

Indoor Location: Inertial Tracking and Pedestrian Dead Reckoning

Dr Robert Harle

Part II

Easter 2012/13

Part II
Easter 2012/13

The diagram illustrates a cantilever-based mass sensor. The schematic on the left shows a central green mass with a double-headed arrow indicating vertical movement. This mass is connected to a cantilever structure. Labels include: "Fixed" at the top support, "C1" and "C2" for capacitors on the cantilever, "Outer plate" for the fixed base plate, "Proof mass (moves)" for the central mass, and "Spring" for the cantilever's restoring force. The SEM image on the right shows the physical device with labels: "BEAM" (the cantilever), "MOVABLE PLATE" (the proof mass), "FIXED PLATES" (the base), and "ANCHOR" (the base connection). Technical specifications at the bottom of the SEM image are: 1.00kV, 20.0k x5, 15µm, 401 - 10µm, 17 FEB 05, AMT-10.

- Have a proof mass between springs and a series of 'plates'
- Measure deflection via capacitance changes
- 1-D only

-

Inertial Tracking (Dead Reckoning)

- **Relative** positioning – we compute position by concatenating a series of position deltas
- Used in robotics (deltas usually come from wheel revolutions)
- Used in vehicular navigation (esp. submarines). Deltas come from integrating the accelerations after correcting for various things like orientation, gravity).

A white and grey electronic device, likely an inertial navigation system (INS) or dead reckoning unit, with a handle on top and various ports and buttons on the front panel. The device has a rugged, industrial design with a handle on top and a front panel featuring several ports, buttons, and a small display or indicator. The front panel has a label that reads "INS" and "VLSI". There are also some smaller labels and a red square logo on the bottom right of the front panel. The device is shown from a three-quarter perspective, highlighting its compact and functional design.

-

- Based on measuring Coriolis force as experienced by a moving object in a rotating frame of reference
- Many implementations but the 'tuning fork' method is most common

$\vec{F}_{Coriolis} = -2m\vec{C} \times \vec{v}$

Driven horizontally @ resonance in opposing directions

Coriolis produces vertical displacement

-

Inertial MEMS

- MicroElectroMechanical Machines (MEMS) are revolutionising electronics
- Now have tiny, cheap inertial sensors (accel, gyro)

Accelerometer MEMS size evolution

All are 3-axis accelerometers
USB values are Production Cost

MEMS Size (mm²)

Year

ST (MEMS100)

ST (MEMS100)

VTI (MEMS1000)

SORCER (MEMS100)

KETEX

ST (MEMS100)

ADXL146

ST (MEMS100)

\$0.0-0.075

\$0.075-0.1

\$0.1

Hand holding a red rectangular sensor module.

Hand holding a smartphone displaying a home screen with various app icons.

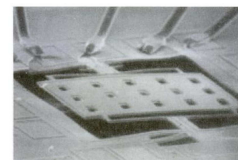
-
- Accelerometer MEMS size evolution**
- All are >2-3x accelerometers
<1/5th values are Production Cost
- | Year | Device | MEMS Size (mm²) |
|------|--------------|-----------------|
| 2007 | ST (0.01616) | ~10.5 |
| 2008 | ST (0.0101) | ~5.5 |
| 2009 | VTI CH4009 | ~4.5 |
| 2009 | 0.0061 | ~3.5 |
| 2009 | 0.0051 | ~3.0 |
| 2009 | 0.0041 | ~2.5 |
| 2010 | ST (0.0031) | ~2.2 |
| 2010 | ADXL345 | ~2.0 |
| 2011 | 2011 (0.001) | ~1.5 |
- Legend: All are >2-3x accelerometers, <1/5th values are Production Cost



- Most use Lorentz Force
- A current-carrying wire in a magnetic field experiences a perpendicular force

The diagram illustrates a microelectromechanical system (MEMS) actuator. It features a rectangular loop of current-carrying wire (Aluminium) in a magnetic field B_x . The Lorentz force causes the loop to move. The device is supported by a support beam and includes polysilicon piezoresistors for sensing. The diagram also shows L-beams and a current path. A legend indicates that black squares represent Aluminium and white squares represent SiO_2 . The micrograph shows the physical device with a grid of piezoresistors and a central loop structure.

