Virtual multi-antenna arrays for radio transmitter bearing estimation

or

How to do synthetic aperture radar with cell phones ?

François Quitin
Université libre de Bruxelles (ULB), Belgium

Virtual AOA estimation / Synthetic aperture radar

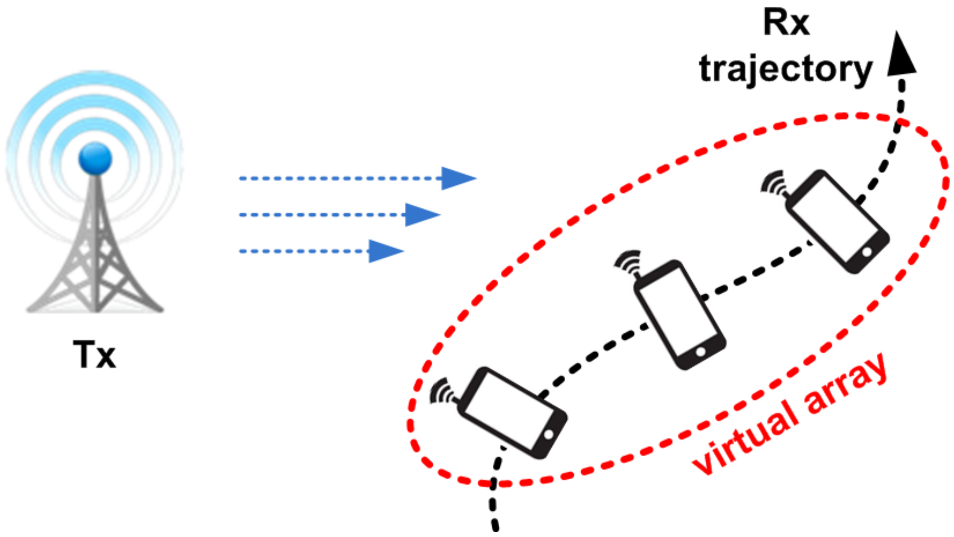
we want to measure the AOA of a Tx

- Tx sends multiple packets (e.g. synch' signal)

- Rx receives packets at multiple points along its trajectory

  $\Rightarrow$ each received packet can be seen as a « virtual » antenna element

  $\Rightarrow$ conventional MIMO AOA techniques



Outline

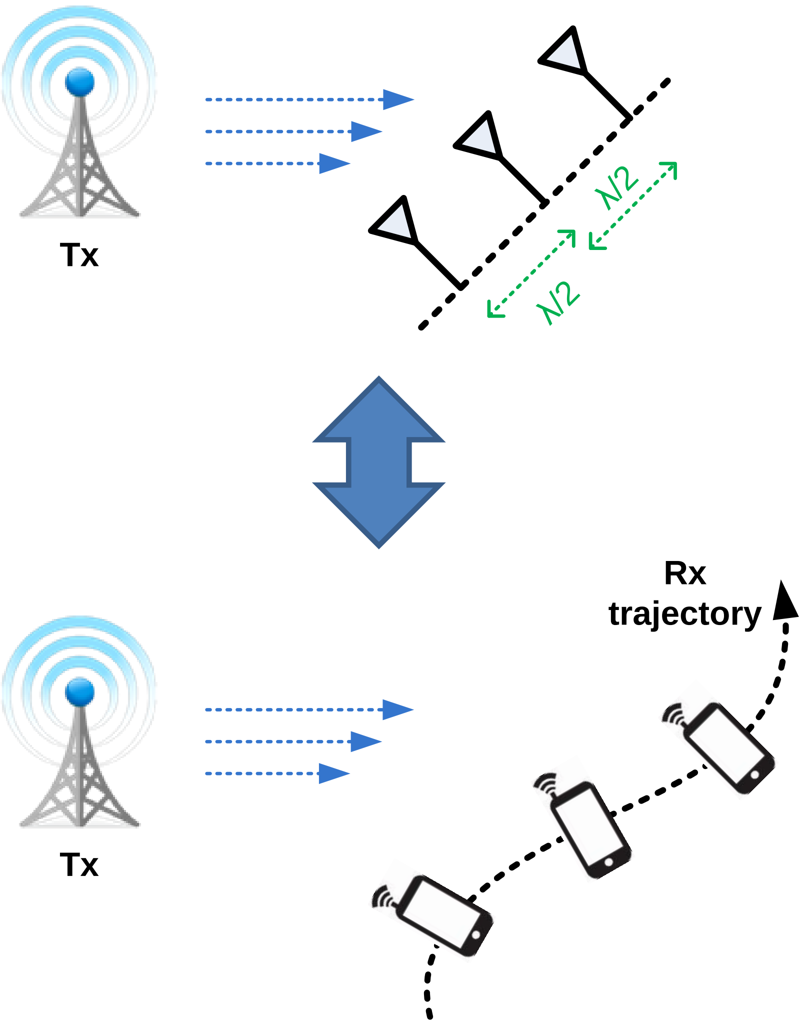
Virtual AOA estimation

- Method description
  - ⇒ Difference with conventional MIMO AOA
- Algorithms for LO offset and AOA estimation
- IMU sensor processing
- Implementation and results

Difference between V-AOA and MIMO-AOA

2 main differences in V-AOA case:

1) Position of « virtual antenna elements » depends on the movement of Rx



2) LO offset introduce phase rotation in received packets

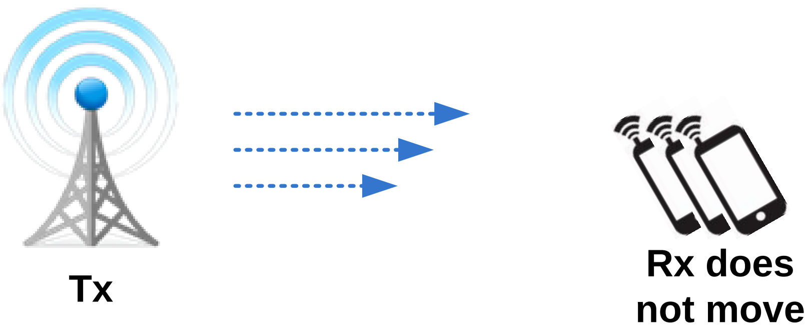
LO offset between Tx and Rx

… introduces a phase rotation in Rx packets

- LO offset between Tx and Rx

  $\Rightarrow$ net effect: frequency difference/offset ω_0 between Tx and Rx

- Receiver receives different packets (suppose no movement):
  - at time t_0: $r[m]$
  - at time t_1: $r[m]e^{j2\pi f_0(t_1-t_0)}$
  - at time t_2: $r[m]e^{j2\pi f_0(t_2-t_0)}$



AOA estimation: system description

System model

- Transmitter sends packet with known header
- Receiver correlates received baseband samples with (known) header

  $\Rightarrow$ Phase of peak of correlation function corresponds to the phase of the channel
- In a Line-of-Sight case (and periodic Tx), the angle is given by

