamma_1 & \gamma_2 & 1 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

measured

cross coupling matrix $= I$ if no cross coupling

$\neq I$ if cross coupling

actual axis where you apply acc^n components of a_x, a_y, a_z

people often combine scaling + cross coupling into one matrix

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} S_1 & \alpha_1 & \alpha_2 \\ \beta_1 & S_2 & \beta_2 \\ \gamma_1 & \gamma_2 & S_3 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

$$a_M = a_T + b + M a_M + \epsilon$$

measured

true value

bias

scaling errors + cross coupling (combined)

random noise

$\epsilon \sim N(0, \sigma^2)$

no guidelines need experience to choose it ~ science +art on how to choose this.

Manufacturer usually do calibration ~ to remove Mam
so error due to Mam gets vanished in model

$a_M = a_T + b + \epsilon$ ~ most people use this model they don't worry about scaling + cross coupling and assume manufacturer removed that by calibration.

NOW, how to choose $\sigma = ?$

Random Walk

1. velocity random walk ~ for accelerometer
2. angle random walk ~ for gyroscope

VRW ~ $m/s/\sqrt{s}$
ARW ~ degree or radian per $\sqrt{\text{hour}}$

$$\dot{\theta} = \dot{\theta}_T + \epsilon$$
$$\theta_1 = \theta_0 + \dot{\theta}\Delta t + \epsilon\Delta t$$
$$\theta_2 = \theta_1 + \dot{\theta}\Delta t + \epsilon\Delta t$$

Grades

1. MEMS
2. Tactical
3. Marins

↓ cost ↑es
performance ↑es

↑ error ↑es

RLG ✓
FOC
Mechanical → Best

not available (US only)

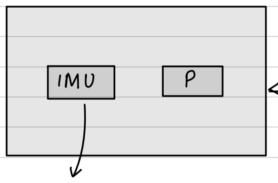
missile, fighter jet (USA)
you can't buy legally
they won't sell you.

if you are able to manufacture, you'll be rich :)

gyrocompassing ~ people in ships use gyroscope as a compass from gyroscope ~ find N ~ mostly in ships ~ by measuring earth's rotation gyroscope is not affected by magnetic fields unlike compass ~ that's why use gyro as compass in ships.

MATHEMATICS PART

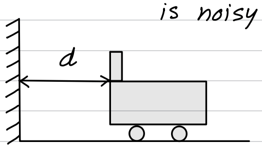
least squares and sequential least squares



This entire setup is INS/AHRS.

IMU (called IMU only)

sensor ~ measure the distance to the vehicle from observation is noisy ~ the noise is coming from random error



BLUE
(Best Linear Unbiased Estimator)

$V \sim N(0, \sigma^2)$
$d_t = d^T + V_t$

Case 1: Crane not moving

- Mean distance $\hat{d} = \frac{\sum d_i}{n}$
- If after every 1 hour, the noise is changing, the estimation will be weighted.

estimated $\hat{d} = \frac{\sum w_i d_i}{\sum w_i}$

When $n \to \infty$
$\hat{d}$ approaches true value

How to arrive at this eqn that the weighted average gives you the best estimate.

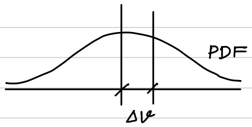
$$V_1 + d_1 = d$$
$$\vdots$$
$$V_i + d_i = d$$
$$\vdots$$
$$V_n + d_n = d$$

OR

$V = AX - L$
$X = [d]$

$$V = \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix} \quad L = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix}$$

· Why least squares? OLS ~ why even root? why square only? OLS(min e_{sos})



$= f(v_i)\Delta v$

$V \sim N(0, \sigma^2)$ Random Noise

$$\text{Probability}(V = v_i) = \frac{1}{\sqrt{2}\pi 6^2} e^{-v_i/6^2} \Delta v_i$$

$v_i = d - d_i$

$$P(V) = \pi P_i = e^{-\Sigma v_i^2/\sigma}$$ ~ maximize this probability

sum of squares of residuals
(This is from normal distribution assumption)

Least squares ~ assumes normal distribution of residuals. If residuals are not normally distributed then least squares won't give you the best estimate. Here, we are trying to find the value of d such that probability of occurence of d i.e. P(d) is highest. That is going to be the best estimate.

~ From least squares we are trying to derive kalman filter.

Case 2 : crane start moving

In this case true value d is changing at every time distance.

$d_1 + v_1 = d_1^t$ $t=1$ d_1

$d_2 + v_2 = d_2^t$ $t=2$ $\frac{d_1+d_2}{2}$

$\vdots$

$d_n + v_n = d_n^t$ $t=n$ $\frac{\Sigma d_i}{n}$

How to find true distances? n-equations & n-unknowns

Least squares is time consuming here, so I don't want that. The estimated value can be found out by the equation

d_t d_i

$$\hat{d}_{t+1} = \hat{d}_t + \text{update} = \hat{d}_t + K(\underbrace{\hat{d}_t - d_i})$$

update: sequential least squares
K: scale factor
$\hat{d}_t - d_i$: new knowledge

Now to find K, use the concept of that when varience (6^2) is minimum then it gives the best estimate.
For any random variable x, defn of mean and std. dev.

Mean $\bar{x} = E(x) = \int f(x) x\,dx$

Varience $\Sigma = E([x-\bar{x}][x-\bar{x}]^T) = E[(x - E[x])^2] = 6_x^2$

For a general case, Sequential least squares

X_t

$X_t = X_{t-1} + K$ (new info. or knowledge)

$X_t = X_{t-1} + K(y_t - y_i)$

y_t: actual measurement
y_i: predicted measurement $= HX_{t-1}$

$y = HX$

y: estimate
H: identity (in this case)
X: measurement

$$x_t = x_{t-1} + K(y_t - y_i)$$
$$= x_{t-1} + K(y_t - HX_{t-1})$$
If I know K, I can find x_t

To find K,

$\Sigma = E[[X - X_{t-1}][X - X_{t-1}]^T]$ (varience)

plug in these values → minimize this Σ w.r.t. unknown K

$\frac{\partial \Sigma}{\partial K} = 0$ ~ solve and find K

Why minimize? bcoz we are assuming noise is coming from normal distribution.

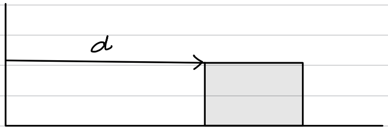
$$K = P_{t-1} H^T (H P_{t-1} H^T + R)^{-1}$$

P_{t-1}: varience-covarience matrix
R: noise in observations $(n \times n)$

$P_{t-1} = E([X - X_{t-1}][X - X_{t-1}]^T)$ varience-covarience matrix for these measurements

R = measured noise ~ comes from knowledge about sensor

Foundation for Kalman Filter ~ make sure to understand this.



{ ^ means estimate }

$X = X_t + K_t (Y - H X_{t-1})$

$\sigma_0^2, d_0 \rightarrow d_0$

d_1, σ_1^2

$d_1 = d_0 + K(d_1 - d_0)$

$K_1 = \sigma_0^2 (\sigma_0^2 + \sigma_1^2)^{-1}$

$K_1 = \frac{\sigma_0^2}{(\sigma_0^2 + \sigma_1^2)}$ Plug in K

$$d_1 = d_0 + \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2}(d_1 - d_0)$$

$$d_1 = \frac{d_0\sigma_0^2 + d_0\sigma_1^2 + d_1\sigma_0^2 - d_0\sigma_0^2}{\sigma_0^2 + \sigma_1^2} = \frac{d_0\sigma_1^2 + d_1\sigma_0^2}{\sigma_0^2 + \sigma_1^2}$$

(weighted average) $w_i \propto \frac{1}{\sigma_i}$

If $\sigma_0 = \sigma_1$, $K = \frac{1}{2}$

After 2 observations, giving exact same thing.

$\hat{d}_1 = \frac{w_0 d_0 + w_1 d_1}{w_0 + w_1}$ ~ varience covarience of this

$\hat{\sigma}^2 = \frac{\sigma_0^2}{4} + \frac{\sigma_0^2}{4} = \frac{\sigma_0^2}{2}$

$d_2 = \hat{d}_1 + K(d_2 - \hat{d}_1)$

$K_2 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} = \frac{\frac{1}{2}\sigma_0^2}{\frac{1}{2}\sigma_0^2 + \sigma_1^2}$ only depends on

σ_0^2 ~ previous estimate quality.

σ_1^2 ~ current estimate quality.

$\hat{d}_2 = \frac{d_1 + d_2 + d_3}{3}$ if all σ's are equal

Law of propogation of varience $c = \alpha a + \beta b$

$\sigma_c^2 = \alpha^2 \sigma_a^2 + \beta^2 \sigma_b^2$ varience co-varience matrix

---

$y_k = f(x_k) + \epsilon_k$

K - time instant

we want to estimate

x_k given y_k

observed / measured

estimated parameter

$$\boxed{y_k = H_k x_k + \epsilon_k}$$

measurement matrix (design matrix)

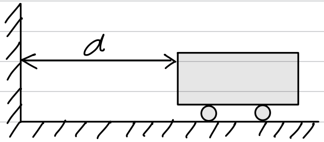
measurement noise

$\epsilon_k \sim N(0, \Sigma)$ Normal distribution

R_k

linear function here.