-
- The diagram shows a microfabricated Hall sensor. It features a central rectangular loop with L-shaped beams at the corners. Polysilicon piezoresistors are located on the L-beams. A current flows through the loop in a clockwise direction. A magnetic field B_s is applied perpendicular to the plane of the loop. The device is supported by a support beam. A legend indicates that black areas represent Aluminum and white areas represent SiO_2 . Arrows indicate the direction of current and motion.

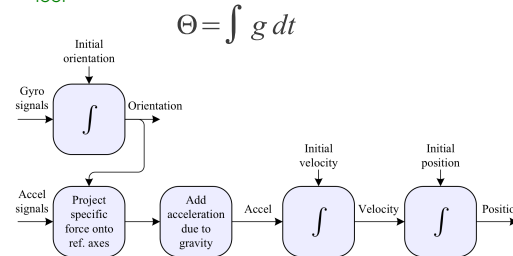


Sources of Error

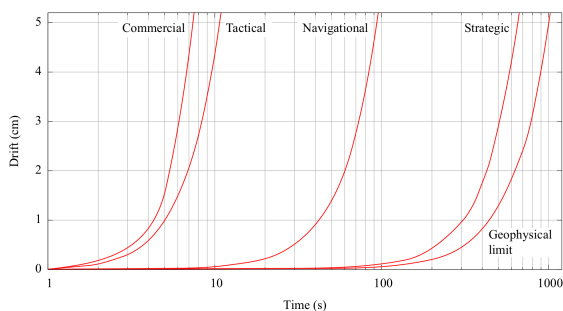
- When things get this small, all sorts of error sources crop up. Common ones:
 - Bias** – this is the offset from the true value. It can be calibrated out, except that it is temperature dependent at this scale!
 - White Noise** – Standard additive measurement noise
 - Bias Instability** – The bias may exhibit fluctuation
 - Ageing** – gradual chemical changes to the crystals
 - These are often very serious as we will see...

Adding Rotation: Strapdown Systems

- If we allow free rotation we need to use the gyroscopes to track the orientation, then resolve the accelerations into the world axis, then double integrate
- BUT: gyros give us rotational rates – we need to integrate them too!**



Sources of Error



Pesky Errors Again...

- Gyro bias gives a linear error growth in orientation
- Accelerometer bias gives quadratic error growth
- Therefore we get cubic error growth. :-)**
- Actually orientation errors are usually the biggest problem
- Imagine perfect acceleration signals. If we are a degree out in rotation and we try to subtract gravity, we get residual accelerations that are then integrated. Twice.
- (Usually in these systems the hardest thing is to remove gravity)*

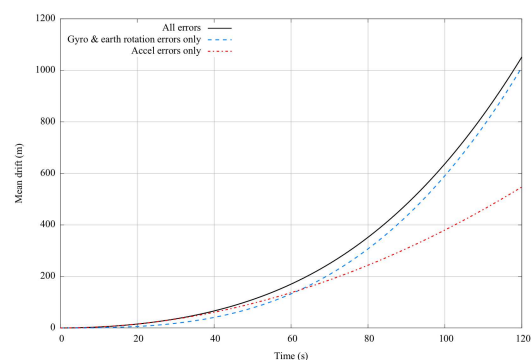
Positioning from Linear Acceleration

- If the pose of the device is constant, double integrating the accelerometers **after removing gravity** should give displacement

$$s = \iint (\vec{a} - \vec{g}) dt$$

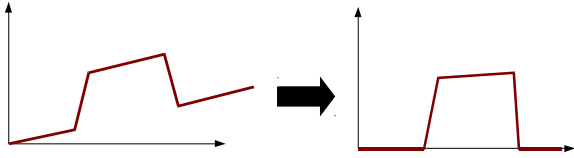
- However, this means **bias** introduces error that grows quadratically with time
- Double-integrating **white noise** produces a random walk
- End result is a fast (and unlimited) accrual of error

Simulation Example



So What Do We Do?