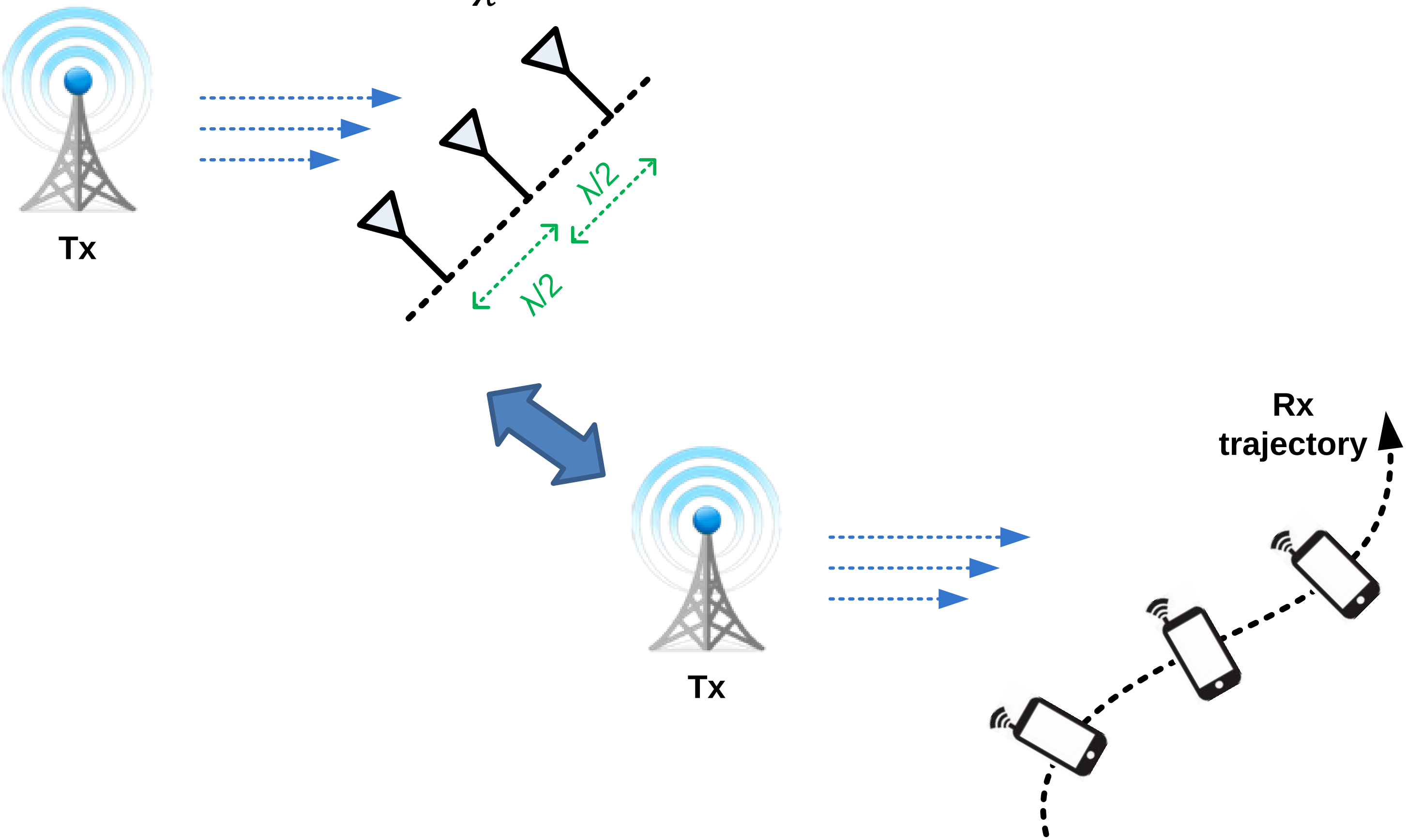
$$\varphi_n = \varphi_0 + 2\pi f_0 n T_0 + \frac{2\pi}{\lambda}(x_n \cos\theta + y_n \sin\theta)$$

t_n	time elapsed between packet 0 and n
x_n	change in x-coordinates between packet 0 and n
y_n	change in y-coordinates between packet 0 and n

AOA estimation: system description

Difference with conventional MIMO

$$\varphi_n = \varphi_0 + \frac{2\pi}{\lambda}(x_n \cos\theta + y_n \sin\theta)$$



$$\varphi_n = \varphi_0 + 2\pi f_0 n T_0 + \frac{2\pi}{\lambda}(x_n \cos\theta + y_n \sin\theta)$$

Outline

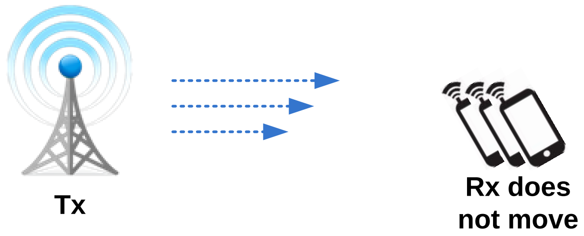
Virtual AOA estimation

- Method description
  - $\Rightarrow$ Difference with conventional MIMO AOA
- Algorithms for LO offset and AOA estimation
- IMU sensor processing
- Implementation and results

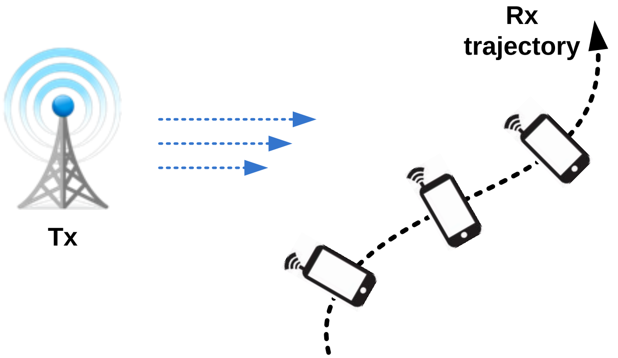
LO offset and angle estimation

Start- and-stop (SaS) approach

- Step 1: Receiver stands still
  - ⇒ Only LO frequency offset cause phase to change



- Step 2: Receiver starts moving
  - ⇒ LO frequency offset is compensated:
  - ⇒ Conventional MIMO estimation can be used (MUSIC, ESPRIT, …)



- Works if LO frequency offset does not change during movement phase
  - ⇒ Movement should be short
  - ⇒ Compatible with WSSUS assumption!

LO offset and angle estimation

Joint estimation

The signal model used in MUSIC can be augmented to accound for LO frequency offset

$$\mathbf{y}[m] = \mathbf{a}(f_0, \theta)x[m] + \mathbf{w}[m]$$

with

$$\mathbf{a}(f_0, \theta) = \begin{bmatrix} \exp\left(j\left[2\pi f_0 t_1 + \frac{2\pi}{\lambda}(x[1]\cos\theta + y[1]\sin\theta)\right]\right) \\ \exp\left(j\left[2\pi f_0 t_2 + \frac{2\pi}{\lambda}(x[2]\cos\theta + y[2]\sin\theta)\right]\right) \\ \vdots \\ \exp\left(j\left[2\pi f_0 t_N + \frac{2\pi}{\lambda}(x[N]\cos\theta + y[N]\sin\theta)\right]\right) \end{bmatrix}$$

$\Rightarrow$ MUSIC (or beamforming) can use this signal model and do joint search over f_0 and θ

Outline

Virtual AOA estimation

- Method description

  $\Rightarrow$ Difference with conventional MIMO AOA

- Algorithms for LO offset and AOA estimation

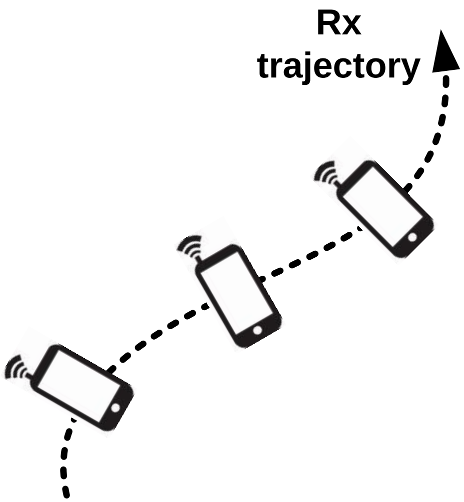
- IMU sensor processing

- Implementation and results

LO offset and angle estimation

Determining x_n and y_n

- Fraction of wavelength accuracy required
  - $\Rightarrow$ D-GPS insufficient!
- If antenna non-isotropic: orientation required
- Only relative position is required
- WSSUS assumption
  - $\Rightarrow$ Movement should be limited

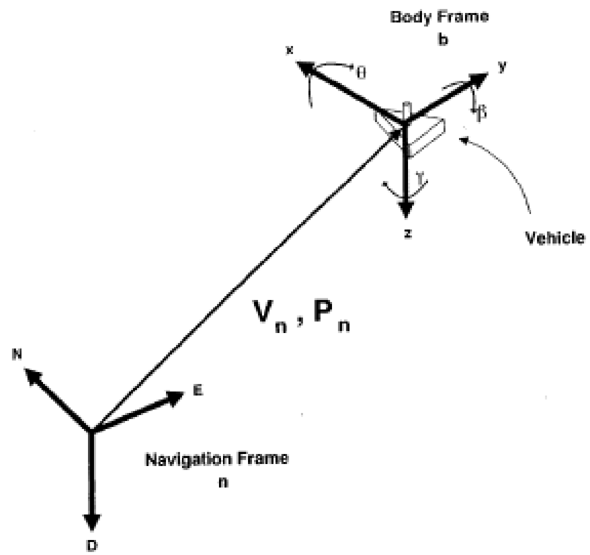


- We use a 3D-Inertial Measurement Unit (IMU)
  - $\Rightarrow$ Contains accelerometers and gyroscopes
  - $\Rightarrow$ Solution will drift from truth, but integration time is short due to WSSUS, so error will remain limited

Strap-down IMU

= IMU attached to vehicle

- accelerometers => measures ***acceleration*** along each axis
- gyroscope => measures ***angular speed*** around each axis

