Motion Compensation in IMUs for Attitude Estimation

Julius-Maximilians-University Würzburg (JMU)

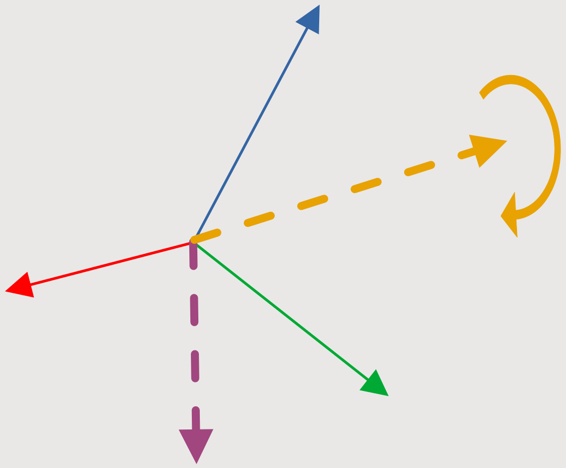
Fabian Arzberger*

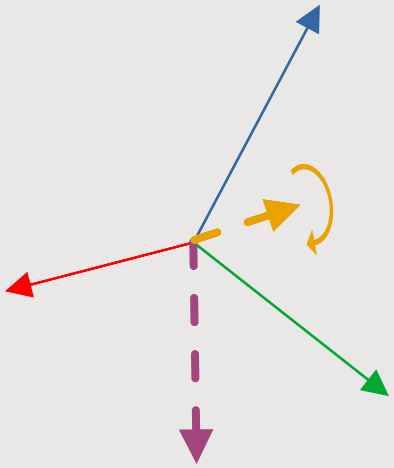
*fabian.arzberger@uni-wuerzburg.de

Prof. Dr. Andreas Nüchter

Inertial Sensing in Mobile Systems

- Inertial sensing using accelerometer and gyroscope
- Attitude estimation: Use gravity vector (2 DoF) + angular velocity (3 DoF)
- Common approach: apply weighting between the two, e.g.:
  Complementary Filter, **“Madgwick”**, **“Mahony”**, **“Kalman”**, etc...



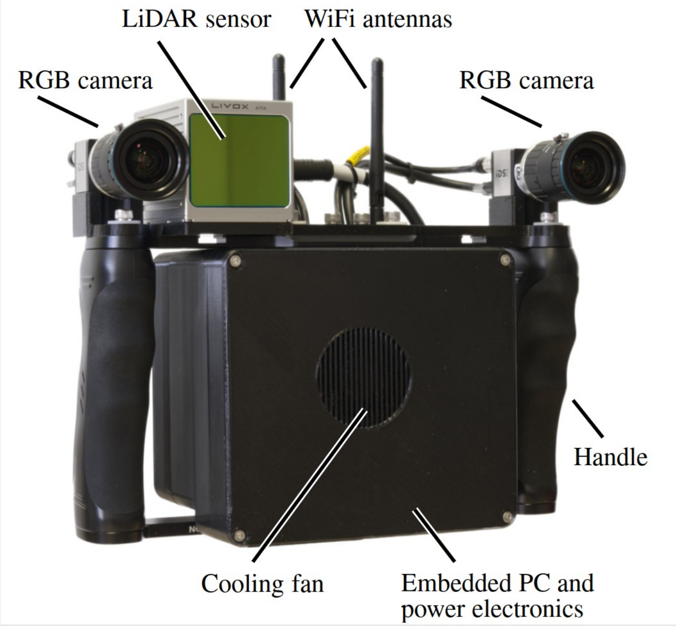


Inertial Sensing in Mobile Systems

- Where should we place Inertial Measurement Units (IMUs)?