In case of GPS, non-linear function x_k ~ position y_k ~ pseudorange



measurements → d_k

parameter → $\hat{d}_k$

(later call something else)

$d_k = \hat{d}_k + e_k$

$K = 0$ $d_0 = \hat{d}_0$

$K = 1$ $d_1 = \hat{d}_1 + \epsilon_1$

$d_0 = \hat{d}_0 + \epsilon_2$

} $\hat{d}_1 = \hat{d}_0 = \hat{d}$ since stationary

now 2 eqn 1 unknown

Now we want to apply sequential least squares.

$$\boxed{\hat{x}_k = \hat{x}_{k-1} + K_k \underbrace{(y_k - H_k \hat{x}_{k-1})}}$$

Gain (scaling factor)

Innovation = $y_k - \hat{y}_k$

old info (prediction)

new info (observation)

$\epsilon_k \sim N(0, R_k)$

$$\boxed{K_k = P_{k-1} H_k^T \underbrace{(H_k P_{k-1} H_k^T + R_k)}^{-1}}$$

computationally complex (takes most time)

$\hat{x}_0 = d_0$

$P_0 = \sigma_0^2$

} initial conditions

$\hat{x}_1 = d_0 + K_1(d_1 - d_0)$

$K_1 = \sigma_0^2 (\sigma_0^2 + \sigma_0^2)^{-1}$ $K_1 = \frac{1}{2} \Rightarrow \hat{x}_1 = \hat{d}_1 = \frac{d_0 + d_1}{2}$

$\hat{x}_2 = \hat{x}_1 + K_2(d_2 - d_1) = \left(\frac{d_0 + d_1}{2}\right) + K_2\left(d_2 - \frac{(d_0 + d_1)}{2}\right)$

$$K_2 = \frac{\sigma_0^2}{2}\left(\frac{\sigma_0^2}{2} + \sigma_0^2\right)^{-1} = \frac{1}{3}$$

$$\hat{x}_2 = \frac{d_0+d_1}{2} + \frac{1}{3}\left(d_2 - \frac{d_0+d_1}{2}\right) = (d_0+d_1)\left(\frac{1}{2}-\frac{1}{6}\right) + \frac{d_2}{3} = \frac{d_0+d_1+d_2}{3}$$

If we keep doing this
for 3^{rd} time instant $\quad \hat{x}_3 = \frac{d_0+d_1+d_2+d_3}{4}$

Now with this modification, I am able to do the exact same thing with the added advantage that I don't need to do batch estimation. This is helpful when your datasize becomes large and you cannot accomodate all of your data in memory at once. Also, helpful when you want to do processing in real time.
The most time consuming step in this process computationally is the calculation of the inverse in K_k. Otherwise it is all simple matrix multiplication.

So, this is Sequential Least Squares.

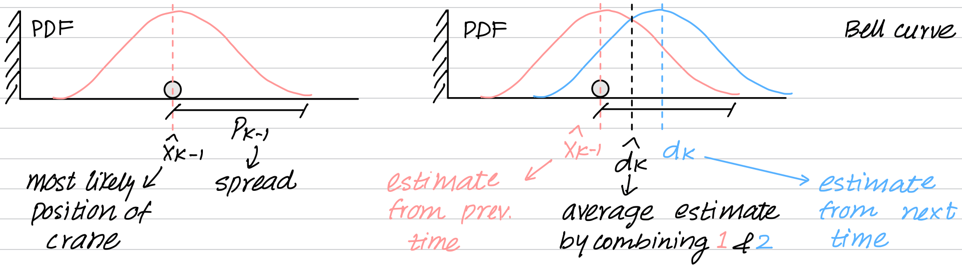
$$\hat{x}_k = \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1})$$

$$\hat{x}_k = (I - K_kH_k)\hat{x}_{k-1} + K_ky_k$$

$$P_k = (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T$$

P_k = varience covarience matrix for parameter at k
R_k = " " " for y_k
K_k = gain = Kalman Gain for Kalman Filter
$(y_k - H_k\hat{x}_{k-1})$ = Innovation = actual measurement − predicted measurement

Let's see what is the meaning of this in the context of this example



We are simply taking the weighted average. If the spread σ_0^2 is same then it becomes the normal average.

$d_1 \quad \sigma_1^2$
$d_2 \quad \sigma_2^2$ } combine these two by taking weighted average

Everytime I am doing a weighted average. It is turning out weighted average because
1. This is a normal distribution
2. This is a linear problem
If this is a non-linear problem this won't be simple weighted average. So then, how to tackle the case for non-linear problem (like GPS)?

$$\rho = \sqrt{(x-x_\mu)^2 + (y-y_\mu)^2 + (z-z_\mu)^2} + \qquad \text{pseudo-range non-linear}$$

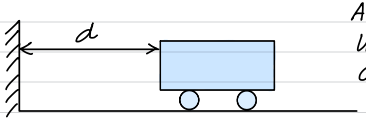
We will linearize this using Taylor series & ignore H.O.T.

$$y_k = f(x_0) + \left.\frac{\partial f}{\partial x}\right|_{x=x_0}(x-x_0) + \epsilon$$

Here, H_k will now change at every instant.

~Lab3: non-linear problem— least squares

Kalman Filter

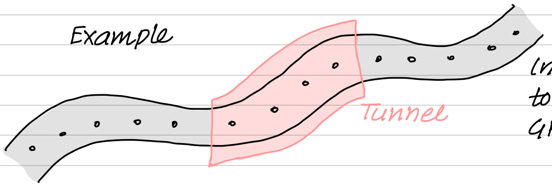


An optimal estimation algorithm.
Use deterministic + statistical properties of the system parameters and measurements to obtain optimal estimates. (Bayesian)

Same example of crane ~standing stationary
Why use Kalman Filter and not sequential least squares?
What is the advantage of using Kalman Filter?

If the sensor was working perfectly everytime you are taking the measurement then you could find out where you are but for some reason the inertial sensor stops working for some time then your least squares will not give you any solution. Least squares can only work whenever you get this sensor working because it relies on this model. $y_k = H_kx_k + \epsilon_k$

Example

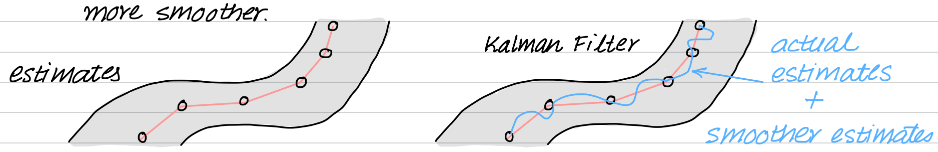


Inside Tunnel ~ gets harder to estimate position through GPS since no open sky present

At each epoch you get pseudoranges ~ solve to get position.
For some reason you don't get pseudoranges for some epoches.
Someone jammed GPS, it stopped working or came under tunnel
Didn't get pseudoranges ~ can't find position estimates

① Kalman Filter helps you overcome this situation when GPS not working. It helps predict the position estimates. That may be inaccurate but you get the position estimates.

② By using Kalman Filter, the results that I get are much more smoother.

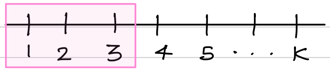


KALMAN FILTER

Name of person who invented it
Rudolf E. Kalman (1930-2016)

Why called filter?

Three terms in optimization
prediction
filtering
smoothing



Prediction Use observations till time K to predict parameter at time K+1. the notation for prediction is:

$\hat{x}_{K+1|K}$ $\hat{x}^{-}_{K+1}$ ↙ minus indicates prediction

Filtering use all observations upto K+1 to get parameter at K+1.

$\hat{x}_{K+1|K+1}$ $\hat{x}^{+}_{K+1}$ ↙ plus indicates filter

---

Smoothing use observations until K+1 time but predict parameter in the previous time. It is something that happens in the past ~ go back and revise past.

$\hat{x}_{K|K+1}$

Filtering ~ can happen in real time
Smoothing ~ can only happen once you have accumulated all the observations. ~ only post-processing.

↑ won't talk about in this course

Kalman Filter = predictor corrector combination
(filter) ↙

Make prediction then do the correction in Kalman Filter.

~ Now we are not calling it parameter ~ terminology
We are calling it state in Kalman filter

X = state vector (state)

this is the parameter you want to estimate which is usually denoted by x. It could be pos^n, vel., acc^n, etc anything that you want to estimate.

Z_K = measurements (measurements)

these are independent observations you take.

For GPS ~ state = position measurement = pseudorange

X_K ~ random variable ~ mean $\hat{x}_K$ and varience covarience matrix P_K
$\hat{X}_K$ ~ estimate
(simply covarience in kalman)

R_K = measurement covarience ~ means noise in measurements
P_K = covarience for state

H_K = measurement matrix ~ tells relationship b/w state vector and measurements

measurement model

$$\boxed{z_K = H_K x_K + \epsilon_K}$$

It establishes relationship b/w what you are measuring and what you want to estimate

motion model or state propogation model

This new model in case of kalman filter that tells you how state propogates over time. This model was not available in case of least squares.

$$\boxed{x_K = f(x_{K-1}) + \epsilon_K}$$ This is based on your understanding.

$$\hat{x}_K = \hat{x}_{K-1}$$

$$\hat{x}_K = \hat{x}_{K-1} + v_{K-1} \Delta t$$

Now with this understanding, let's see this example

$$x_K = d_K$$

$$\hat{x}_{K|K-1} = \hat{x}_{K-1|K-1}$$

Because this is static, position at K is going to be same as position at K-1.

correct motion model would be

$$x_{K|K-1} = x_{K-1|K} + \epsilon_K$$

estimate still the same because ϵ_K has zero mean.
we include this random noise component. why?
to account for any uncertainties.
Using motion model I predict what my state is going to be.

I get observations z_K and use them to adjust this.
I can correct for errors using z_K

$$\hat{x}_{K|K} = \hat{x}_{K|K-1} + K(z_K - H_K \hat{x}_{K|K-1})$$

This is how it is going to run

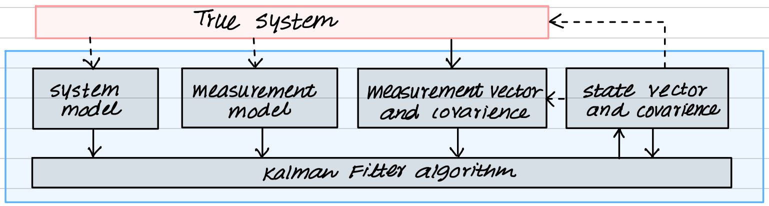
1. prediction
2. correction

If observations not available then we can only predict.

$$P_{K|K-1} = P_{K-1|K-1} + Q$$

Q = process noise matrix
If Q is low then I have more confidence in predictions.
If Q is high then I have less confidence in predictions.