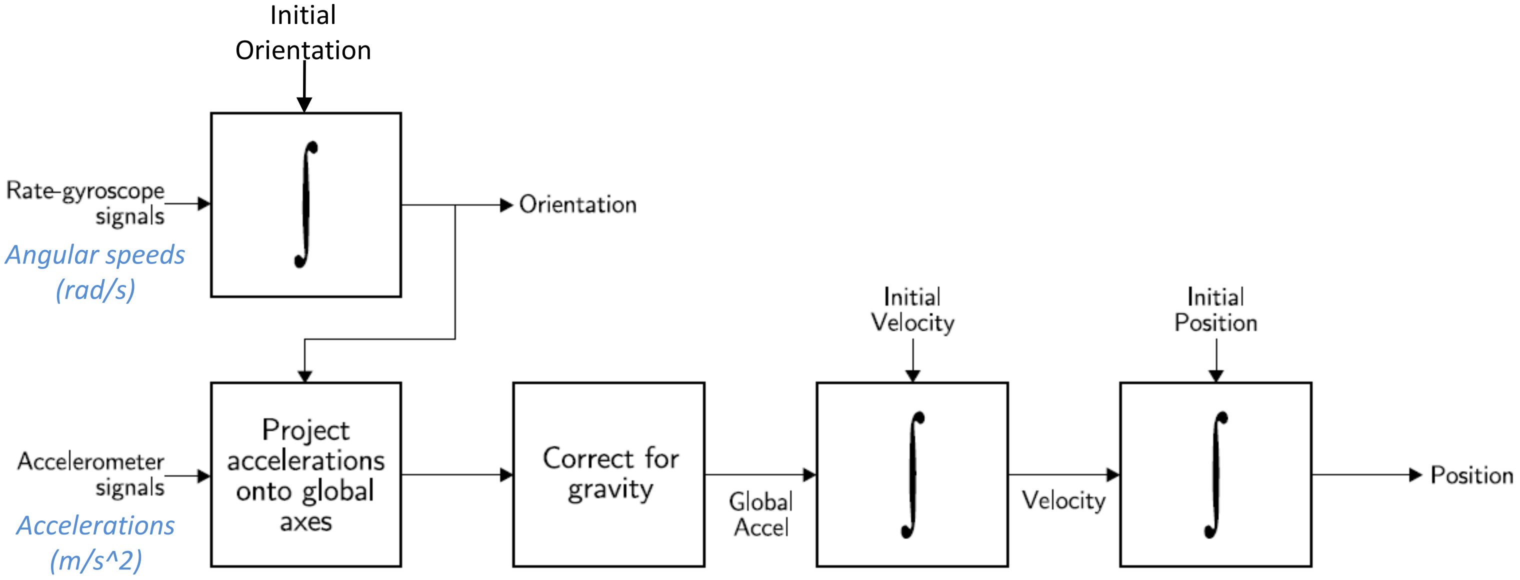


  – Measurements are done in ***body frame***, but positions needs to be known in ***navigation frame***
  – Note: gravitation of ~9.78 m/s^2 (along D-axis) is always measured by accelerometer(s)

IMU processing

Can be processed in EKF/UKF

- Initial position/orientation need to be known



- Problems:
  1) how to estimate initial orientation ? => use gravitation vector
  2) how to estimate IMU biases ? => calibration procedure
  3) Augment stability by using nonholonomic constraints

Outline

Virtual AOA estimation

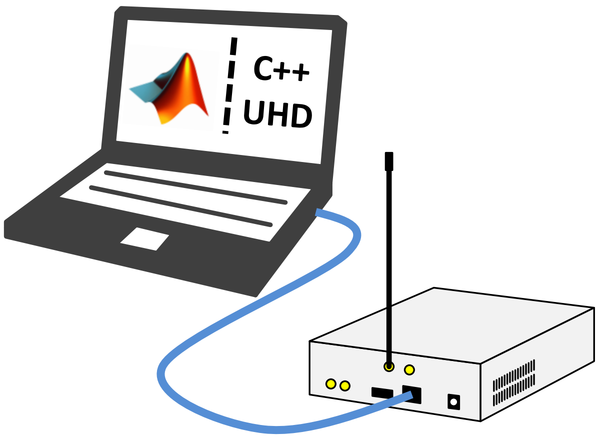
- Method description
  - ⇒ Difference with conventional MIMO AOA
- Algorithms for LO offset and AOA estimation
- IMU sensor processing
- Implementation and results

Implementation

Transmitter and receiver: USRP-N210

- Carrier frequency: 1 GHz
- Tx and Rx use GPSDO with OCXO LO (20 ppb accuracy)
- Tx sends 3G primary sequence
  - 128 samples long @ 1.8 MHz sample rate
  - Periodicity: 0.667 ms, but only one packet out of 15 considered

  $\Rightarrow T_0 = 10$ ms
- Rx sample rate = 3.6 MHz

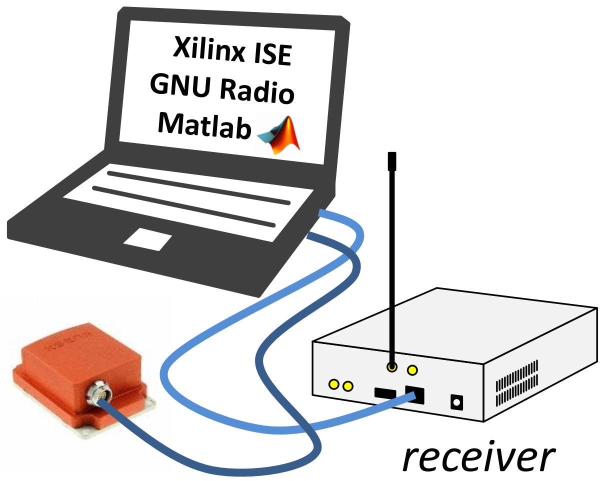


transmitter

Implementation

Receiver details

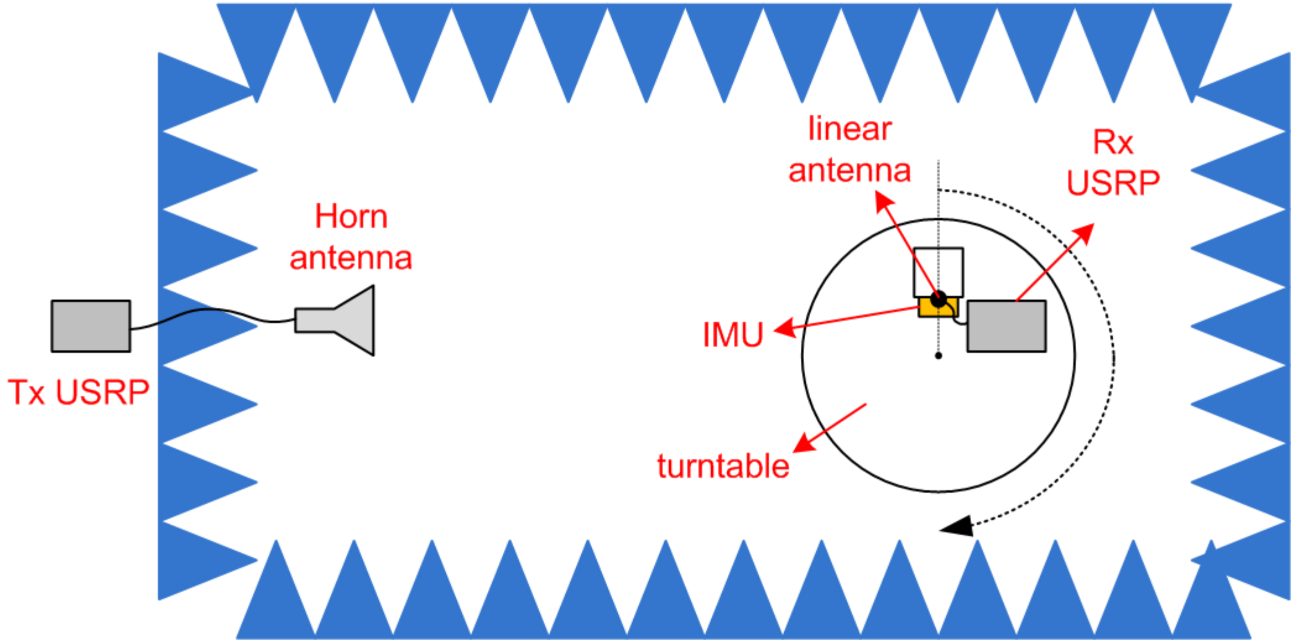
- Rx performs correlation in FPGA
  ⇒ Sends both correlation function (« peaks ») and BB samples to host
- Rx accumulates 3 peaks (host processor)
  ⇒ Increased SNR
- Peak detector in host processor
  ⇒ Phase of peak is written to output file
- IMU: XSens MTi-10 (automotive-grade)
- Parallel thread to read IMU data @ 200 Hz
  ⇒ IMU values written to output file