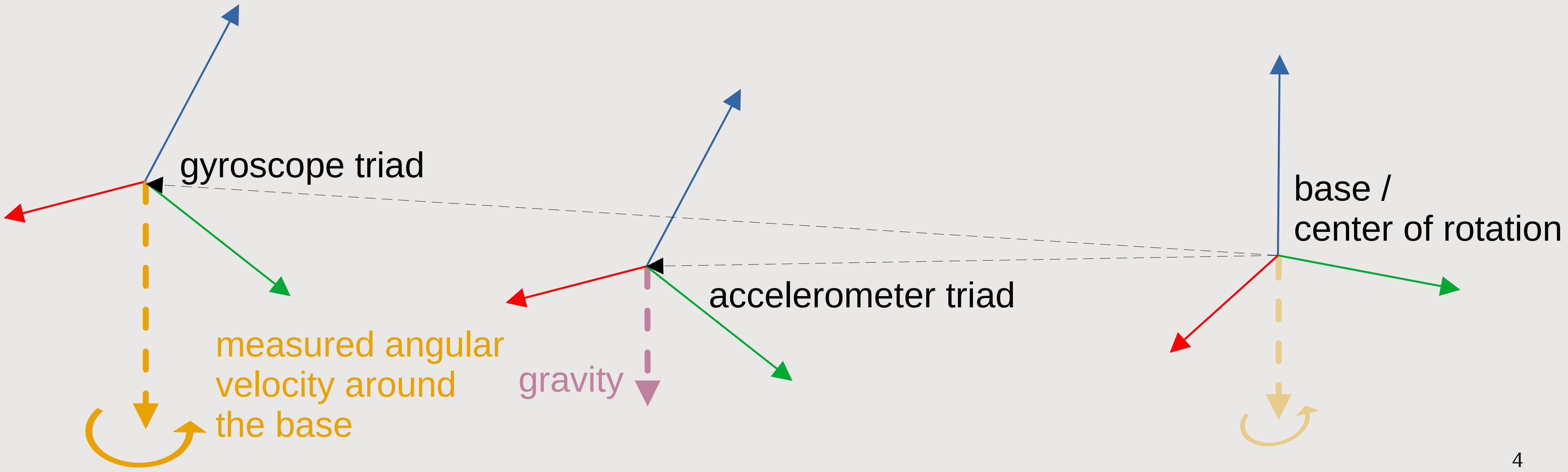




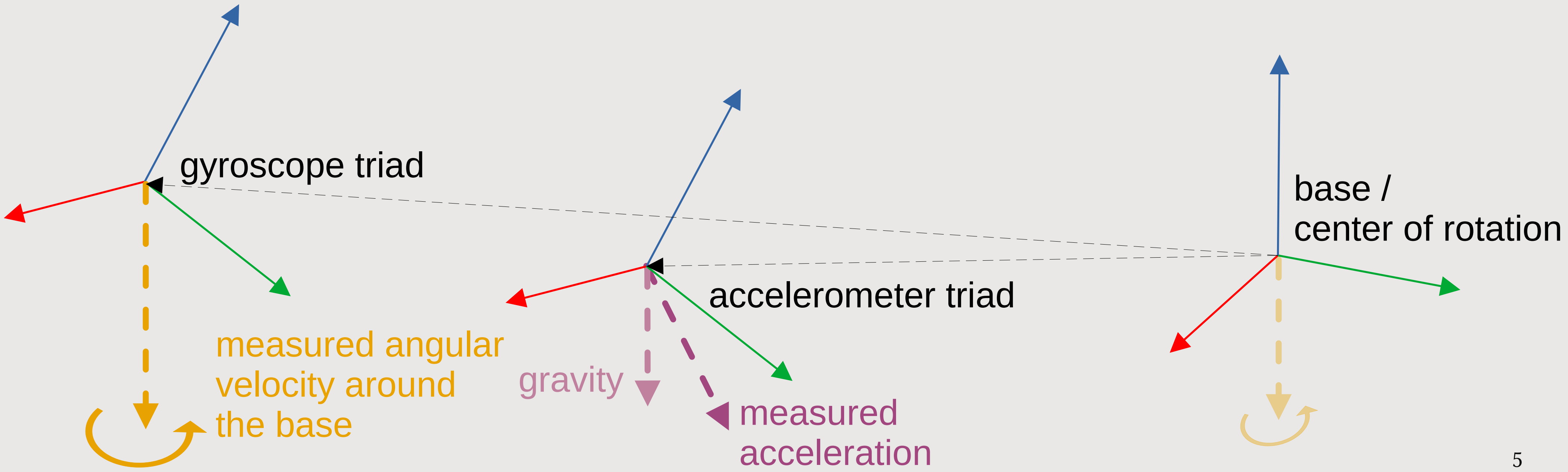
Inertial Sensing in Mobile Systems

- Common center of rotation might exist → usually called the „base“
- Mount IMU as close to the base as possible. But...
  - Design constraints could make it impossible
  - Location of the accelerometer- and gyroscope-traids on chip might be unknown