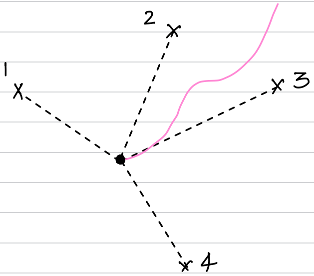
- Beauty of Kalman Filter ~ use upto K+1 observations to estimate K+1 state, do correction and predict again. The beauty is that we intuitively use this every day. Example of prediction-correction model ~ All of us have build models inside throughout our lives. This person going to behave like this if put in a condition like that. We do that to everybody. But that model may not be correct. This model is different for everybody. Once I get observations. based on what happens actually we apply correction & update.
- Apollo project ~ very first application of Kalman Filters
- 5 core elements of Kalman Filter



Initial estimate

In case of least squares, the choice of initial estimate do not have any impact on estimated solution. A good or poor initial estimate just ↓ or ↑ the no. of iterations but both lead to same final estimates.

But when we talk about sequential least squares or the Kalman Filter if your initialisation is wrong, whole solⁿ will be wrong. Initial estimate is very very important there.



6_0^2 $t=1$
26_0^2 $t=2$
⋮
$t=10$

$d_1, d_2, d_3, d_4 \longrightarrow (x_1, y_1)$

$x_K = x_{K-1} + K(y - \bar{y})$
$K = (\qquad)^{-1}$

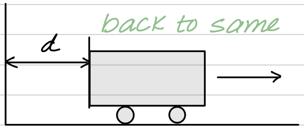
Assignment ~ Kalman Filter estimate
Solve first using least squares then use that as initial estimate for the Kalman filter and find the estimates.

Kalman Filter ~ works in a prediction – correction mode

- State
- Motion Model or State transition model
- Measurement Model or Observation model
- Measurement Matrix
- Measurement noise covarience (R)
- Process noise covarience (Q)
- Measurement vector

KF
↗ predictor ↘
↖ corrector ↙

back to same example



General motion model:
$x_K = F_K x_{K-1} + \epsilon_K$
here, $x_K = x_{K-1} + \epsilon_K$ i.e. $F_K = I$
$\hat{x}_K = \hat{x}_{K-1}$
$\to \epsilon_K \sim N(0, 6_Q^2)$
Random Error ~ always keep it some value keep very small if we believe no randomness but we never keep it zero.

	x_0	P_0
$t=0$	d_0	6_0^2

x_0: initial state
P_0: initial covarience
motion model : $\hat{x}_{K|K-1} = \hat{x}_{K-1|K-1}$
$\hat{x}_{1|0} = d_0$
$P_{1|0} = 6_0^2 + 6_Q^2$ → assumed zero covarience

Predicted covarience: $P_K = F_K P_{K-1} F_K^T + Q_K$ ~ made one assumption

we know for $c = a+b$, $6_c = 6_a^2 + 6_b^2 + [\quad]$
we assume P_K & Q_K uncorrelated ~ varience co-varience = 0

Assumptions

① Process noise (Q) is uncorrelated with covarience (P).
② AWGN ~ Additive White Gaussian Noise

- $E(\epsilon_K \epsilon_{K-1}^T) = 0$ (zero covarience) ϵ is uncorrelated over time } White Noise
- $E(\epsilon_K \epsilon_{K-1}^T) \neq 0$ ~ Coloured Noise ~ ways to handle that exist as well ~ won't discuss here.

Particle Filter ~ takes care of the case when noise is not gaussian. If it is gaussian it simplifies to kalman. This is computationally very expensive and people not able to use it for a lot of real world applications. One reason is we don't know what distribution it follows, we assume normal distribution. Other we assume the noise to be white but it is never white, it is colored.

---

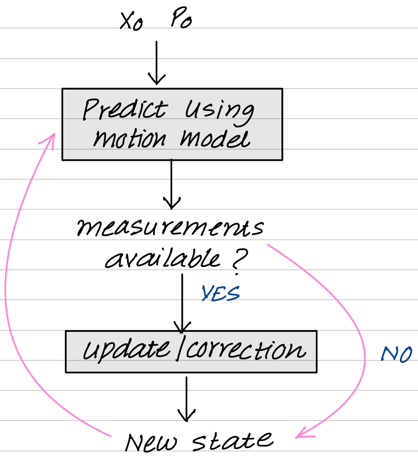
measurement : $z_K = H_K x_K + V_K$ $\quad V_K \sim N(0, R_K)$
$d_K = I \cdot d_K + V_K$
(both assumptions made here)

	x_0	P_0
$t=0$	d_0	6_0^2

x_0 : state $= d$
P_0
$\hat{x}_{K|K-1} = \hat{x}_{K-1|K-1}$
$\hat{x}_{1|0} = d_0$
$P_{1|0} = 6_0^2 + 6_Q^2$
$\hat{x}_{K|K} = \hat{x}_{K|K-1} + K(Z - \bar{Z})$
$\hat{x}_{1|1} = d_0 + K(d_1 - d_0)$
$K = P H_K^T (HPH^T + R)^{-1}$
$P \to \hat{P}_{K|K-1}$

$x_K = F_K x_{K-1} + \epsilon_K$
$x_K = x_{K-1} + \epsilon_K$
$\hat{x}_K = \hat{x}_{K-1}$
$\epsilon_K \sim N(0, 6_Q^2)$
$F_K = I$
$z_K = H_K x_K + V_K$
$V_K \sim N(0, R_K)$
$d_K = I \cdot d_K + V_K$
$\bar{Z} = H_K \hat{x}_{K|K-1}$

What happens when you don't have observations? Then you don't do correction. You just make predictions and treat that as your estimate.



Initialisation ~ very very imp. for Kalman filter.

If measurement not available new state = predicted state

How different from sequential least squares?
No prediction in sequential least squares ~ limitation of it.

Notations revise :—

∧ means estimate ~ at K

$\hat{X}_{K-1|K-1}$ means used observations upto K-1 to estimate at K-1.
$\hat{X}_{K|K-1}$ means used observations upto K-1 to estimate at K.
$P_{K|K-1}$ corresponding covarience at K using K-1

$$\underset{\text{updated}}{\hat{X}_{K|K-1}} = \underset{\text{predicted}}{\hat{X}_{K-1|K-1}}$$

(stationary case)

constant velocity case

$$\hat{X}_{K|K-1} = \hat{X}_{K-1|K-1} + \underbrace{V_{K-1|K-1}}_{\text{need velocity here}} \Delta t$$

~ motion model

$$X_K = \begin{bmatrix} X_K \\ V_K \end{bmatrix} \begin{matrix} \text{pos}^n \\ \text{vel.} \end{matrix}$$

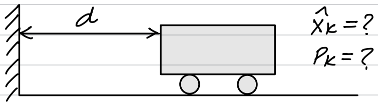
even if no motion still case use → $V_K \approx 0$ (very small)

accelerating case

accelerometers on board. ~ acc^n also given now. f_b, b_a ← bias
with bias in readings.
later we can move 2D to 3D scenario.

---

case 2 : constant velocity motion



$\hat{X}_K = ?$
$P_K = ?$

1. Identify state
2. Identify measurement
3. Establish self-transition model (motion model)
4. Establish measurement model
5. Establish covarience reasonably
6. Initialise properly (X_0, P_0)
7. Make suitable assumption and estimate

$$X_K = \begin{bmatrix} r_K \\ V_K \end{bmatrix} \qquad Z_K = \begin{bmatrix} d_K \end{bmatrix}$$

$$X_K = F_{K-1} X_{K-1} + \epsilon_{K-1}$$

$$\begin{bmatrix} r_K \\ V_K \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_{K-1} \\ V_{K-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{r,K-1} \\ \epsilon_{V,K-1} \end{bmatrix}$$

$$Z_K = H_K X_K + \epsilon_{2K}$$

$$d_K = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} r_{K-1} \\ V_{K-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{r,2K} \\ \epsilon_{v,2K} \end{bmatrix}$$

$$r_K = r_{K-1} + V_{K-1} \Delta t + \epsilon_r$$

$$v_K = v_{K-1} + \epsilon_v$$

$$d_K = r_K + \epsilon_{d_K}$$

$$R_K = E[\epsilon_{zK} \epsilon_{zK}^T] = \sigma^2$$

$$Q_K = E\left[\begin{pmatrix} \epsilon_r \\ \epsilon_v \end{pmatrix} \begin{pmatrix} \epsilon_r \\ \epsilon_v \end{pmatrix}^T\right] = \begin{bmatrix} \sigma_r^2 & \sigma_{rv} \\ \sigma_{vr} & \sigma_v^2 \end{bmatrix} = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}$$

← { processed noise

1. r and v, uncorrelated
2. $\sigma_r \to 0$ but $\neq 0$

You stand for 5 minute ~ take average of all observations treat that as the initial position estimate.

$$r_0 = \frac{\Sigma d_i}{n}$$ (initial position)

$$\sigma_{r_0}^2 = \frac{\sigma^2}{n} \rightsquigarrow \sigma_{r_0}^2 = \frac{n\sigma^2}{n^2} = \frac{\sigma^2}{n}$$ (law of propogation of variences)

$$\hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} \qquad P_{k|k-1}$$

$$= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_v \\ V_0 \end{bmatrix}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \underbrace{(d_k - \bar{z}_k)}$$

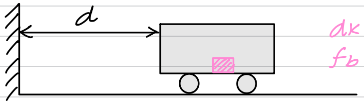
correction ~ small if accurate prediction, large if not.

$$\bar{z}_k = H_k \hat{x}_{k|k-1}$$

$$= r_{k|k-1} = r_{k-1|k-1} + V_{k-1|k-1}\Delta t$$

when no observations ~ you only rely on prediction.