Experimental setup

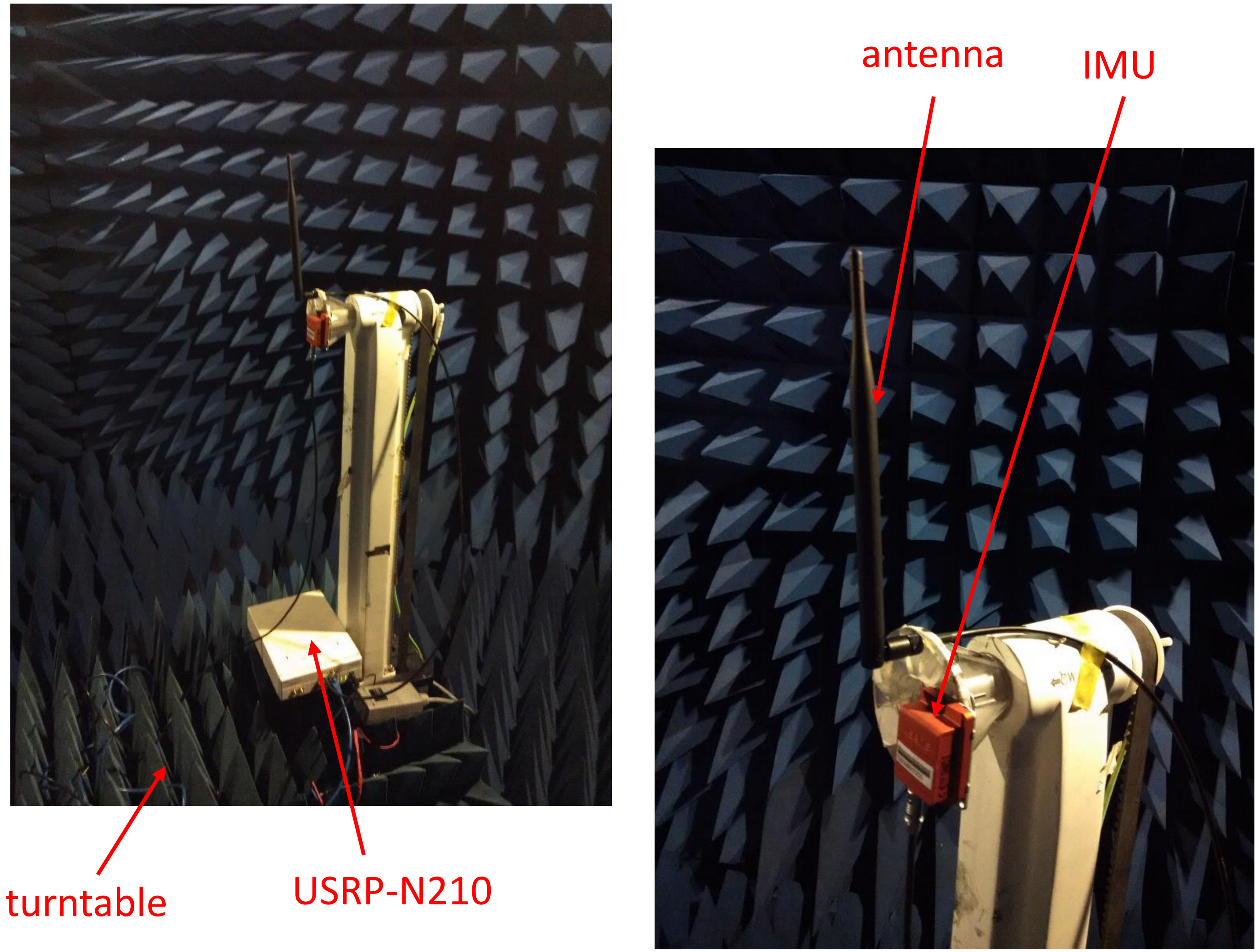
in anaechoic chamber => only LOS

- IMU z-axis placed parallel to vertical axis
  - ⇒ Error of few ° cannot be avoided!
- Turntable still for 30 s
  - ⇒ then turned by 180° (about 5 s)
  - ⇒ Radius of 30, 40 and 50 cm



Experimental setup

note the « vertical » IMU placement



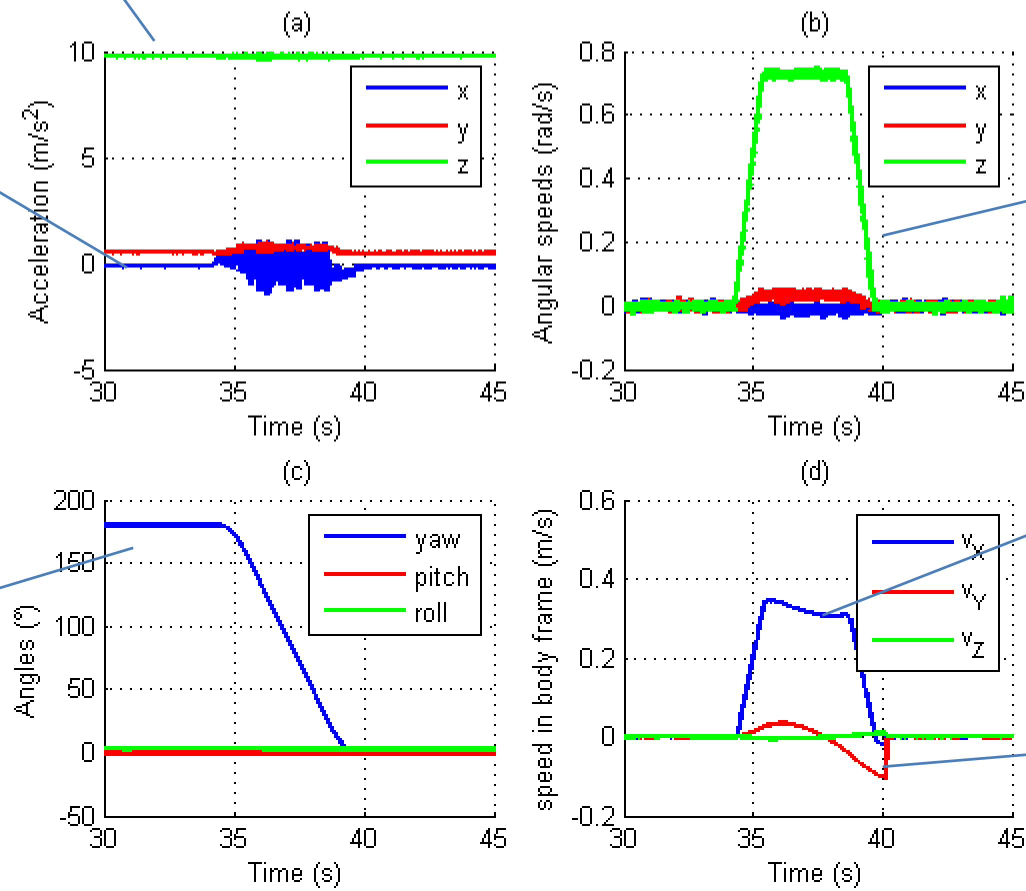
IMU processing

Initial orientation: (pitch,roll)=(-0.79°, 3.18°)

g along z-axis

Small acceleration and deceleration along x-axis

Rotation around z-axis



Yaw changes from 180° to 0°

Speeds mainly along x-axis

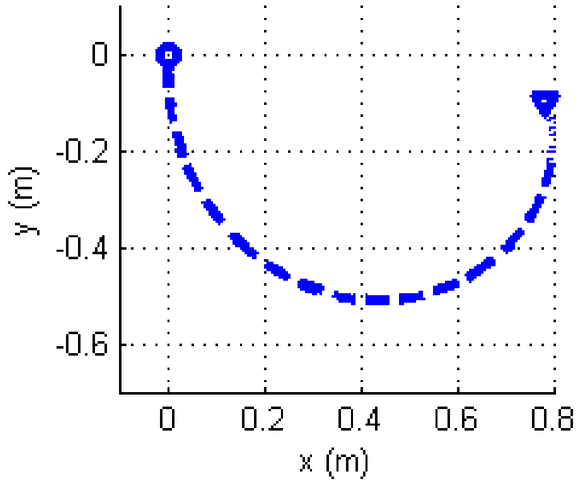
Speeds along y-axis:

- Centrifugal force
- Integration errors

IMU processing

Final estimated trajectory

- Estimated trajectory drifts off at the end of movement



- Room for improvement!
  - Introduce nonholonomic constraints (already done for standstill)
  - Improve bias estimation
  - Improve EKF/UKF parameters (requires to know process model accurately)