Motion Acceleration Effect in IMUs

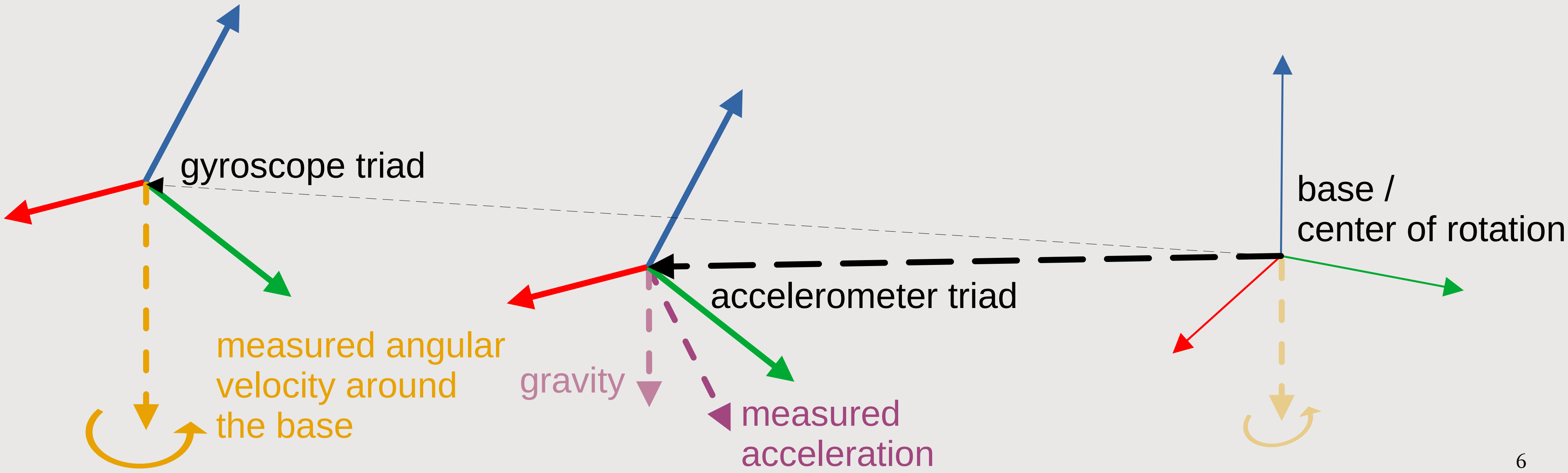
- Problem: During rotation, measured acceleration is not exactly gravity
- Every state-of-the-art attitude filter relies on the direction of gravity
- If we recover the correct gravity during rotation, we improve any filter



Motion Acceleration Effect in IMUs

- We need:
  - Extrinsic base-to-accelerometer vector, aka. „lever-arm“
  - Intrinsic calibration between gyroscope- and accelerometer-triads

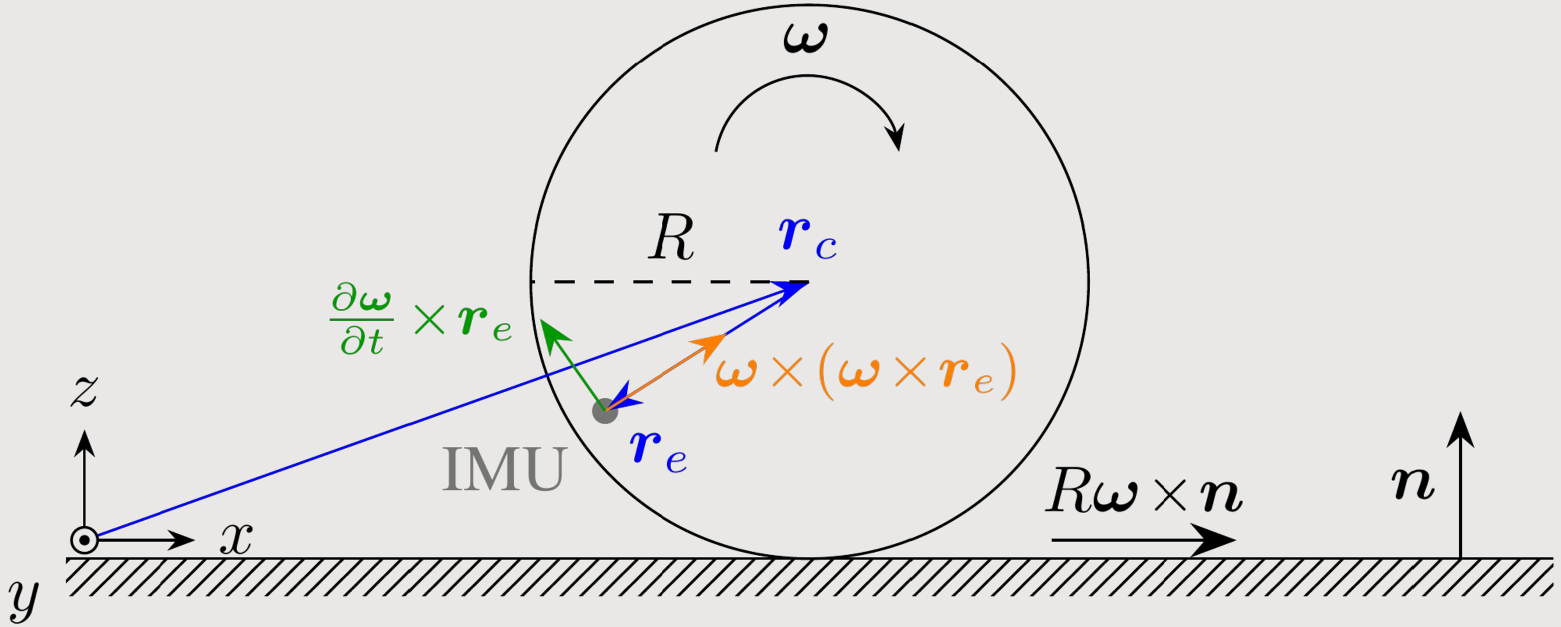
- Non-uniform circular motion describes tangential- and centripetal-accelerations