Case 3: Accelerating Crane



d_k

f_b ~accelerometer onboard

$$x_k = \begin{bmatrix} r_k \\ V_k \\ b_k \end{bmatrix} \qquad P_0 = \begin{bmatrix} \\ \\ \end{bmatrix} \qquad z_k = [\,d_k\,]$$

↳ acceleration can also be part of state if you want

$$\left.\begin{aligned} r_k &= r_{k-1} + v_{k-1}\Delta t + \epsilon_r \\ v_k &= v_{k-1} + (f_{k-1} - b_{k-1})\Delta t + \epsilon_v \end{aligned}\right\} \text{state transition model}$$

$$b_k = b_{k-1} + \epsilon_b$$

Accelerometer has bias + noise. ~estimate bias also as part of state.

fixed bias   unknown bias

Gauss–Markov sequence is a quantity that varies with time as a linear function of its previous values and a white noise sequence ie. b_k depends only on b_{k-1} and not b_{k-2}.

$$b_k = b_{k-1} + \epsilon_k \quad \text{where } \epsilon_k \sim N(0, \sigma^2) \text{ white noise}$$

White Noise sequence is a discrete-time sequence of mutually uncorrelated random variables from a zero mean distribution.

For white noise w_i, $E(w_i w_j) = \begin{matrix} \sigma_w^2 & i=j \\ 0 & i \neq j \end{matrix}$

$$\hat{x}_{k|k-1} = F_{k-1}x_{k-1|k-1} \qquad P_{k|k-1}$$

$$= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_0 \\ V_0 \end{bmatrix}$$

$$\bar{z}_k = H_k \hat{x}_{k|k-1} = r_{k|k-1} = r_0 + V_0\Delta t$$

$$\begin{bmatrix} r_k \\ v_k \\ b_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & -\Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{k-1} \\ v_{k-1} \\ r_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ f_{k-1} \\ 0 \end{bmatrix} \Delta t + \epsilon_{k-1}$$

Now my model has somewhat changed.

$$x_k = F_{k-1}x_{k-1} + G_{k-1}U_{k-1} + \epsilon_{k-1}$$

control input

something you are applying externally to control the system.

$$d_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_k \\ v_k \\ b_k \end{bmatrix} + \epsilon$$

If you are standing stationary, initial state will be
initial position = 0
initial acc^n = 0
take average of acc^n, whatever value you get is bias.
take that as initial bias, it includes turn on bias + fixed.
as long as you don't turn off, no remain same.

General Motion Model

$$\underset{n\times1}{x_k} = \underset{n\times m}{F_{k-1}}\,\underset{m\times1}{x_{k-1}} + \underset{n\times m}{G_{k-1}}\,\underset{m\times1}{U_{k-1}} + \underset{n\times1}{\epsilon_{k-1}}$$

$$z_k = H_k x_k + v_k$$

three coveriences:—

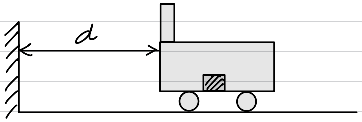
1. $E(\epsilon_{k-1}\epsilon_{k-1}^T) = Q_{k-1}$ (process noise coverience)
2. $P_0 \sim x_0$ (initial coverience)
3. $E(v_k v_k^T) = R$ (measurement noise coverience)

P → easiest one ~ because if you understand how good your measurements are, estimates become better.
eg:- GNSS – code pseudoranges
It tells confidence that you have on this value. If very sure assign low value, if not sure assign high value.

Q → dirtiest one → tells how good is your motion model
eg:- driving car in plane vs mountains ~ Q different
conservative vs rash driver ~ Q different

What value to take? It comes from your experience.

The other way is Tuning of Kalman filter. ~ won't talk about it. But the idea is you choose a value of Q, R, then choose a value of P, φ and then check effective behaviour based on observations by varying covariences. This is very cumbersome process. People mostly rely on experience to choose P, Q, R.



multi-sensor positioning

can put any no. of sensors as long as I can relate it.

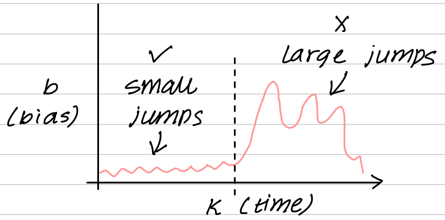
$$\begin{bmatrix} r_K \\ v_K \\ b_K \end{bmatrix}$$

Accelerometer ~ have biases
Gyroscope ~ have biases
} need to estimate in state vector

How to check your estimates are correct or not?

① Bias vs time plot

Plot how bias changes with time. Our assumption was that bias changes very slowly. But if you notice large jumps in bias then some assumptions made are not correct.



② Innovation vector ~ If everything is fine then you will observe that your innovations are white.

$I_K = z_K - \bar{z}_K$

z_K → actual observations

$\bar{z}_K$ → predicted observations

i.e. $E(I_K I_{K-1}^T) = 0$

I_K & I_{K-1} uncorrelated

⇓

implies good value of Q

GPS positioning ~ code observations ~ moving case
~ pseudorange eqⁿs

$$X_K = \begin{bmatrix} r_K \\ v_K \\ c_K \end{bmatrix}$$

position
velocity
receiver clock bias
} state space model for GPS

Assume receiver clock bias follows Gauss-Markov sequence.

$$(C_r)_K = (C_r)_{K-1} + \epsilon_{K-1}$$

Here it is a non-linear case

---

linear case

state vector $X_K = \begin{bmatrix} r_K \\ v_K \\ f_K \\ b_K \end{bmatrix}$ f_K → true value of accⁿ ~ don't have any bias

state transition model

$$r_K = r_{K-1} + v_{K-1} \Delta t + \epsilon_1$$
$$v_K = v_{K-1} + \underbrace{f_{K-1}} \Delta t + \epsilon_2$$
$$f_K = f_{K-1} + \epsilon_3$$
$$b_K = b_{K-1} + \epsilon_4$$

previously it was ($f_b - b$) but now it is true value of accⁿ. previously we were using accelerometer that's why bias.

$$\hat{\begin{bmatrix} r_K \\ v_K \\ f_K \\ b_K \end{bmatrix}} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & \Delta t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \hat{\begin{bmatrix} r_{K-1} \\ v_{K-1} \\ f_{K-1} \\ b_{K-1} \end{bmatrix}} + \epsilon$$

measurement model

$$d_K = r_K + \epsilon_5$$
$$f_b^K = f_K + b_K + \epsilon_6$$

$$z_K = \begin{bmatrix} d_K \\ f_b^K \end{bmatrix}$$
$$z_K = H_K X_K + v_K$$

$$\begin{bmatrix} d_k \\ f_b^k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_k \\ f_k \\ b_k \end{bmatrix} + \epsilon$$

In Kalman Filter, the rates on measurement data collection can be different for different sensors.

accelerometer → 200Hz → 200 observation/sec
distance sensor → 100Hz → 100 " "

Depending on what measurement is available you include or omit it in the measurement model.

S_1 1 Hz
S_2 3 Hz } can still combine them
S_3 50 Hz

Beauty of Kalman Filter
~Need not have sensors working at same rate

<u>Example</u>: at every 1 sec interval → $z_k = \begin{bmatrix} d_k \\ f_b^k \end{bmatrix}$ ~ both acc^n + distance

at every 1/500 sec interval → $z_k = [f_b^k]$ ~ accelerometer

t
(1) ← both position + acc^n
(2)
← only acc^n
(3)

eg:- camera 60fbs
odometer 60Hz
accelerometer 400Hz

So far seen three cases
- case1 stationary
- case2 constant velocity
- case3 constant acc^n

several other variations possible for this.
In Kalman Filter ~ assumption that measurement and motion model is linear. What to do when either of them or both becomes non-linear.

Extended Kalman Filter (EKF)

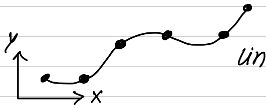
- For non-linear measurement measurement /motion model
- Apollo Mission ~ went to moon with few kbs of RAM. They developed this EKF. they had non-linear measurement model.
- Beauty of this is that I don't need to estimate everything at once.
- Takes care of case when either of x_k or z_k non-linear. somehow try to linearize ~ errors incorporate but okay.
- EKF — all other assumptions same, just non-linear x_k/z_k.
- Developed by NASA ~ for trajectory estimation of moon landing.

After midsem recess →

{ EKF ~ 2D or 3D case ~ EKF for GPS positioning ~ Add INS to it
Explore use cases eg:- Marsian Rover ~ n no. of sensors
cameras, LiDAR, depth cameras, Inertial sensors but there is no GPS. }

Kalman Filter ~ become so popular from economics to electronics to civil everywhere.

$x_{k+1|k} = F_k x_{k|k} + \epsilon_k$
$z_k = H_k x_k + v_k$



linearise KF

Extended Kalman Filter

can get rid of them

$$\rho = \sqrt{(x-x_s)^2 + (y-y_s)^2 + (z-z_s)^2} + b_r + b_s + I + T + \epsilon$$

state vector
$$x_k = \begin{bmatrix} x_k \\ y_k \\ z_k \\ v_x \\ v_y \\ v_z \\ b_r \end{bmatrix}_{n \times 1}$$

measurement
$$z_k = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \rho_n \end{bmatrix}_{n \times 1} = f(x) + v = f(x_0) + \left| \frac{\partial f}{\partial x} \right|_{x=x_0} (x - x_0)$$
$$= H_k x_k + (f(x_0) - H_k x_0)$$

state transition model
$x_k = x_{k-1} + v_{k-1} \Delta t$
$v_k = v_{k-1} + \epsilon$
$(b_r)_k = (b_r)_{k-1} + \epsilon$

$x_0 = \hat{x}_{k|k+1}$ (nominal value about which linearizing)
$H_k \neq$ constt

STEPS FOR EXTENDED KALMAN FILTER

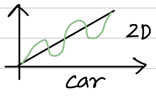
Initialize
↓
State Transition $(\hat{x}_{K|K+1})$
↓
linearize measurement model about $\hat{x}_{K|K+1}$
↓
$H_K, Z_K, \bar{Z}_K \longrightarrow I_K$

Here no need for iterations. Why? Because the initial estimate is very close to true value. Some people still carry out for better accuracy ~ Iterative EKF. But very maginal improvement over EKF. That's why people typically don't do that.