AOA estimation

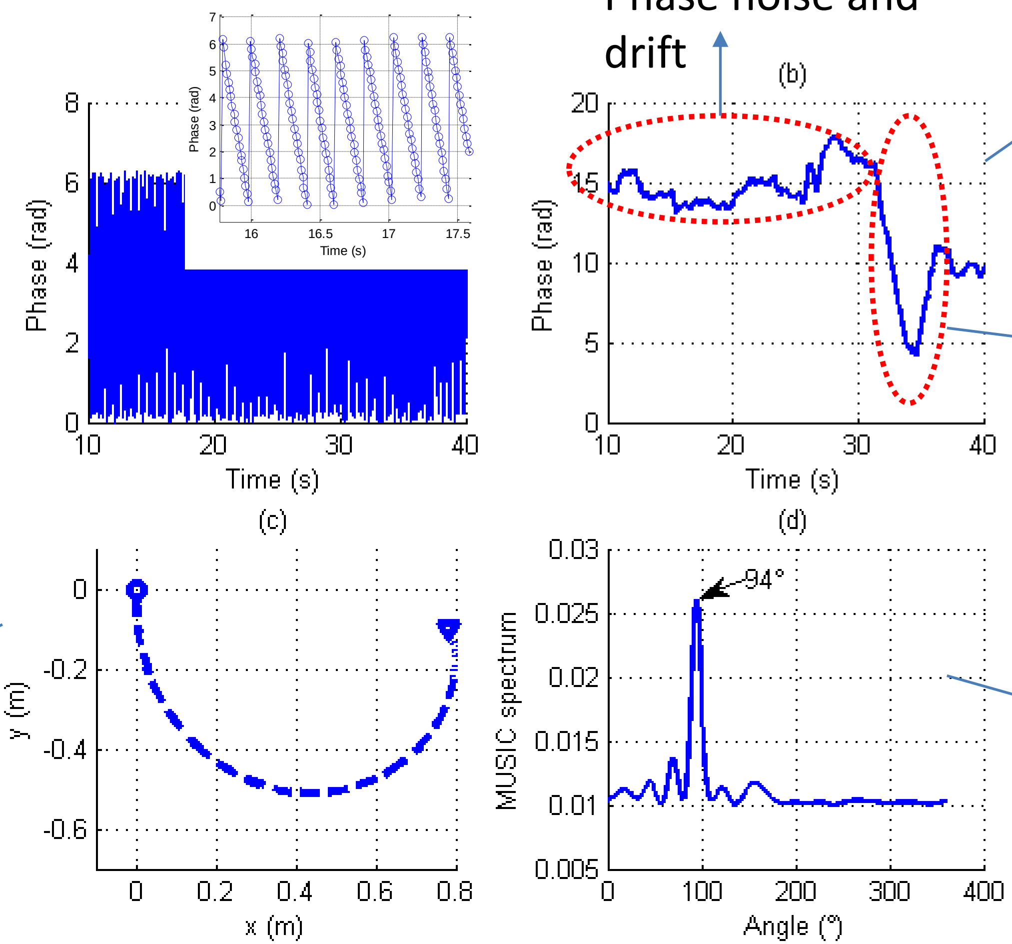
Stop-and-Start approach

Packet phases *before* LO offset compensation

Phase noise and drift

Packet phases *after* LO offset compensation

Phase change due to movement



Rx movement from IMU

MUSIC spectrum with peak close to 90°

AOA estimation

SaS approach: notes about MUSIC

$\Rightarrow$ AOA estimation error

- Zero-mean
- Standard deviation

Movement Radius	Stop-and-Start
30 cm	12.45°
40 cm	7.91°
50 cm	5.70°

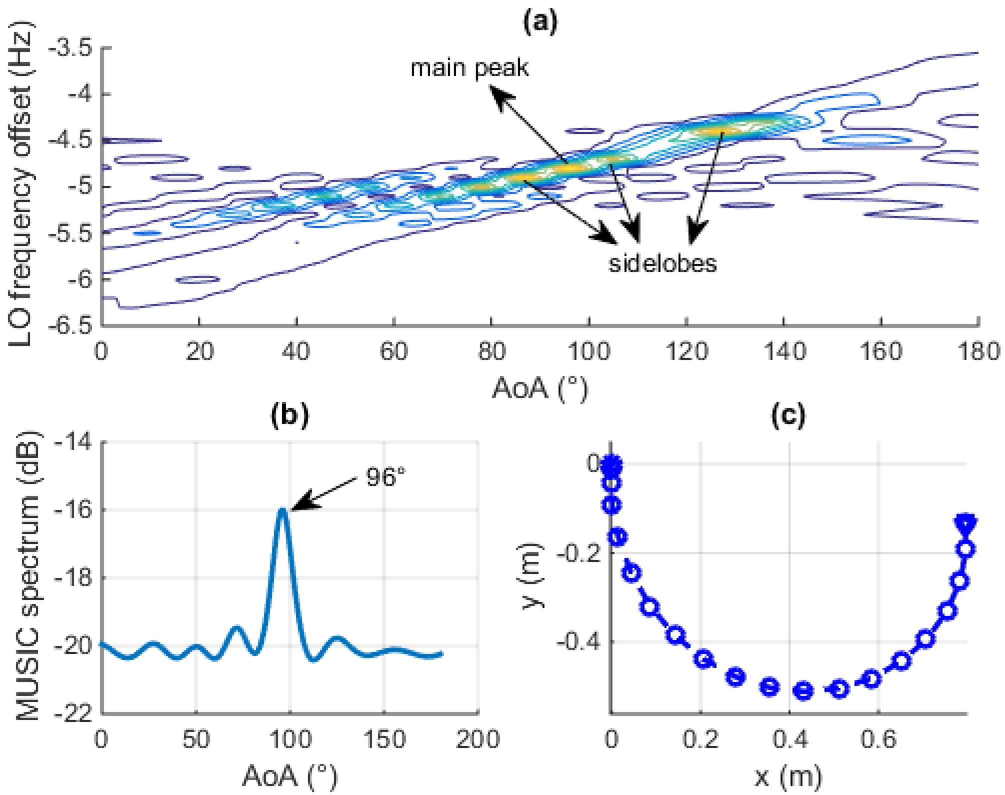
$\Rightarrow$ Larger (virtual) arrays have better accuracy

$\Rightarrow$ Consistent with conventional MIMO theory

AOA estimation

Joint estimator

- Augmented signal model
  - joint search over f_0 and θ



AOA estimation

Joint estimator

⇒ AOA estimation error

- Zero-mean
- Standard deviation

Movement Radius	Stop-and-Start	Joint estimation
30 cm	12.45°	29.78°
40 cm	7.91°	17.55°
50 cm	5.70°	6.45°

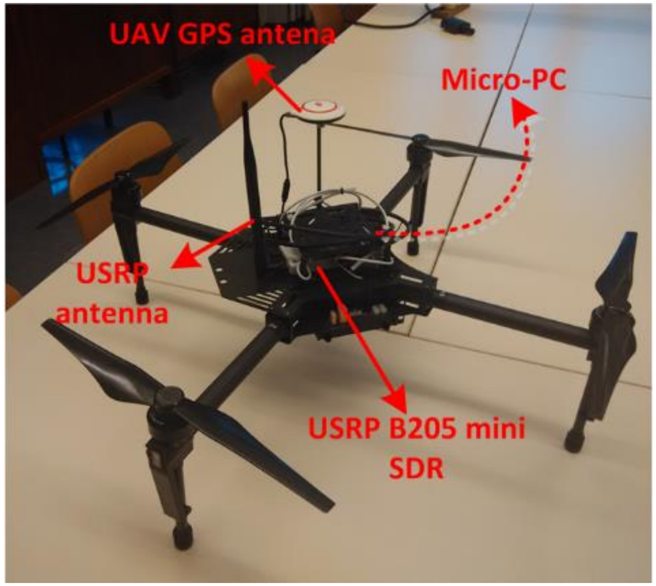
⇒ Larger (virtual) arrays have better accuracy

⇒ Performance of joint estimation worse than SaS approach, but more flexible !