- To make progress, we need to apply **external constraints** to the system to correct the drift before it gets too big.



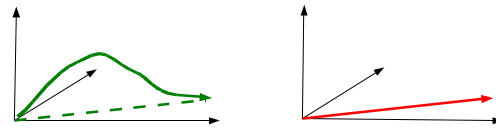
Limitations of ZUPTs

- ZUPTs typically allow tracking to ~1% of true distance travelled.
- It can be difficult to spot ZUPT periods in complex motions
- The sensors must be on the foot!

ZUPTs

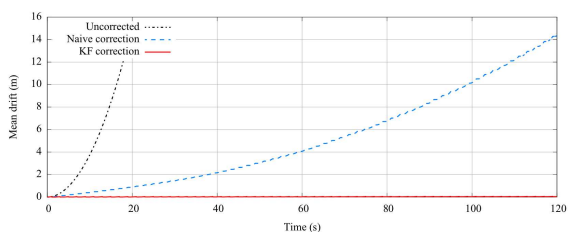
- Zero velocity Updates
- Applicable when the **sensors are on the shoes of pedestrians** (where's the iShoe when you need it?)
- Simple idea
 - When foot is in stance phase, it is not moving
 - Therefore sensors not moving
 - Therefore any sensor data is just noise
 - So we can recalibrate our sensors when the foot is down

Step-and-Compass Systems



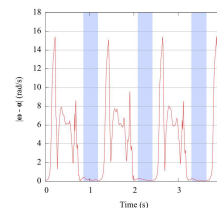
- Don't bother tracking the detailed foot trajectory in a stride
- Try to characterise each step in terms of a length and direction

Results



Step Detection

- Now we can use step detection algorithms that just try to spot a step taking place



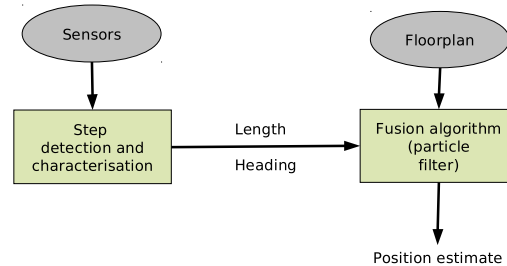
- Often done using just accelerometers
 - Thresholding
 - Frequency search

Step Length

- Without integrating the signals we don't have an accurate length for each step :-)
- Three common approaches
 - 1) Assume length is constant
 - 2) Pre-calibrate the step duration/step length function for each person (turns out to be ~linear)
 - 3) Model the length as a random variable at some higher level of processing

Fusing Building Maps

- Around 2008 we decided to try adding a floorplan to the mix



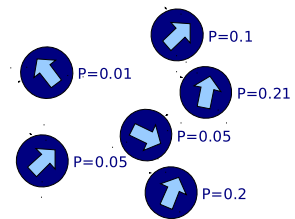
Step Direction

- Compass → absolute heading, but non-Gaussian noise on short timescales
- Gyro → relative heading, but drifts on long timescales
- Therefore combine both outputs to form gyro-stabilised compass

Particle Filters for Location

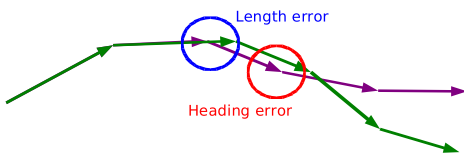
- Each particle is an individual hypothesis about the position and orientation of the user
- Each particle has an assigned probability

- Particles are propagated by:
 - Update
 - Correct
 - Resample



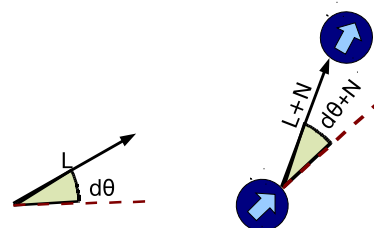
Drift...