HA1 ~ when you change Q and R, trajectory changes changing Q changes motion model i.e. your understanding of how system behaves. Trajectory estimate on how good you are able to write your model.

eg1: person A lies 40% → shop open } whom to trust?
person B lies 60% → shop close } you decide how well you can capture

eg2: rash car driver
conservative car driver



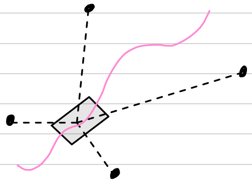
high R low Q
low R high Q

---

state transition model : $X_{K|K+1} = F_K X_K$

F_K is now jacobian i.e. $F_K = \dfrac{\partial f}{\partial x}$

All other assumptions still same, Noise gaussian, sensor observation uncorrelated, etc.
Here we are not deriving the equations, just understand working.
Whenever X_K or Z_K is non-linear, we use EKF.
In books ~ they try to derive generic equations.



$$X_K = X_{K-1} + V_{K-1}\Delta t + \epsilon_1$$
$$V_K = V_{K-1} + R_b^n a_K^b \Delta t + \epsilon_2$$
$$(ba)_K = (ba)_{K-1} + \epsilon$$

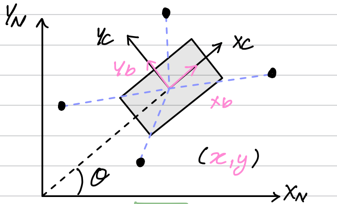
$$R_b^n = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

$$Z_K = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix}$$

acceleration — part of control input
$$X_K = F_{K-1} X_{K-1} + G_{K-1} U_{K-1}$$

CASE-1 : with compass

Assumption — No magnetic interface



Body – Car – Navigation

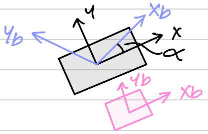
$$X_K = \begin{bmatrix} x_K \\ y_K \\ \theta_K \\ b_{1,K} \\ b_{2,K} \\ v_{x,K} \\ v_{y,K} \end{bmatrix}$$ state vector

θ_K ~ compass gives accurate reading unless no magnetic interference (assumption)

$b_{1,K}$, $b_{2,K}$ → accelerometer bias (can be equal)

2 axis accelerometer ~ 2 acc. co-located, sensitive axis are orthogonal
Ranging sensor ~ measure distance w.r.t. stations
Compass ~ measures heading

What we want? orientation of car frame w.r.t. navigation frame
We install acc. in such a way that body frame is aligned with car frame.



$$a_c = R_b^c a_b$$

$$\begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$

$\alpha = 0 \Rightarrow R_b^c = I$ otherwise we need to know R_b^c if $\neq I$.

---

state vector $$X_K = \begin{bmatrix} x_K \\ y_K \\ \theta_K \\ b_{1,K} \\ b_{2,K} \\ v_{x,K} \\ v_{y,K} \end{bmatrix}$$

measurements : measurement model
at $t = k$; d_1, d_2, d_3, d_4
θ_K

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} + v_i \quad ; \quad i = 1,2,3,4$$

$$\theta_K = \theta_K + v_j$$ ~ measurement model is non-linear

state transition model →

$$\begin{bmatrix} \hat{x}_{K|K-1} \\ \hat{y}_{K|K-1} \end{bmatrix} = \begin{bmatrix} \hat{x}_{K-1|K-1} \\ \hat{y}_{K-1|K-1} \end{bmatrix} + \hat{V}_{K-1|K-1} \Delta t$$

$R_b^n = R_c^n R_b^c$ — if not identity

$$\hat{V}_{K|K-1} = \hat{V}_{K-1|K-1} + R_b^n (a_{K-1} - \hat{b}_{K-1|K-1}) \Delta t$$ ~ non-linear (cos, sin, etc)

$$\hat{\theta}_{K|K-1} = \hat{\theta}_{K-1|K-1}$$ ~ linear Assumption — this changes slowly ☆

$$\hat{b}_{i\,K|K-1} = \hat{b}_{i\,K-1|K-1}$$ ~ linear

Here both state transition and measurement model → non-linear.

$x_0, P_0, Q \longrightarrow F_{k-1} \longrightarrow \hat{x}_{k|k-1} \longrightarrow H_k$

Jacobian →

$$\left.\begin{aligned} F_k &= \frac{\partial f}{\partial x}\Big|_{x=\hat{x}_{k|k-1}} \\ H_k &= \frac{\partial z}{\partial x}\Big|_{x=\hat{x}_{k|k-1}} \end{aligned}\right\}$$

$\hat{x}_{k|k-1} = \hat{x}_{k|k-1} + K_k I_k$; $I_k = z_k - \bar{z}_k$

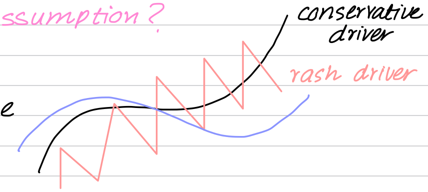
↖ innovation

$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1}$

What is the consequence of this assumption?

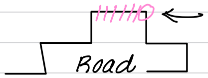
$\hat{\theta}_{k|k-1} = \hat{\theta}_{k-1|k-1} + \epsilon_{k-1}$

↖ Random Noise



Depends on

1. Behaviour of person
2. Dynamics of vehicle/type (bike/car)
   aircraft you can't abrupty change trajectory
   drone you can change abruptly
3. Road type



← accuracy poor at sharp turns

· ϵ_{k-1} low – conservative

· ϵ_{k-1} high – rash driver

Challenge

How to get correct values of P, Q, R?

P → P_0 we get P easily

Q → confidence on your model on model

R → confidence on your measurements — quality of measurements

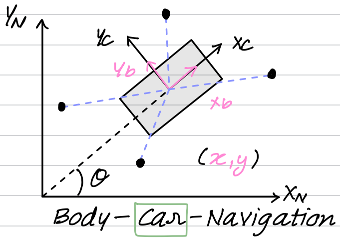
There are various scenarios like feeling sleepy, GPS gave poor accuracy, etc. All these behaviours have to be captured. Getting Q is a part of stochastic model. Capture them in $\epsilon_k \rightarrow Q$ & R ~ randomness

People say getting a filter to run is art + science.

↓

get Q, R (beliefs)

CASE-2 : with gyroscope



$$x_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ b_{1,k} \\ b_{2,k} \\ v_{x,k} \\ v_{y,k} \end{bmatrix}$$ state vector

2 axis accelerometer

Ranging sensor

single axis gyroscope

ω_{ib}^b

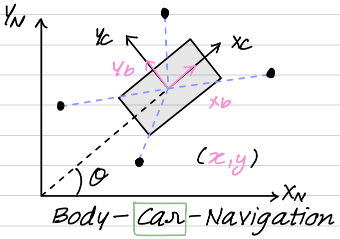
$\dot{\theta} = \omega$

$\theta_k = \theta_{k-1} + \square \omega t$

know this in body frame ~convert in navigation frame

$\theta_{k-1} + R\omega_{ib}^b \Delta t$

CASE-3 : with gyroscope + compass both



$$x_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ b_{1,k} \\ b_{2,k} \\ v_{x,k} \\ v_{y,k} \end{bmatrix}$$ state vector

2 axis accelerometer

Ranging sensor

single axis gyroscope

compass

Have a cup of tea or coffee!
write all eqⁿs on a sheet of paper without seeing anything from notes for all three cases

Best way to learn!
write all matrices clearly on paper.

1. compass
2. gyroscope
3. gyroscope + compass both

what is size of Z_K? 5x5
H_K? 5x7
F_K? 7x7 / 8x8
V_K? 3x1
G_K?

$$x_K = F_{K-1} x_{K-1} + G_{K-1} U_{K-1} + \epsilon$$

↳ $G_K = ?$

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \\ b_1 \\ b_2 \\ \theta \\ a_n \end{bmatrix}_K$$

$f_b = R_n^b a_n + b$

f_b ← accelerometer reading

a_n ← true value

If you want to add this to measurement, you have to add a_n in state vector

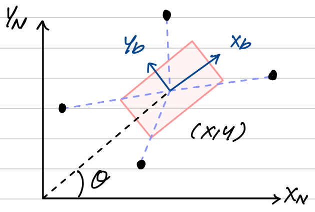
$z = HX$

$\theta_K = \theta_{K-1} + \omega \Delta t$

how this measurement related with what you are observing i.e. state vector?

$(\theta_K)_m = \theta_K + \vartheta$

GNSS/INS Integration



Ranging sensor
2- Accelerometer
Compass
1- Gyroscope

Two ways of solving this problem

1) measurement model
$$z = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ \theta \end{bmatrix}$$
Tightly Coupled

· use raw observations

---

2) apply least squares to distance to get the positions.
Instead of using distances in measurement, I want to use those computed positions as part of measurement.
By doing so, I remove the non-linearity in the measurement model.

$$z = \begin{bmatrix} x \\ y \\ z \\ \theta \end{bmatrix}$$
Loosely Coupled

$(x)_{in} = x_K$
· use computed observations

eg:- GPS sometimes gives you pseudoranges ~ use tightly coupled
GPS sometimes gives you positions ~ use loosely coupled

Advantage of tightly coupled systems :-

① If only two distances available d_1, d_2, you can use tightly coupled $z = \begin{bmatrix} d_1 \\ d_2 \\ \theta \end{bmatrix}$ but you can't do loosely coupled.

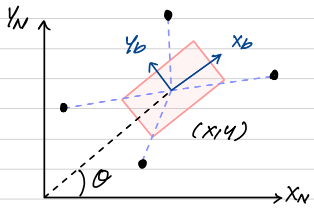
$$z = \begin{bmatrix} x \\ y \\ z \\ \theta \end{bmatrix} \sim \text{can't get derived observations } x, y, z$$

② In loosely coupled, you are using least squares twice. once for computing positions second while doing updates. This is not an optimal solution.

Assumptions so far :

1. ignored gravity
2. ignored rotation of earth
3. 2D case

} now write complete model without these assumptions for full 3D case.