E-310 implementation

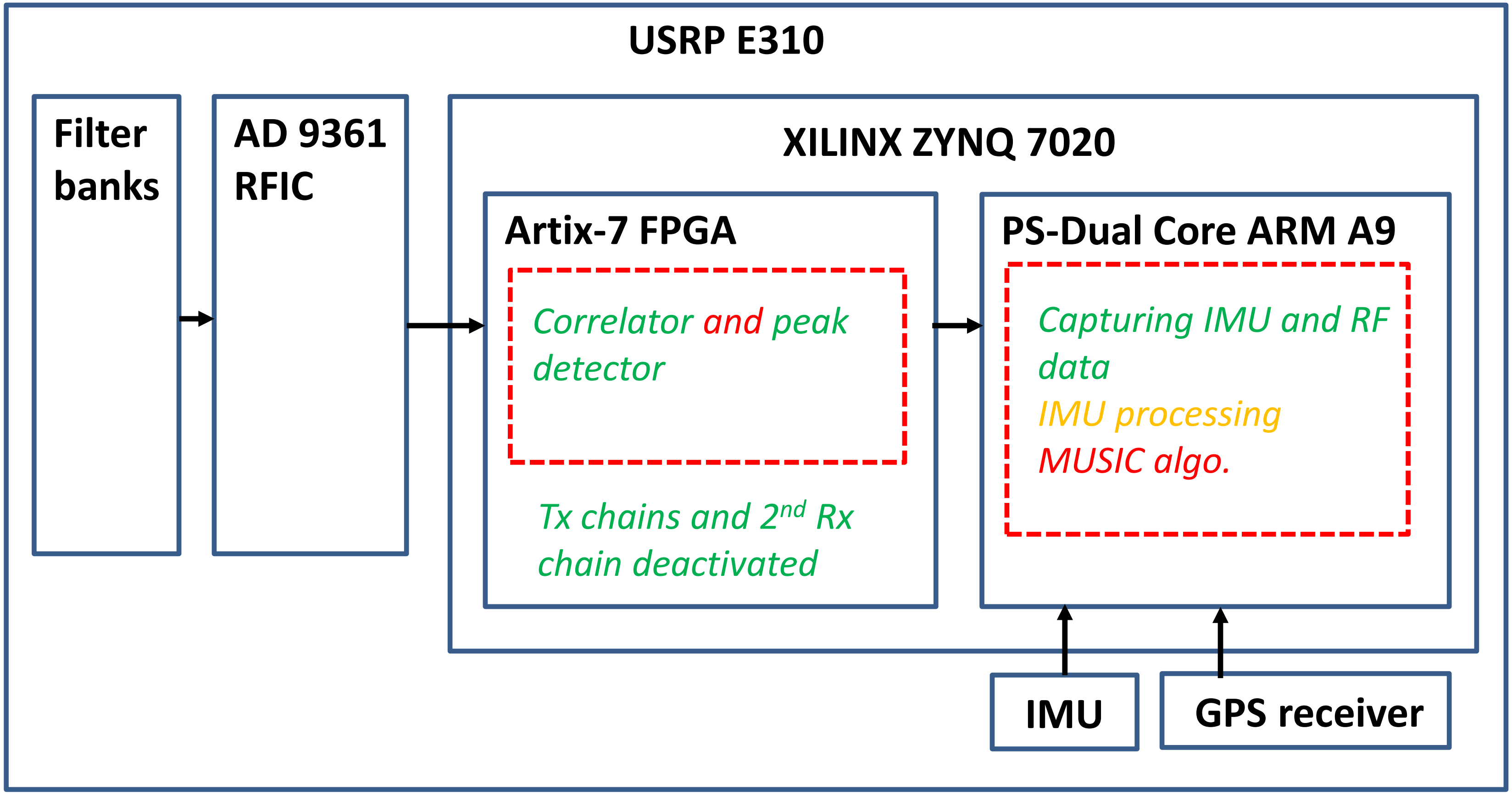
Why?

- Why not ?
- Use embedded IMU and SDR
- Test with low(er)-quality IMU and TCXO
- Possible to mount on (autonomous) vehicles



E-310 implementation

Architecture



E-310 implementation

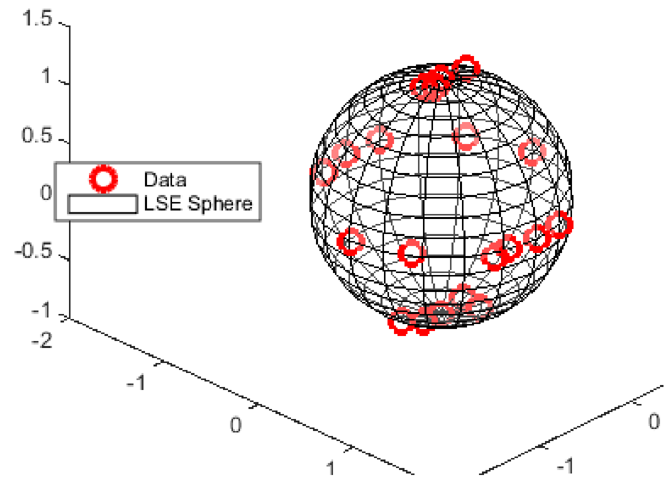
Determining IMU biases

- Gyroscope bias: can be measured at standstill
- Accelerometer bias:

$\Rightarrow$ IMU placed at all kind of orientations (but static!), N realizations

$\Rightarrow$ Find biases by solving least-squares problem (sphere-fit):

$$\begin{cases} (a_{x1} - b_{ax})^2 + \left(a_{y1} - b_{ay}\right)^2 + (a_{z1} - b_{az})^2 = g^2 \\ \vdots \\ (a_{xN} - b_{ax})^2 + \left(a_{yN} - b_{ay}\right)^2 + (a_{zN} - b_{az})^2 = g^2 \end{cases}$$



Example for IMU placed at a lot of orientations (more or less along main axes)

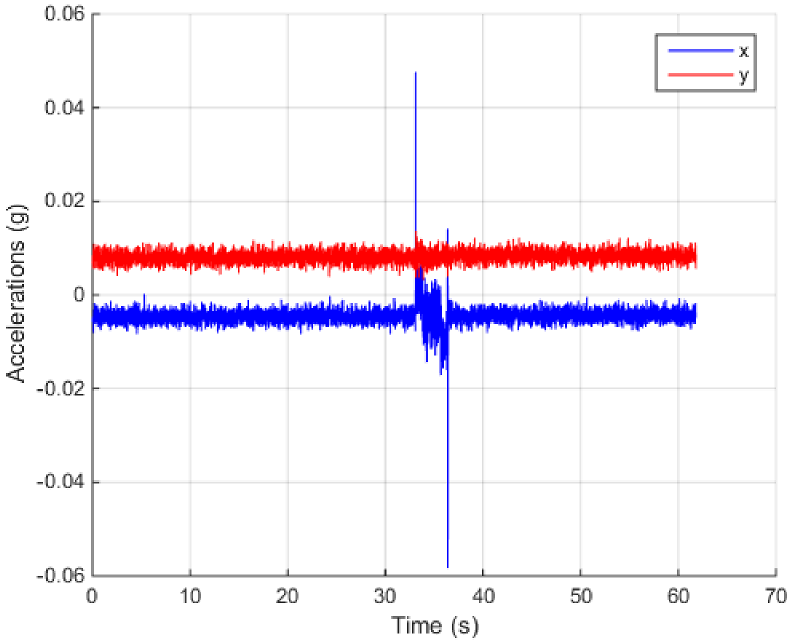
E-310 implementation

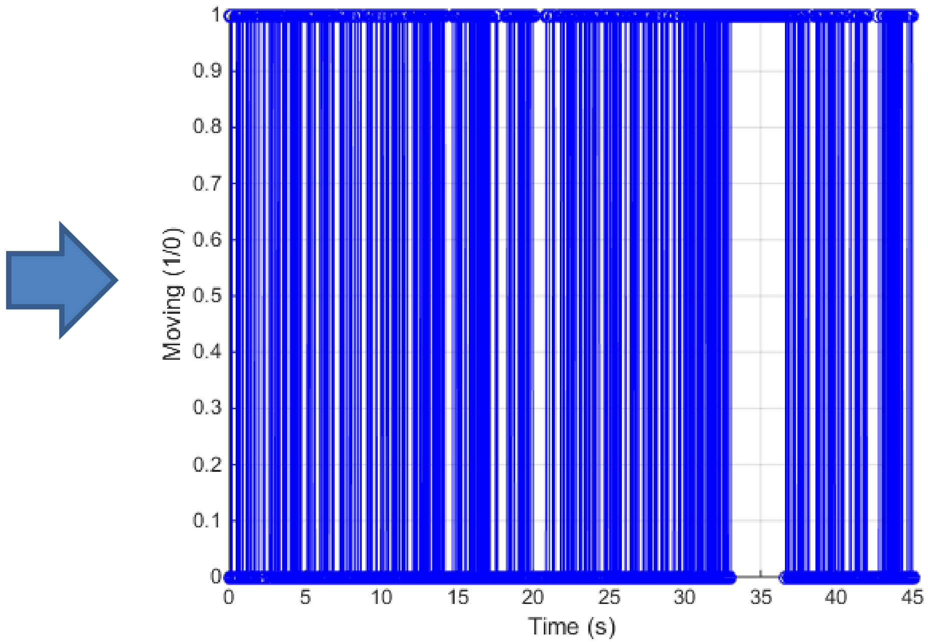
IMU processing

- Through use of an unscented Kalman filter
- Use nonholonomic constraints when standing still