Motion Acceleration Effect in IMUs

- We need:
  - Extrinsic base-to-accelerometer vector, aka. „lever-arm“ r_e
  - Intrinsic calibration between gyroscope- and accelerometer-triads

- Non-uniform circular motion describes tangential- and centripetal-accelerations



Intrinsic and Extrinsic IMU Calibration

I. Data collection

Initial static period:
Do not move the system
~ 60 s

Rotate the system
around the base
~ 5 s

Put the system down in a
new attitude, do not move
~ 5 s

Repeat ~50x

Accelerometer and gyroscope data

II. Calibration

Calibrate intrinsic parameters bias $\boldsymbol{b}$, scale $\boldsymbol{S}$, and misalignment $\boldsymbol{M}$, s.t.

$$\boldsymbol{a} = \boldsymbol{M}_a \cdot \boldsymbol{S}_a \cdot (\hat{\boldsymbol{a}} - \boldsymbol{b}_a)$$

$$\boldsymbol{\omega} = \boldsymbol{M}_\omega \cdot \boldsymbol{S}_\omega \cdot (\hat{\boldsymbol{\omega}} - \boldsymbol{b}_\omega)$$

Calibrate extrinsic lever-arm $\boldsymbol{r}_e$, s.t.

$$\boldsymbol{a}_g = \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_e) + \frac{\delta \boldsymbol{\omega}}{\delta t} \times \boldsymbol{r}_e - \boldsymbol{a}$$

Bias Scale Misalignment Lever arm

Intrinsic and Extrinsic IMU Calibration

I. Data collection

Initial static period:
Do not move the system
~ 60 s

↓

Rotate the system
around the base
~ 5 s

↓

Put the system down in a
new attitude, do not move
~ 5 s

Repeat ~50x

↓

Accelerometer and gyroscope data

II. Calibration

Calibrate intrinsic parameters bias $\boldsymbol{b}$, scale $\boldsymbol{S}$, and misalignment $\boldsymbol{M}$, s.t.

$$\boldsymbol{a} = \boldsymbol{M}_a \cdot \boldsymbol{S}_a \cdot (\hat{\boldsymbol{a}} - \boldsymbol{b}_a)$$

$$\boldsymbol{\omega} = \boldsymbol{M}_\omega \cdot \boldsymbol{S}_\omega \cdot (\hat{\boldsymbol{\omega}} - \boldsymbol{b}_\omega)$$

↓

Calibrate extrinsic lever-arm $\boldsymbol{r}_e$, s.t.

$$\boldsymbol{a}_g = \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_e) + \frac{\delta \boldsymbol{\omega}}{\delta t} \times \boldsymbol{r}_e - \boldsymbol{a}$$