GNSS
3-axis Accelerometer
3-axis Gyroscope

$$X = \begin{bmatrix} \left.\begin{matrix} \boxed{\begin{matrix} x \\ y \\ z \end{matrix}}\ r \end{matrix}\right\} \text{positions} \\ \left.\begin{matrix} v_x \\ v_y \\ v_z \end{matrix}\right\} \text{velocity} \\ \left.\begin{matrix} b_{1,a} \\ b_{2,a} \\ b_{3,a} \end{matrix}\right\} \text{acc. bias} \\ \left.\begin{matrix} b_{1,g} \\ b_{2,g} \\ b_{3,g} \end{matrix}\right\} \text{gyro. bias} \\ \left.\begin{matrix} \phi \\ \theta \\ \psi \end{matrix}\right\} \text{roll, pitch, yaw} \\ \left.b_r\right\} \text{receiver clock bias} \end{bmatrix}_{16 \times 1}$$

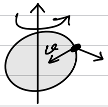
Use GNSS measurements — n satellites available

$Z_{n \times 1} = f(X) + V$ (pseudorange observation model)

$$\rho_i = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} + b_r + v$$

Try to reduce no. of unknowns through some means.

$$\underset{n \times 16}{z} = HX + v$$



State transition model

$$r_{k+1} = r_k + V_k \Delta t + \epsilon$$

$$v_{k+1} = v_k + R_b^n (f_b - b_a)\Delta t + \epsilon - g^n \Delta t - \underbrace{(\vec{\omega} \times \vec{v_k})}_{\text{coriolis force added to compensate for earth's rotation.}} \Delta t$$

includes net gravitational effect gravity added to compensate g in navigation frame.

coriolis force added to compensate for earth's rotation.

$$(b_a)_{k+1} = (b_a)_k + \epsilon$$

$$(b_g)_{k+1} = (b_g)_k + \epsilon$$

$$(b_r)_{k+1} = (b_r)_k + \epsilon$$

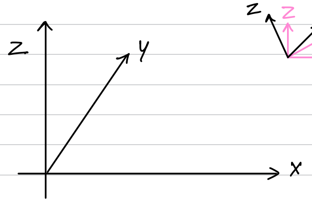
$$\left\{ -2(\vec{\omega} \times \vec{v})\Delta t = -2 \overset{3\times3}{\Omega} V \right\}$$

↓ skew-symmetric matrix of this vector

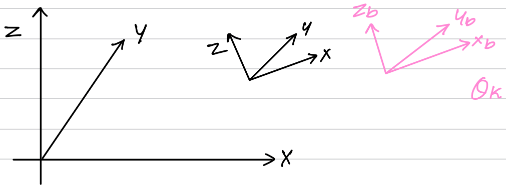
Euler angles

$$\left.\begin{aligned} \phi_{k+1} &= \phi_k + \phi \Delta t + \epsilon \\ \theta_{k+1} &= \theta_k + \dot{\theta} \Delta t + \epsilon \\ \psi_{k+1} &= \psi_k + \dot{\psi} \Delta t + \epsilon \end{aligned}\right\}$$

how to find these rate of changes $\dot{\phi}, \dot{\theta}, \dot{\psi}$?



$\dot{\phi} = \omega_x^b$



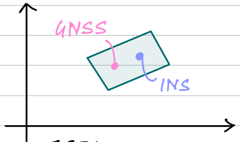
no longer rotation matrix.
(bcoz these are euler angles)

$\theta_K = \theta_{K-1} + R\omega\Delta t$

$$\begin{bmatrix}\dot\phi\\ \dot\theta\\ \dot\psi\end{bmatrix} = \begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix}$$

(need not rem this)

GNSS Integration



ECEF ✓
ECI
ENU

$$\begin{bmatrix} r_K \\ v_K \\ \theta_K \\ b_{a,K} \\ b_{g,K} \\ b_{r,K} \end{bmatrix}$$

$\dot{r} = v$
$\dot{v} = R_b^n (f_b - f_a) - g^n - 2\vec{w_e} \times \vec{v}$
$\dot{\theta} = R(w_{ib}^n - b_g) - w_e$
$\dot{b} = 0$

$$\rho_{i,K} = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} + b_{r,i} + v_{i,K}$$

$z_K = \underset{\substack{\downarrow \\ n\times16}}{H_K} X_K + v_K$

$\dot{X} = f(X)$

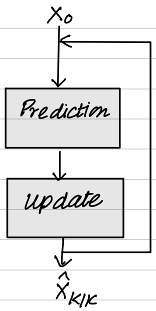
$\frac{X_{K+1} - X_K}{\Delta t} = f(X_K)$

$X_{K+1} = F_K X_K + G_K X_K + \epsilon_K$
$X_{K+1} = X_K + f(X_K)\Delta t$
$X_{K+1} = X_K + \left.\frac{\partial f}{\partial x}\right|_{x=x_0} (x - x_0)\Delta t$

Plug $x = x_K$

$X_{K+1} = X_K + \left.\frac{\partial f}{\partial x}\right|_{x=x_K} (x - x_K)\Delta t$

$X_{K+1} = \underset{n\times n}{(I + F_K\Delta t)} X_K - F_K \underset{n\times1}{X_K} \Delta t$



closed loop

people generally use closed loop and not open loop implementation.

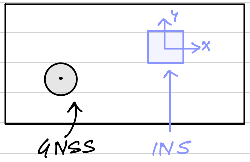
preferred implementation

- Tightly coupled ✓
- Loosely coupled

closed loop ✓
open loop

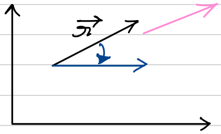
What practical problem faced in the implementation?

Assumption → All sensors located at same point.
This is not practically possible.



l^b

R_b^n



translation — no change in magnitude/direction.

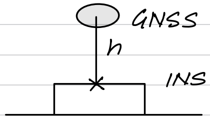
rotation — no change in magnitude but change in direction.

All the computation/estimation you are doing for origin in the body frame. ⊡

$r_b = r_a + R_b^n l^b$ (Lever arm)
Boresight parameter

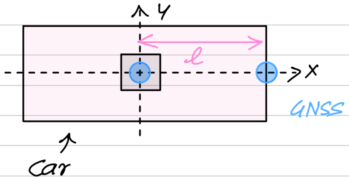
manufactures tells you to drive in this trajectory in order to calibrate

Best practice is to install GNSS directly on top of INS.



· INS typically installed on CoM of object.

more GNSS added



two GNSS receivers
~to get heading information

Quality depends on 'l'

more — more quality — good estimate
less — less quality — bad estimate

Dual Antennas

Helpful when stationary for long period of time. when you stop and stand still there is heading drift due to noise. Dual antennas helps to distinguish and identify real movement from that of noise.

Three Implementation possible

① $$z = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \\ \theta \end{bmatrix}$$

pseudoranges from GNSS

heading angle from INS

② $$z = \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_n \\ \rho_1' \\ \vdots \\ \rho_n' \end{bmatrix}$$

Fully tightly coupled

pseudoranges from two GNSS

computationally very expensive
but very good performance

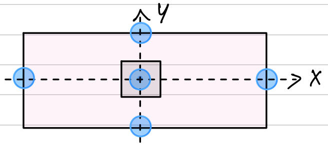
③ $$z = \begin{bmatrix} x \\ y \\ z \\ \theta \end{bmatrix}$$

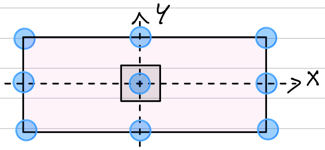
Loosely coupled

position estimates + heading (INS)

---

can also do MULTI-GNSS





can calculate
roll, pitch, yaw
also with this

Price too high ?

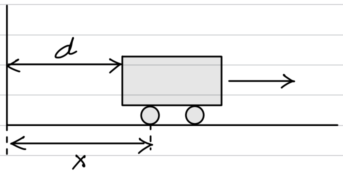
The price of such systems is very high. Not solely due to more no. of GNSS but because of more filters added now and filter has become complex.
— Very few companies that provide this.
— No Indian company, startups can lead here....

MLS (Mobile Mapper) ~ LiDAR

Positioning — by combining various sensors together
Improve positioning quality and precision.
Also helpful one sensor fails or stop working. — The only change you do is adjust measurement model to accomodate available sensors

---

case 1: constant velocity



State, STM

state space Model

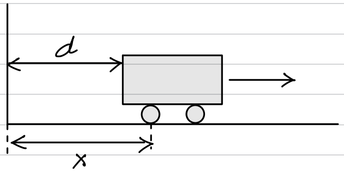
state vector: x_K, v_K
measurement = d_K
$\hat{d}_K = \hat{x}_K$ (measurement model)
$x_K = x_{K-1} + v_{K-1} \Delta t + \epsilon$
$V_K = V_{K-1} + \epsilon$
} (state transition)

$$Q = \begin{bmatrix} \ddots & 0 \\ 0 & \ddots \end{bmatrix}$$

The most challenging part - getting Q. People generally take it from least squares. Once you do estimate from least squares you have some idea.

case 2: accelerating crane

If we now put acceleration in above case, then the eq^n will be



$$x_K = x_{K-1} + v_{K-1}\Delta t + \epsilon$$
$$v_K = v_{K-1} + (\underbrace{a_{K-1} - b_{K-1}}_{\text{bias}})\Delta t + \epsilon$$

accelerometer readings have bias

State, STM

we are discritizing these eq^{n}s

state space Model

$$\dot{x} = \frac{x_{K+1} - x_K}{\Delta t} \Rightarrow x_{K+1} = x_K + v_K \Delta t$$

why bias only in acc^n?

$$\dot{v} = \frac{v_{K+1} - v_K}{\Delta t} \Rightarrow v_{K+1} = v_K + a_K \Delta t$$

Note: If you believe your velocity sensor will have errors then you can consider bias in that as well.

eg:- change default tyres ~ bigger tyres

actual and observed are now diff. vel.

$$z = h(x) + v$$
$$H_K = \frac{\partial h}{\partial x}\bigg|_{x=\hat{x}_{K|K-1}}$$
$$z_K = z_K + H_K(x - x_0) + v$$
$$z_K = H_K x_K + (z_0 - H_K x_K) + v$$

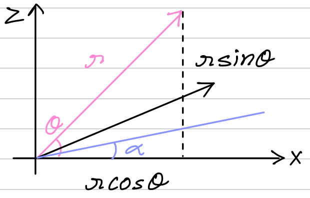
$$\hat{x}_{K|K-1} = \hat{x}_{K|K-1} + K_K I_K$$
$$I_K = z_K - \overline{z}_K$$
$$\overline{z}_K = h(\hat{x}_{K|K-1})$$

QUIZ-2 (KF/EKF)

(open-notes quiz)

① accelerometer put on a vehicle moving with constant velocity in a straight line.
Can you estimate the orientation of sensor using Kalman Filter from accelerometer readings. If yes, how?
Properly define all assumptions, state space model and other models along with proper steps.