  $\Rightarrow v_x = v_y = v_z = 0$

- Standstill is estimated by looking at accelerometers

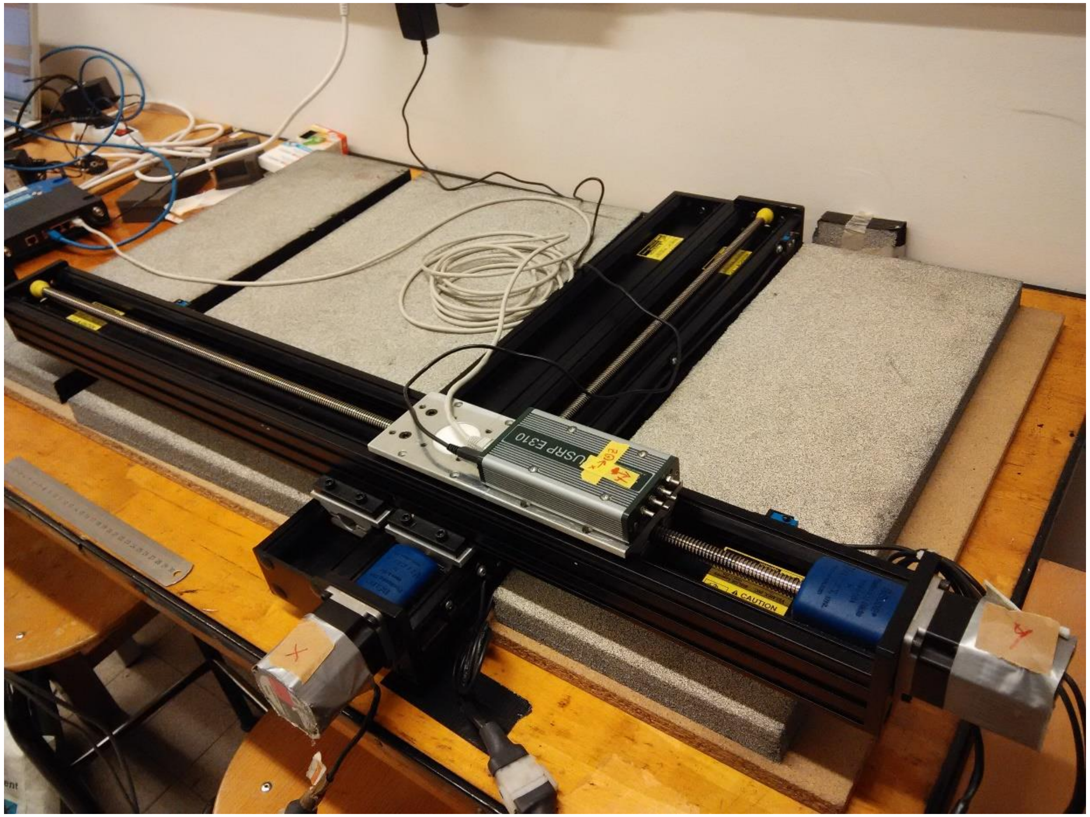




E-310 implementation

IMU analysis results

- MPU-9150 IMU in controlled experiment



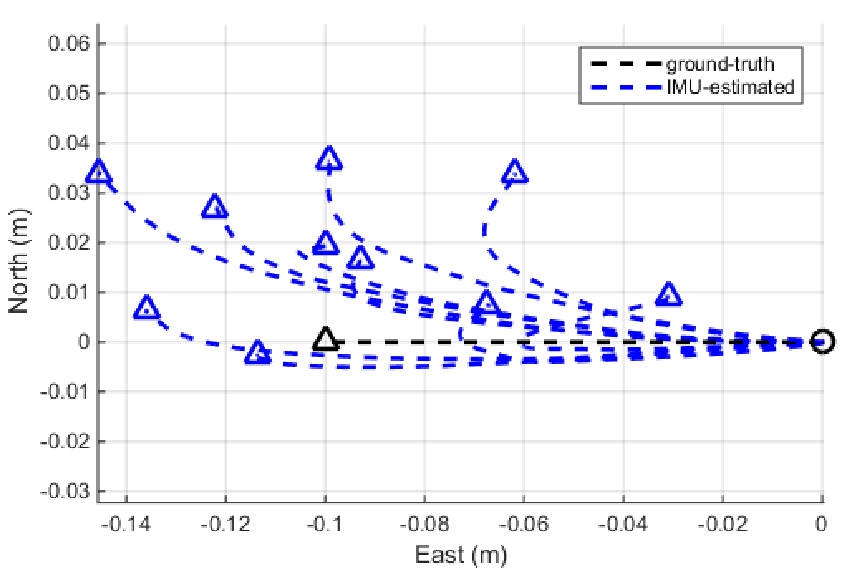
E-310 implementation

IMU analysis results

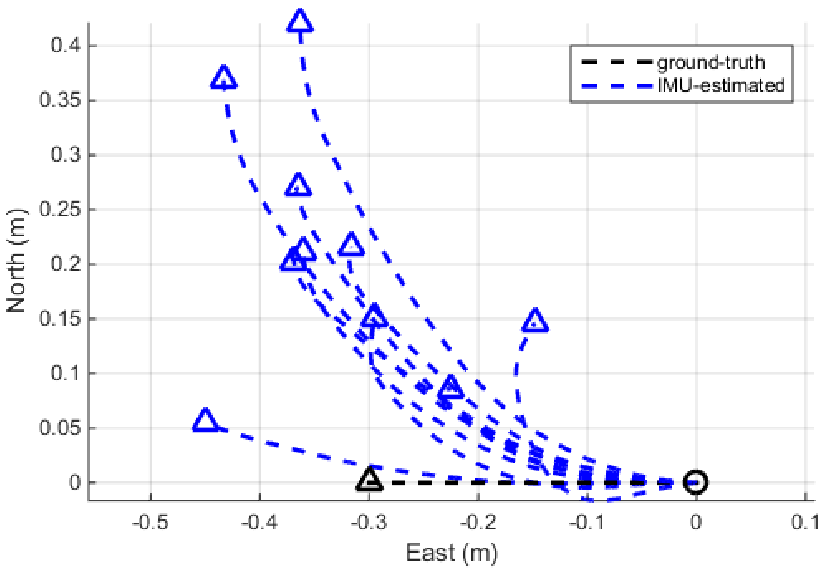
- Things go south pretty quickly …
- Movement should remain short !!!

$\Rightarrow$ Anyway required for WSSUS assumption

Movement time ~ 4 seconds



Movement time ~ 10 seconds



V-AOA estimation

Conclusion and to-do List

Even with single-antenna transceivers, it is possible to do localization!

$\Rightarrow$ Integration with other sensors (IMU, GPS, …)

To-do list …

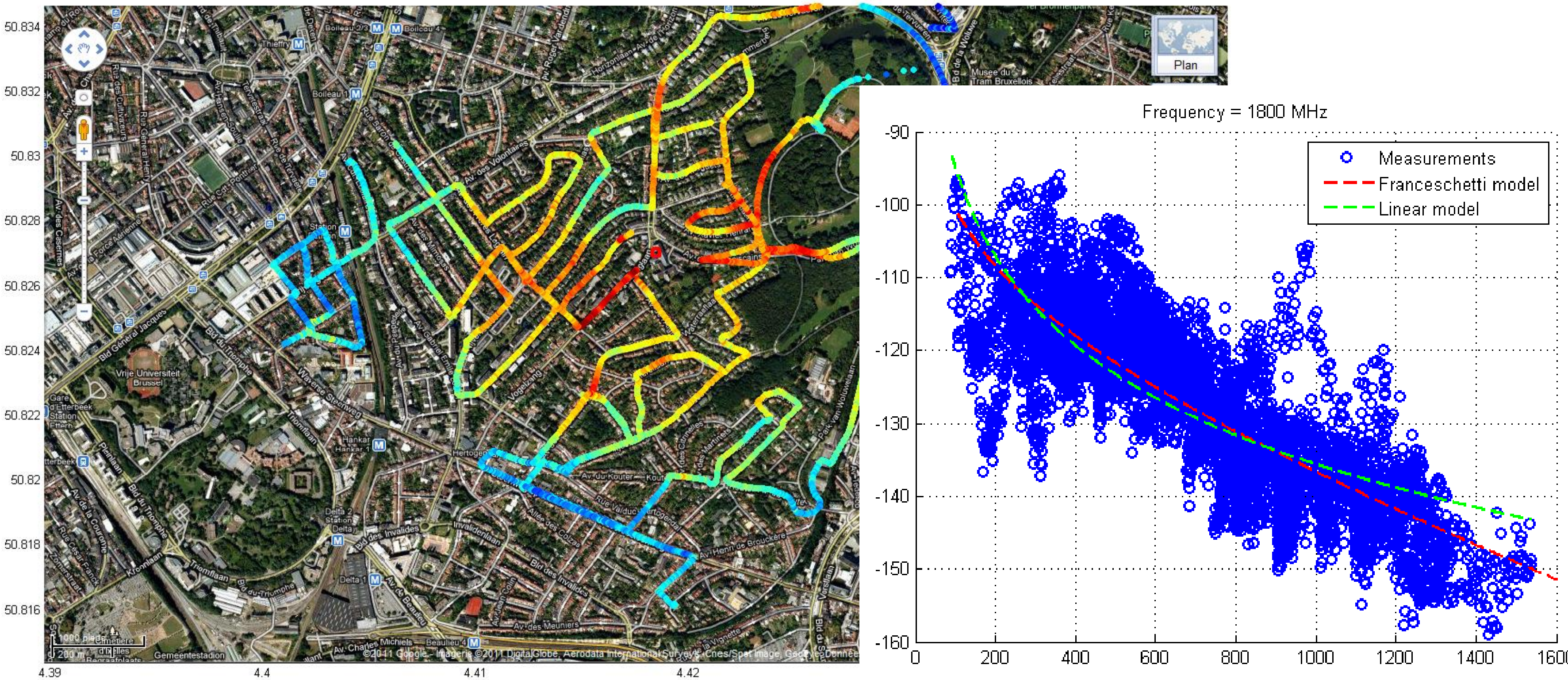
- wrap up E310 implementation
- Integrate IMU navigation uncertainty from EKF/UKF with the actual AOA estimation
- Multipath environment
- Robotic application: what are the « best » trajectories for accurate estimation ?