↓

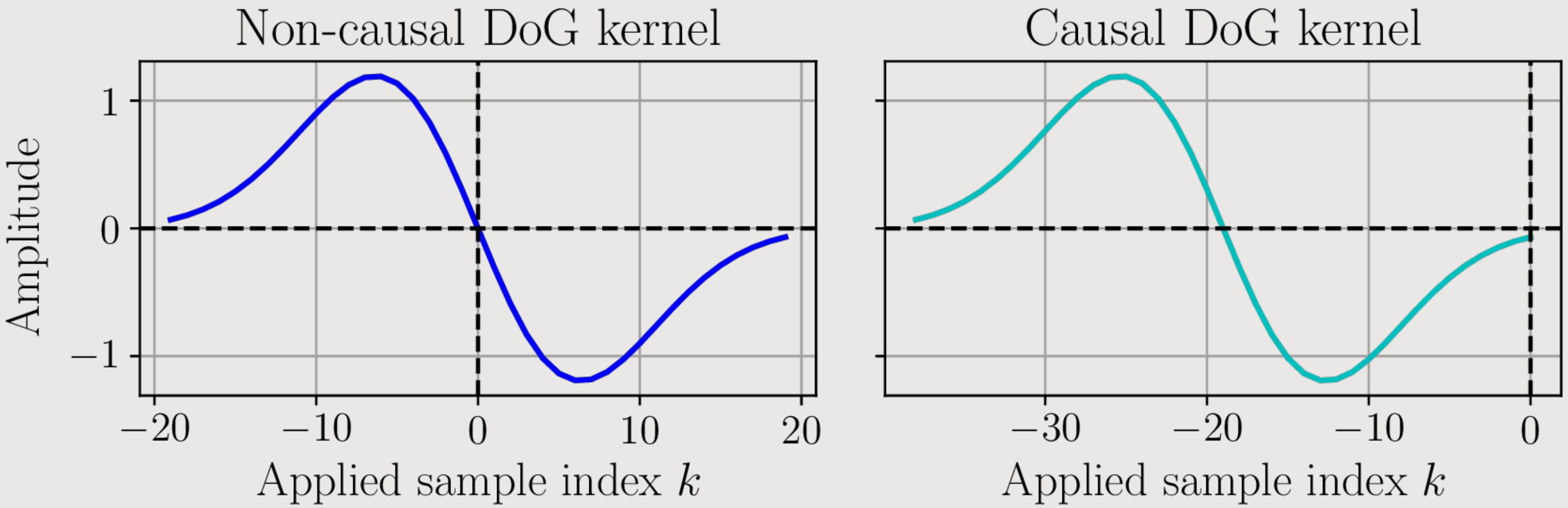
Bias | Scale | Misalignment | Lever arm

Numerical Derivative with Gaussian Kernels

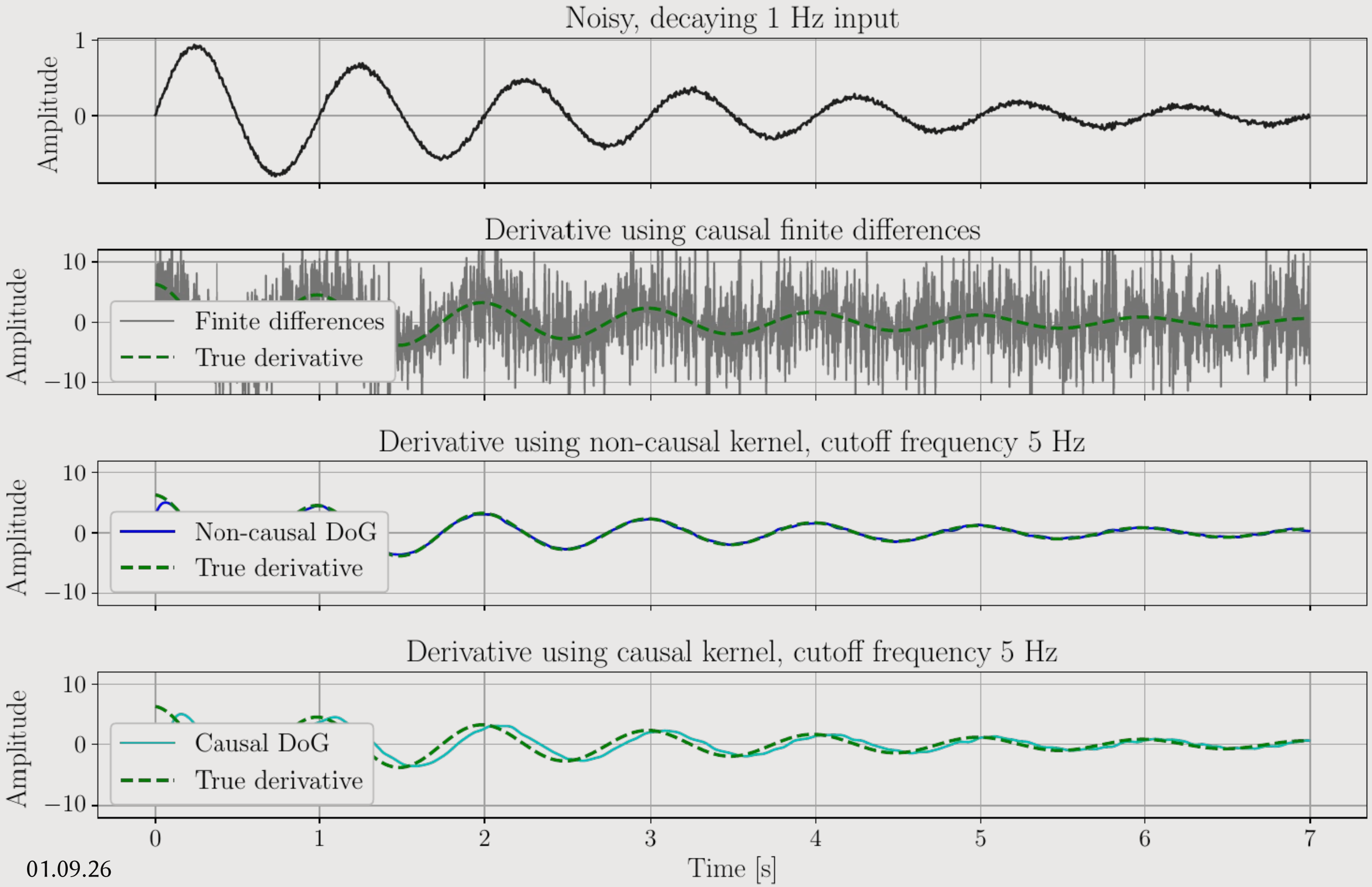
→ Apply convolution ($h * \boldsymbol{\omega}$) of the signal $\boldsymbol{\omega}$ with a Derivative of Gaussian kernel h:

$$\boxed{h[k] = -\frac{t_k}{\sigma^3\sqrt{2\pi}}\exp\left(-\frac{t_k^2}{2\sigma^2}\right)} \qquad t_k = k \cdot \Delta t$$

$$\frac{\delta\boldsymbol{\omega}}{\delta t}[n] = \sum_{k=-K}^{K} h[k] \cdot \boldsymbol{\omega}[n-k] \qquad \frac{\delta\boldsymbol{\omega}}{\delta t}[n] = \sum_{k=-K}^{K} h[k] \cdot \boldsymbol{\omega}[n-k-K]$$