Assumptions
1. Error is additive, white, gaussian.
2. Insignificant rotation and acceleration.
(No rotation and acceleration)
3. Rotation of earth not considered
4. No biases in accelerometer observations.
5. Gauss-Markov for θ_K, α_K i.e. roll, pitch changes gradually.



$$x = r\cos\theta\cos\alpha$$
$$y = r\cos\theta\sin\alpha$$
$$z = r\sin\theta$$

$$\tan\theta = \frac{z}{\sqrt{x^2+y^2}} \quad ; \quad \tan\alpha = \frac{y}{x}$$

state vector $x_K = \begin{bmatrix} \theta_K \\ \alpha_K \end{bmatrix}$ $\left.\begin{matrix} \text{roll} \\ \text{pitch} \end{matrix}\right\}$ orientation

measurement model $z_K = h(x_K) + v_K \approx \underset{3\times2}{H_K}\underset{2\times1}{x_K} + \underset{3\times1}{v_K}$

$$z_K = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} g\cos\theta\cos\alpha \\ g\cos\theta\sin\alpha \\ g\sin\alpha \end{bmatrix} + v_K$$

$$H_K = \begin{bmatrix} \frac{\partial f_x}{\partial \theta} & \frac{\partial f_x}{\partial \alpha} \\ \frac{\partial f_y}{\partial \theta} & \frac{\partial f_y}{\partial \alpha} \\ \frac{\partial f_z}{\partial \theta} & \frac{\partial f_z}{\partial \alpha} \end{bmatrix}$$

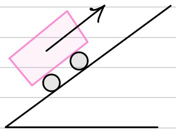
state transition model

$$\left.\begin{aligned} \theta_K &= \theta_{K-1} + \epsilon_{K-1} \\ \alpha_K &= \alpha_{K-1} + \epsilon_{K-1} \end{aligned}\right\} \quad \text{(Gauss Markov)}$$

$$X_K = F_{K-1} \hat{X}_{K-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{K-1} \\ \alpha_{K-1} \end{bmatrix}$$

Kalman Filter :

1. Initial state and covarience X_0, P_0
2. predict using state transition model
3. measurement model
4. correct using Z_K
5. define and update P, Q, R throughout
6. find X_K



If body is non-rotating and non-accelerating, you can use accelerometer to find orientation.
If it is accelerating, you can't use it to find orientation.
eg:- car game playing — phone with no gyro — accelerating
can't control game 

---

② Can you estimate the orientation using reading from IMU sensors (gyroscope, compass and accelerometer) by using an KF/EKF? Write all assumptions, proper steps and models.

compass, accelerometer, gyroscope — sensor fusion with KF/EKF

state vector $X_K = \begin{bmatrix} \theta_K \\ \alpha_K \\ \psi_K \end{bmatrix}$

state transition model

$$\begin{aligned} \theta_K &= \theta_{K-1} + \dot{\theta} \Delta t \\ \alpha_K &= \alpha_{K-1} + \dot{\alpha} \Delta t \\ \psi_K &= \psi_{K-1} + \dot{\psi} \Delta t \end{aligned}$$

assume insignificant rotation of earth

$$\dot{\theta} = R [\omega_{ib}^{b} - \omega - b_g]$$

$$\dot{\theta} = R (\omega_{ib}^{b} - b_g)$$

similarly $\dot{\alpha}, \dot{\psi}$ also.

} to estimate these include them as part of control unit. If acc, gyro considered in state, no need.

---

measurement model

$$Z_K = \begin{bmatrix} f_x \\ f_y \\ f_z \\ \psi_T \end{bmatrix}$$

} accelerometer observations

— assuming no magnetic heading i.e. compass gives true heading
if not there, remove declenation from magn.

$$\psi_T = \psi_M - \Delta$$

ψ_T ↓ true, ψ_M ↓ magnetic, Δ ↪ declenation

$$\left.\begin{aligned} &x \\ &y \\ &z \\ &\tan\theta \\ &\tan\alpha \end{aligned}\right\}$$

To find orientation,
$\theta, \alpha \rightarrow$ KF since linear model
$\theta, \alpha, \psi \rightarrow$ EKF since non-linear model

Assumptions

1. Ignore H.O.T for jacobian
2. Assume no biases
3. others same as before . . .

Aliter: If considered other sensor readings as well then

$$X_K = \begin{bmatrix} f_x \\ f_y \\ f_z \\ \omega_x \\ \omega_y \\ \omega_z \\ \psi_T \end{bmatrix}$$

$$\left.\begin{aligned} &f_x = f_a - b \\ &f_y \\ &f_z \end{aligned}\right\} \text{true values i.e. without bias}$$

f_x, f_y, f_z and θ relationship

$H_K X_K \sim$ highly non-linear problem

If include biases in accelerometer and gyroscope.

$$\left.\begin{aligned} &3 \text{ acc reading} + 3 \text{ acc bias} \\ &3 \text{ gyro reading} + 3 \text{ gyro bias} \\ &3 \text{ orientation} \end{aligned}\right\} X_K = 15 \times 1 \text{ vector}$$

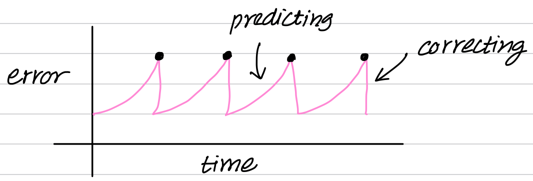
This is called AHRS
This will tell you roll, pitch and yaw only using accelerometer and gyroscope.

By all this in AHRS, you will only get orientation. You won't get positions. But in full INS you get that as well.

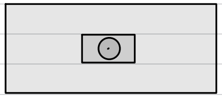
When you go to market to buy, he will ask whether you want AHRS or a full INS.

AHRS	INS
gives orientation only	gives orientation + position
less cost	more cost
simpler KF	complex KF
no GPS	GPS ✓

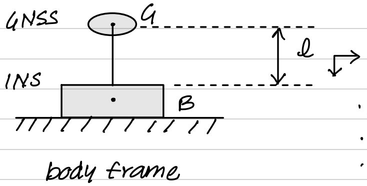
So these were quiz solⁿ, you can answer them now if asked somewhere.



GNSS/INS Integration



GNSS, INS are not co-located.



lever arm

- lever arm is always in body frame
- if you know, treat it as constant
- if you don't know treat as part of state

state vector $x_k = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}$ B is point on INS

measurement model $z_k = \begin{bmatrix} x_G \\ y_G \\ z_G \end{bmatrix}$ G is point on GNSS

If both sensors (GNSS and INS) are co-located.

$$\begin{bmatrix} x_G \\ y_G \\ z_G \end{bmatrix}_k = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}_k + v_k$$

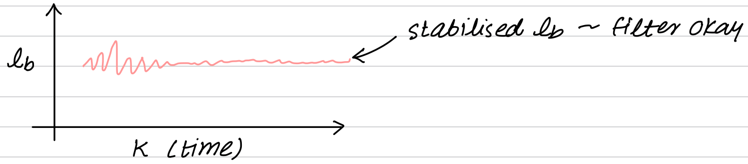
If they are not co-located, as here, the model will be

$$\begin{bmatrix} x_G \\ y_G \\ z_G \end{bmatrix}_k = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}_k + R_B^G l_b + v_k \quad \Big\} \text{ non-linear model here}$$

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

$$= z_0 + \frac{\partial h}{\partial x}\Big|_{x=x_0} (x - x_0)$$

What is the quality check?



leverarm values should be constant over time, then the filter is stabilised.

Assumption was l_b = constant, if this was correct then you should get a constant value of l_b over time.

Two ways to calibrate
1. calibrate in lab beforehand, you keep l_b as constant
2. on-site calibration → keep l_b as unknown value and estimate along state. (l_b as part of state)

Better to calibrate on the go ~ one more computation. because the value is subject to change.

state transition model for ℓ_b

$$(\ell_b)_K = (\ell_b)_{K-1} + \epsilon_{K-1}$$

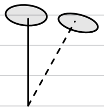
Keep covarience very very small because you want to Keep ℓ_b constant.

Q ~ constraints in KF

can do that in state transition model or measurment model

several reasons for ℓ_b to change

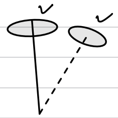
- nuts and bolts get loose
- ℓ_b gets tilted after few years
- expansion/contraction of material



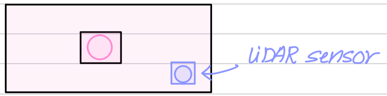
because of these — Keep doing calibration again and again ~on the go ~drive in figure of 8 trajectory, it will calibrate itself.

Note: ℓ_b has three components

$$\ell_b = \begin{bmatrix} \ell_x \\ \ell_y \\ \ell_z \end{bmatrix}$$



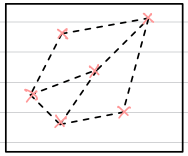
both situations can be modelled in ℓ_b.

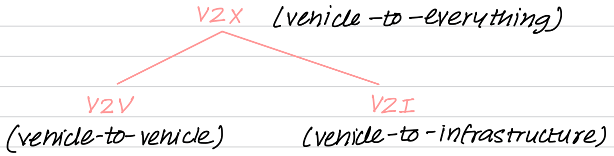


~ Boresight parameters

APPLICATIONS PART

Cooperative Positioning (collaborative or peer-to-peer positioning)




V2X (vehicle-to-everything)

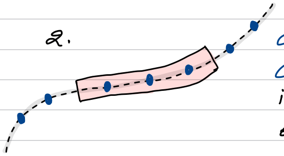
V2V (vehicle-to-vehicle)

V2I (vehicle-to-infrastructure)

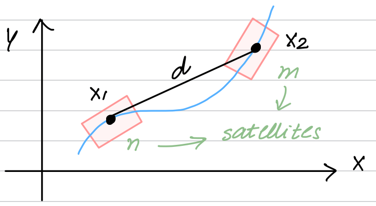
you make them talk to each other and exchange information directly you can incorporate relative positioning also using sensors.