- ...still going to occur from missed/extra steps, error in length estimate or heading errors



- Need to find some more absolute constraints like ZUPTs (but not ZUPTs because the phone won't be on the foot)

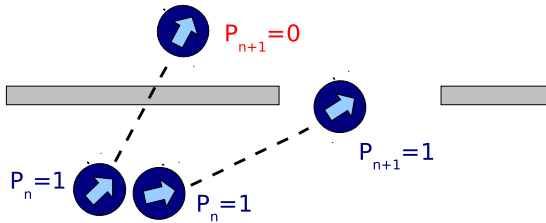
PF: Update



- Each particle is moved by the measured step length and direction
- Plus some additive noise

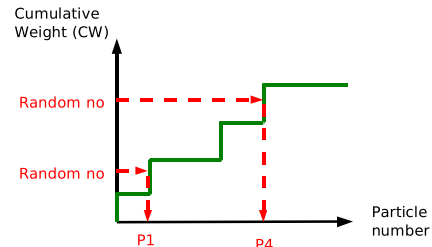
PF: Correct

- Using any info that you have, reassign the particle probabilities
- For walls we can simply set $p=0$ if the particle crossed a wall and leave it alone otherwise



PF: Resample

- Then generate uniform random number a between 0 and $\max(CW)$ and read off the particle to clone



PF: Resample

- Want to get rid of $p=0$ particles but still need some particles!
- We **resample**: generate a new particle set by sampling the old one **in proportion to the particle weights**
- $P=0 \rightarrow$ won't go any further
- $P=1 \rightarrow$ may be reproduced (multiple times) if chosen at random

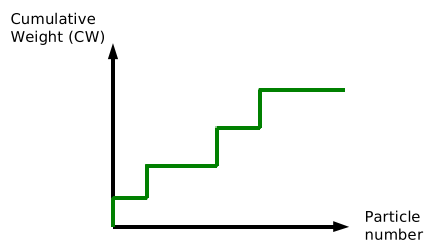


A Note on Performance

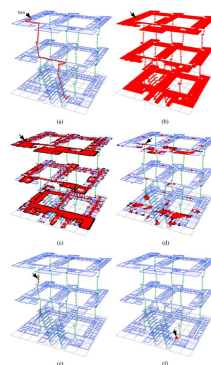
- Update and correct steps are nicely parallelisable
- But forming the cumulative weight for resampling is fundamentally sequential...

PF: Resample

- To select in proportion form a cumulative weight...

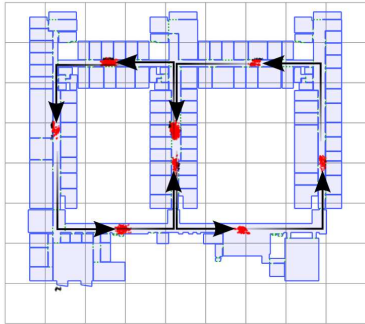


Works well...

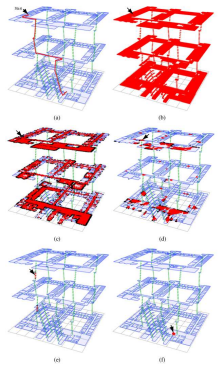


- Initially we have no knowledge of the user's position
 - Lots of particles
 - Localisation Phase

Symmetry Problem



Works well...



- Eventually we figure out where they are and the problem becomes easier
 - Fewer particles needed
 - Tracking Phase
- Can get ~ 0.75m accuracy 95% of the time

Outstanding issues

- Sensor foot-mounted
- PF processing costly esp. if we are in localisation phase
- Step detection not perfect
- Tracking can fail