Numerical Derivative with Gaussian Kernels



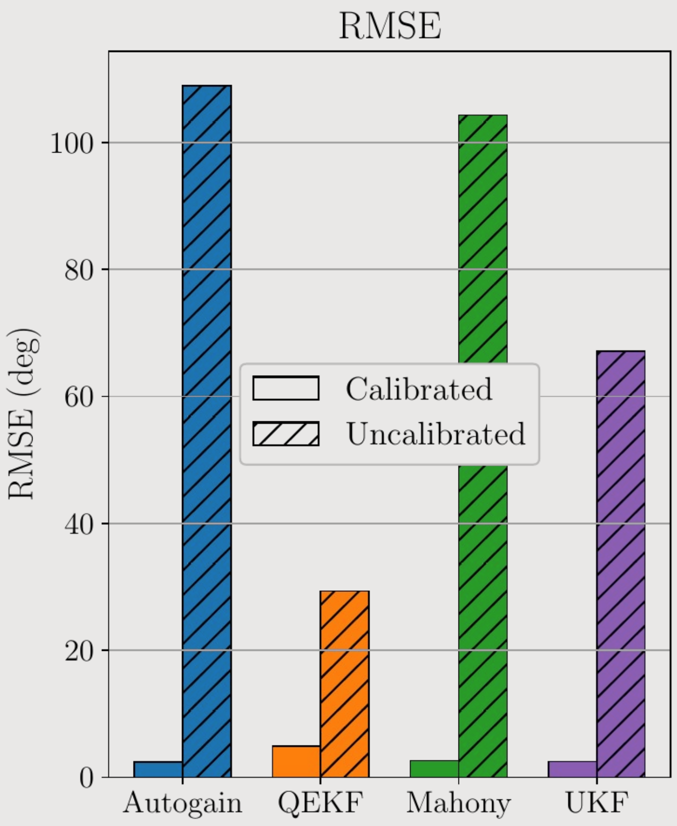
→ naive approach, derivative is unusable

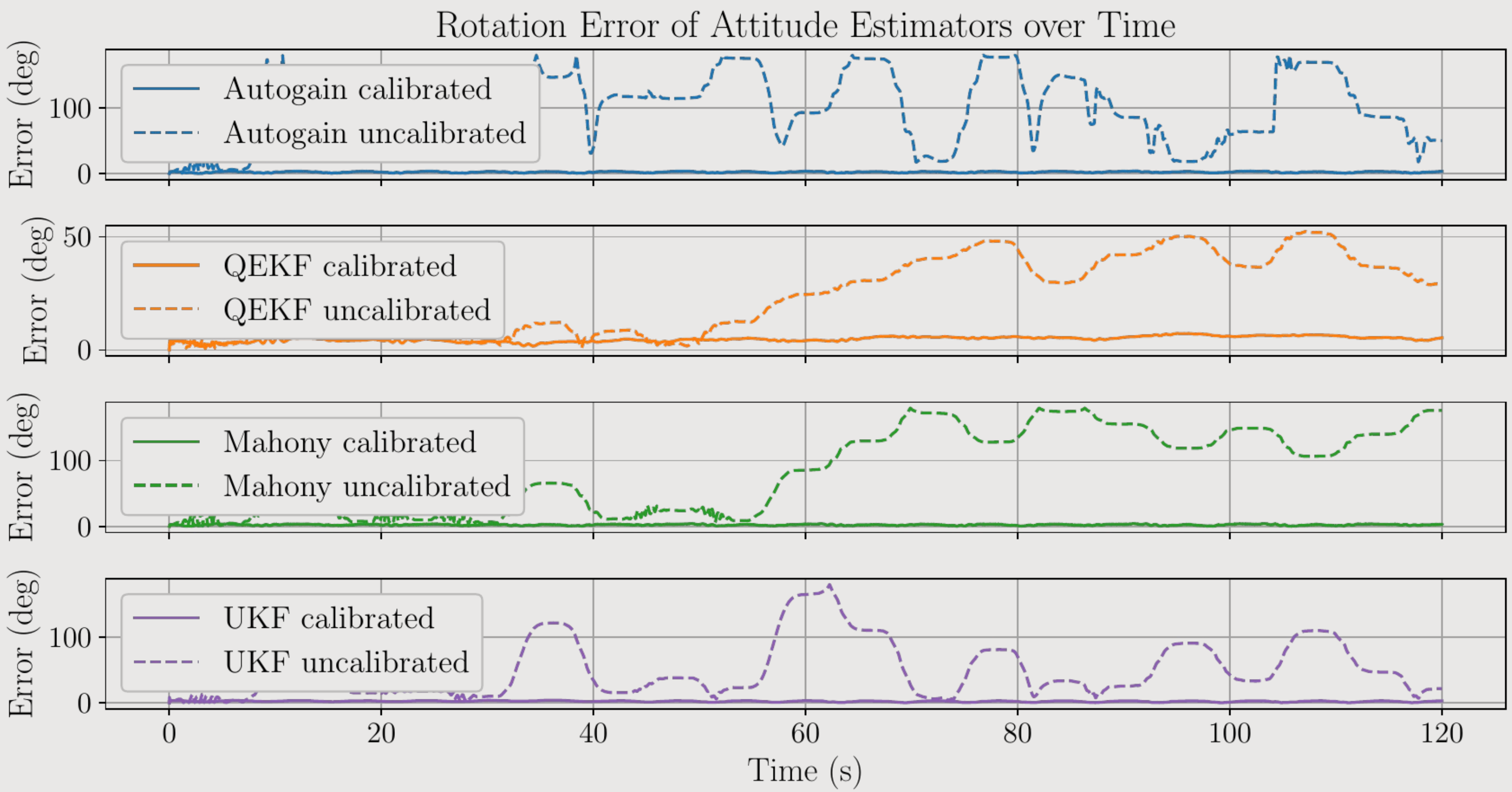
→ requires future data, use for offline calibration