Advantages:

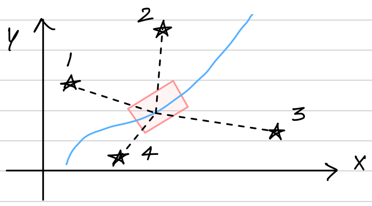
1. By applying the constraint, the uncertainty in position can be reduced. i.e. increased accuracy.

2. 

cars inside tunnel
cars outside tunnel } you can make them talk to each other and share some information about distances and all and then each of them can localise themselves.

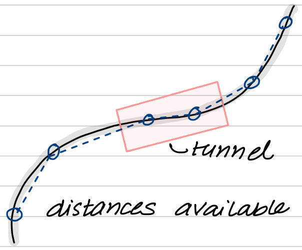


X1 & X2 changing (distance b/w two cars available)



1,2,3,4 — fixed instrastructure (distances w.r.t. them available)

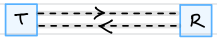
GNSS
Ranging
Gyroscope
Accelerometer
Compass
Barometer
Magnetometer
Vision



sensors used to measure distance

1. Laser range finder
   disadvantage: you need clear line of sight

2. UWB (Ultra-Wideband)
   advantage: you need not have clear line of sight. It can also pass through walls. Of course quality degrades somewhat but okay and workable unless so many walls. They are cheaper as well (5 units cost ≈ ₹ 10,000)



With UWB, people made DSRC.

put them in cars

UWB ⟶ DSRC (Dedicated Short Range Communications)

DSRC enables vehicles to communicate with each other and other road users for direct wireless exchange of V2X & ITS. Australian company trying to install DSRC in vehicles and infrastructure.

Apple technologies

1. FindMy — your device send a signal to other nearby apple devices and that sends securely to iCloud and you can find the location of your lost iPhone.

2. Airtag — It uses Ultra-Wideband (UWB). The U1 chip uses UWB to measure distance and direction b/w devices. In this way they communicate with your apple devices and FindMy.

Find My ~ communicate to nearby Apple devices
Airtag ~ uses UWB for positioning

---

centralized cooperative positioning

$$\text{state} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

concatenation of x_1 and x_2

$$x_0,\ P_0 = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix}$$

initially assume uncorrelated
they became more correlated with time

↓

$$\dot{x} = f(x) + \epsilon$$

(state transition model)

$$(x_1)_k = (x_1)_{k-1} + v_1 \Delta t$$

$P_{k|k-1}$

↓

$$\underset{\downarrow \atop n+m+1}{z_k} = \underbrace{h(x_k)}_{\downarrow \atop H} + v_k$$

(measurement model)

$P_{k|k}$

$$d_k = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

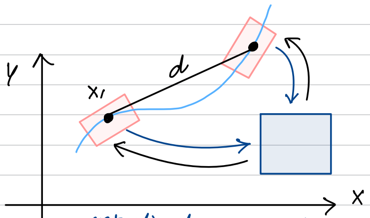
Assumption: transition of both cars are independent of each other (independent movement)

problem: can't have too big / complex network, then the computations ↑, laser ranger don't work after 100m and a lot of limitations for a large network.

$$P_0 = \begin{bmatrix} P_1 & \square \\ \square & P_2 \end{bmatrix}$$

~ off-diagonal elements won't be zero anymore

The more they talk to each other, the more they become correlated.
If you know these correlations then its good.
If you don't know the correlations then becomes a problem.



centralised approach

Using the distance available, can estimate position

Issues →
1. as network grows, computations ↑es
   network shouldn't be too large.
2. need constant communication.

eg:- if two person know each other and they talk to each other then their behaviour and thinking becomes correlated. similarly, when you live with parents you are correlated with them but once you come to college and live here for a long you do not remain correlated with them anymore.

Distributed cooperative positioning

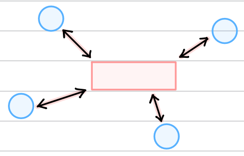
x_1
ρ_1
$\vdots$
ρ_n

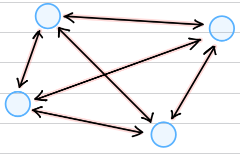
$$P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix}$$

initially assume uncorrelated but as they interact, they become correlated

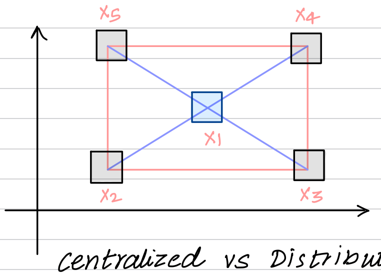
$$d = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$$

In this problem correlation has become unknown and I dont have any way of finding what is its correlation. This is unsolved problem till date, the reason being people don't know how to estimate it well. They use some statistical technique or minimize correlation.
↪ open research problem


centralized approach


distributed approach

↓
centralized better than distributed.
↓
better estimates



$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \quad d = \begin{bmatrix} d_{12} \\ d_{13} \\ d_{14} \\ d_{15} \\ d_{23} \\ d_{25} \\ d_{34} \\ d_{45} \end{bmatrix}$$

Centralized vs Distributed

More number of cars, better the connectivity and network.
Minimum requirement depends on the configuration

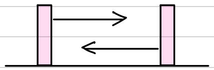
What happens when none of the cars have GPS ?

Relative positioning will be good but
Absolute positioning won't be there.

So, you need few of them to have GPS installed on them.

What you measure and share need not neccesarily be distances. It can be anything that you can exchange.

UWB



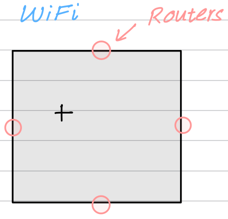
Ultra Wideband

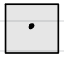
transmit pulse radio to measure distance

WiFi

WiFi Fingerprinting

①




room-level accuracy

RSSI
(Receiver signal Strength Indicator)

map signal strength across all locations now from current signal strength I can map where I am and find position within mapped area.

Indoor Localization

Stage 1 : Training phase ← called Wifi-Fingerprinting
Prepare RSSI map correlating location with signal strength (RSSI)

Stage 2 : Positioning phase
Match current signal strength to trained RSSI Map to estimate distances and thus, position.

Issues with WiFi

1. Room Level Accuracy
   positioning accuracy of around a room, because the signal strength dont vary much from one pt. to another within a room.

2. Map sensitive to room configuration
   location vs strength map is not constant, it changes with the configuration of room. Any rearrangement or changes in room layout affect RSSI map requiring updates for accurate positioning.

3. Indoor Positioning

This works best for indoor environment not outdoor because outdoor environment changes much more rapidly

4. Symmetry (not unique)

If there is a symmetrical building i.e. everything in it is symmetrical then you would not get unique signal strength everywhere.

5. Router Location

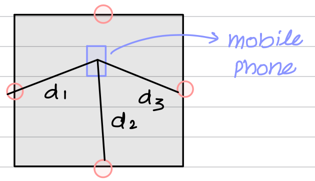
Affected by location of routers in the room. If you add or more routers (turn on hotspot) then the map changes.

WiFi works well only when everything is known precisely or if environment is stationary. It gives room level accuracy. It means you can be anywhere in this room.

---

② Distance based methods

Use SS to estimate distance



$$SS \propto \frac{1}{d}$$

SS ↓ signal strength

d → distance

Path Loss Models : empirical models that relate SS to distance.

$$d = \sqrt{(x-x_i)^2 + (y-y_i)^2} \qquad \text{and } \quad Kn$$

It works well for fixed setup environments like malls, factories industries, etc where there is either no movement or movement in a predefined manner i.e. there is no randomness.

It also has room level accuracy but with less fluctuations.

SoOp (Signals of Oppurtunity)

All those signals that were not originally intended for positioning or navigation but can now be used for positioning / navigation.

eg: UWB, WiFi, Cellular (3G/4G/5G), etc.

WiFi5 + New Hardware

With new standards and hardwares, I can calculate the time it takes for signal to travel from device to destination and come back, called Wi-Fi RTT (Round Trip Time).

Now from m level accuracy → deci-metre level accuracy positioning just because RTT being measured accurately.

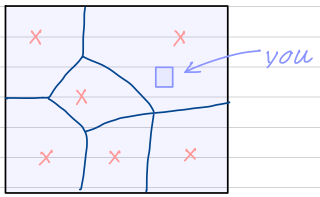
$$2d = (\text{WiFi RTT}) \times C$$

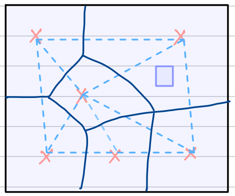
But the limitation is it is good for indoor only.

---

eg:- City with cellular towers

X ≡ cellular towers





Delaunay Triangulation

From every tower the signal you'll get will be of different strength. Ideally the strongest signal received will be from nearest tower.

The cell areas are constructed by using delaunay triangulation. The size of the cells depend on the density of the cell towers.

No. of towers ↑, size of the cells shrink and the accuracy ↑ now.

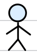
positional accuracy estimated based on signal strength of nearest tower.

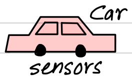
5G signals gives better accuracy and can measure RTT also which is not possible with 4G signals.

4G ↓ (Km level)

RTT ↓ (m level accuracy)

Research → comprehensive and robust positioning system





WiFi
Cellular

GPS
UWB ~ cars
WiFi

~ Can I make use of all these to develop a comprehensive and robust positioning system?

~ How do I combine all of these inertial sensors along with WiFi so that I am able to know position everywhere all the time?

~ Bigger research question ~ people trying to solve for the last few years and still not resolved. → ongoing research

~ Can I know my position everywhere I go?

At the heart of all of this lies Kalman Filter and its variants to solve the problems.

Whatever covered, just tip ~ so much more to talk about only in the positioning itself.

Ashwani ~ Indoor positioning using Inertial sensors and UWB for around 4 years now.