Thank you !

Localization in cellular wireless networks

… does not work very well

- Received signal strength (RSS)
  - ⇒ requires path loss models (which one?)
  - ⇒ **fading** around path loss curve can be huge!

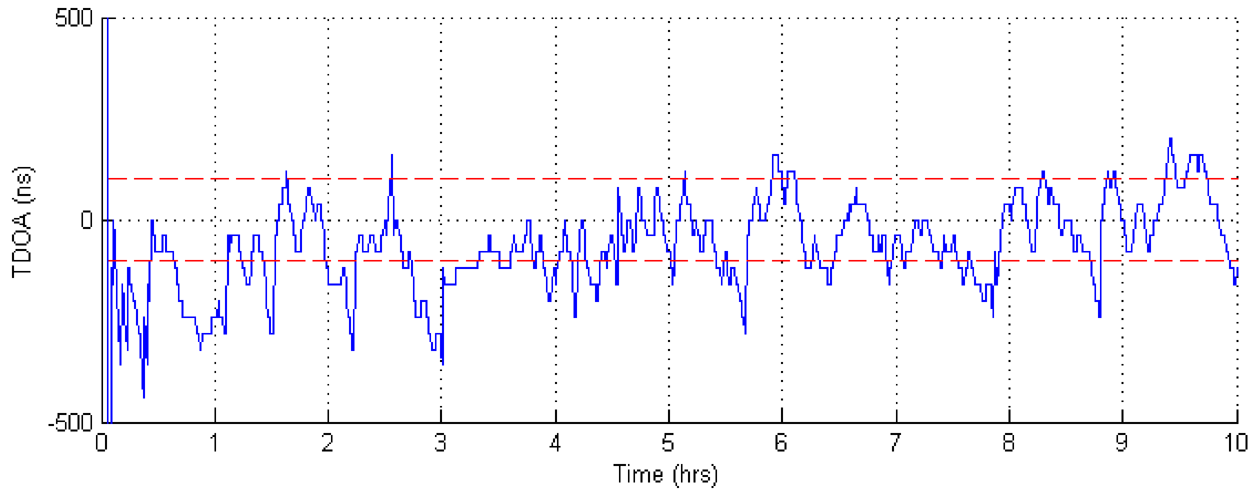


Localization in cellular wireless networks

… does not work very well

- Time-of-Arrival / Time-Difference-of-Arrival (TOA/TDOA)
  - ⇒ works better at high bandwidths
  - ⇒ requires nodes to be **synchronized** (down to ns accuracy)
  - ⇒ requires stringent control of hardware delays
  - ⇒ sensitive to multipath

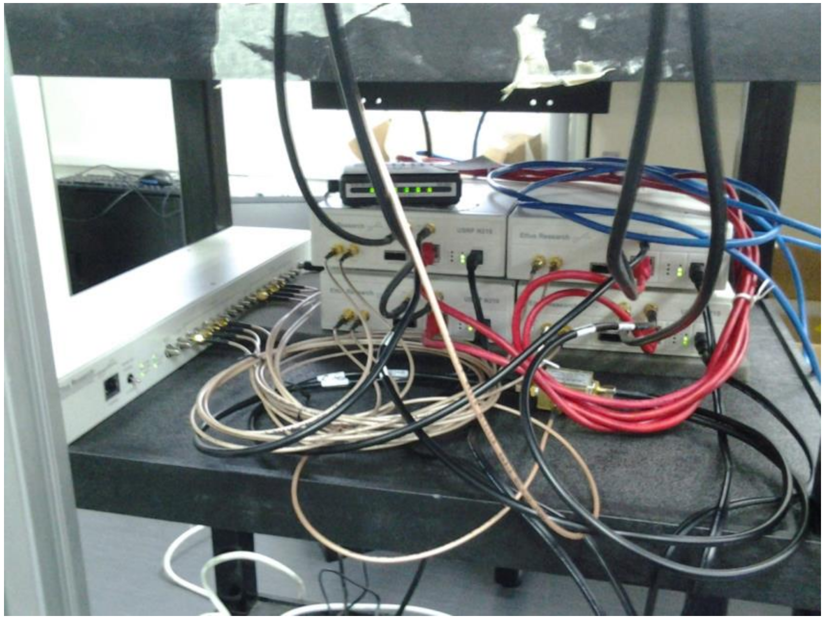
GPS synchronization: measured TDOA (ground truth = 0 ns)



Localization in cellular wireless networks

… does not work very well

- Angle-of-Arrival (AOA)
  - ⇒ requires multi-antenna array
  - ⇒ expensive and **large form factors**!
  - ⇒ sensitive to multipath



AOA estimation: system description

Received packets angle

$$\varphi_n = \varphi_{n-1} + \underbrace{2\pi f_0 \Delta t_n}_{\text{Freq. offset}} + \underbrace{\vec{\beta} \cdot \Delta\vec{r}_n}_{\text{Rx movement}} + \underbrace{2\pi\nu\Delta t_n}_{\text{Doppler shift}}$$

f_0	frequency offset between Tx and Rx
Δt_n	time between packets n-1 and n
$\vec{\beta}$	wave vector
$\Delta\vec{r}_n$	displacement vector between packets n-1 and n
ν	Doppler shift between Tx and Rx

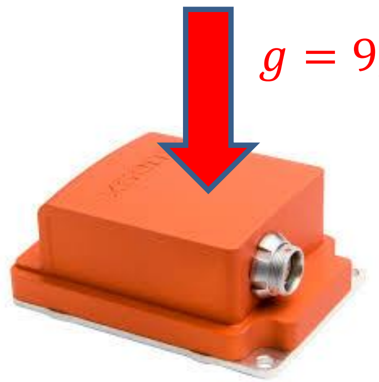
$\Rightarrow$ small in considered cases, can be ignored

IMU processing

Initial orientation determination

- At standstill, only gravitation vector is measured [1]
  - $\Rightarrow$ Gravitation vector is always along D(own)-axis
  - $\Rightarrow$ Can be used to determine pitch/roll

$g = 9{,}78\ m/s^2$

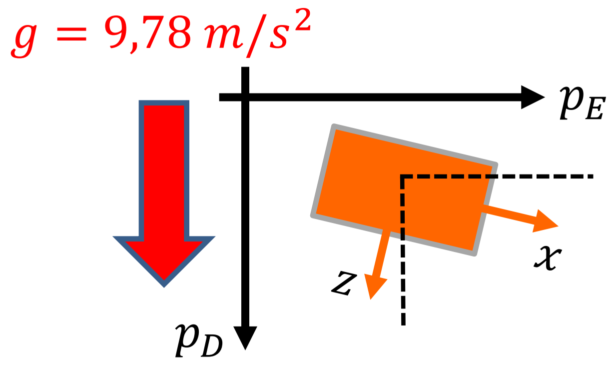


[1] S.O.H. Madgwick, A.J.L. Harrison, and R. Vaidyanathan, “Estimation of IMU and MARG orientation using a gradient descent algorithm,” in Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on, June 2011, pp. 1–7.

IMU processing

Why is initial orientation so important ?

- Imagine an IMU with an inclination of 0.5°



$\Rightarrow$ Inclination of 0.5° leads to an acceleration along the x-axis of

$$g \cdot \sin\theta = 0{,}085\ \mathrm{m/s}^2$$

$\Rightarrow$ double-integrate this: after 5 s, error in distance is > 1 m

$\Rightarrow$ Not exactly sub-wavelength accuracy…