→ requires no future data, use for online compensation

Results

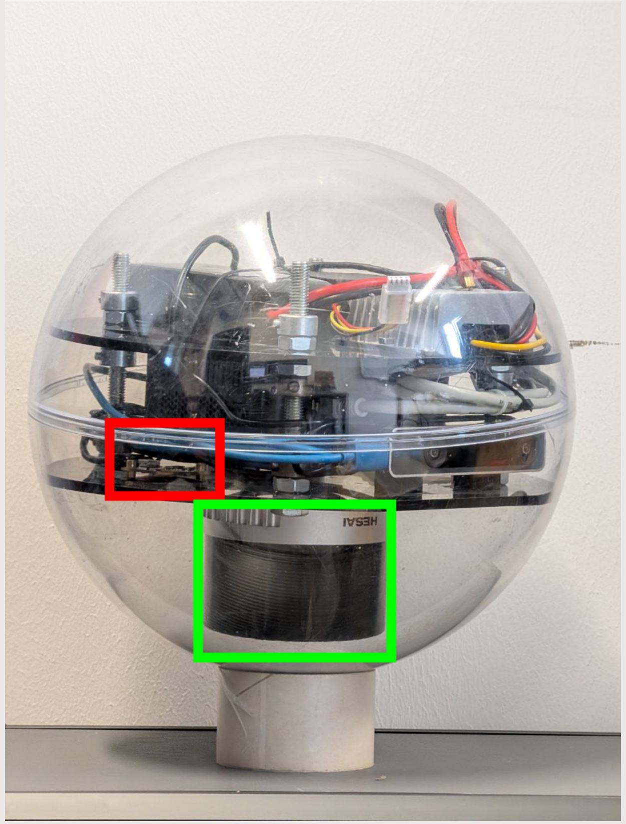
→ Simulated spherical system rolling with 6 rad/s, IMU lever-arm to the center of rotation is 8 cm

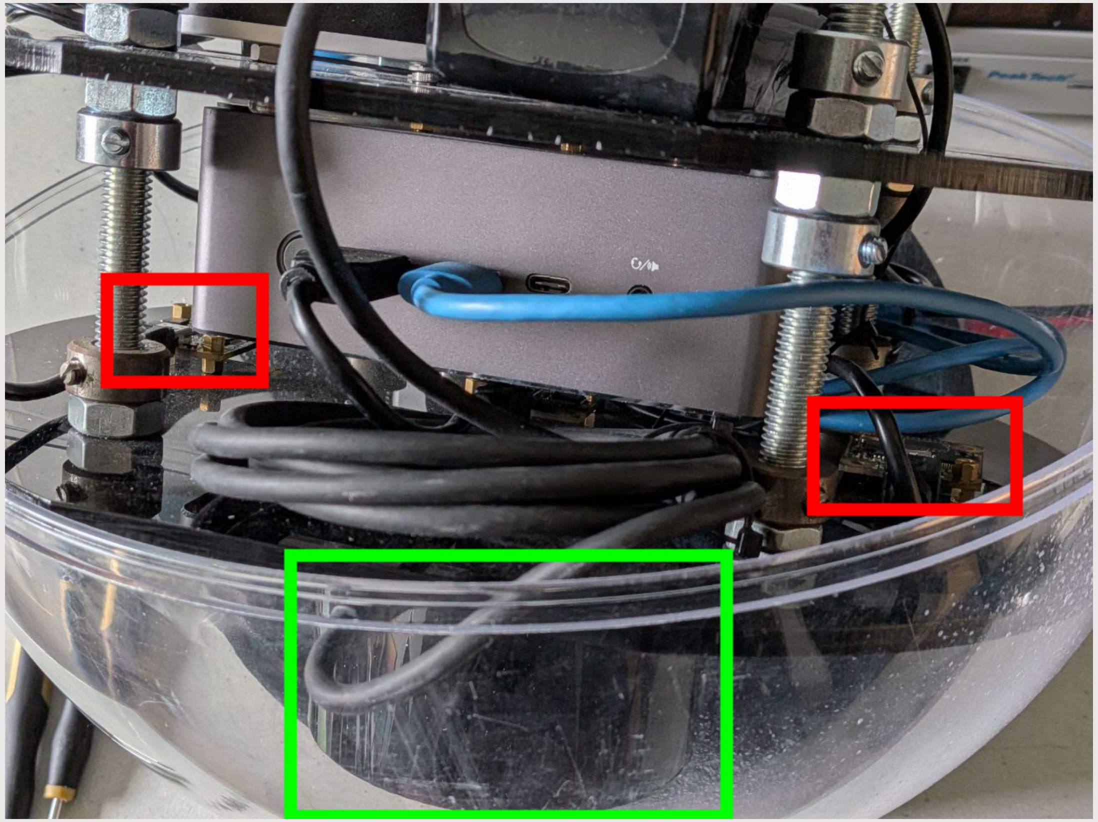




Results

video of the
bowling experiment

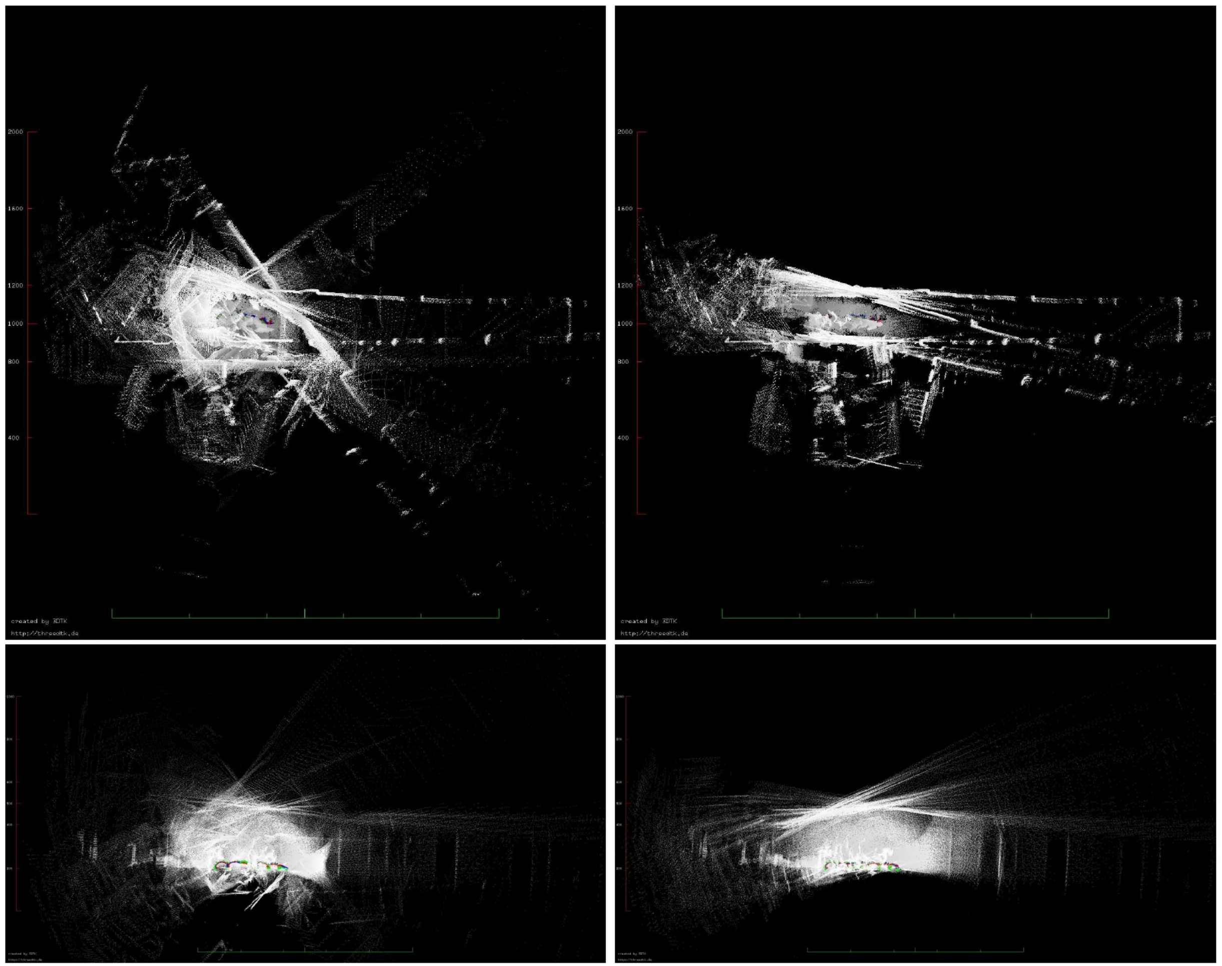




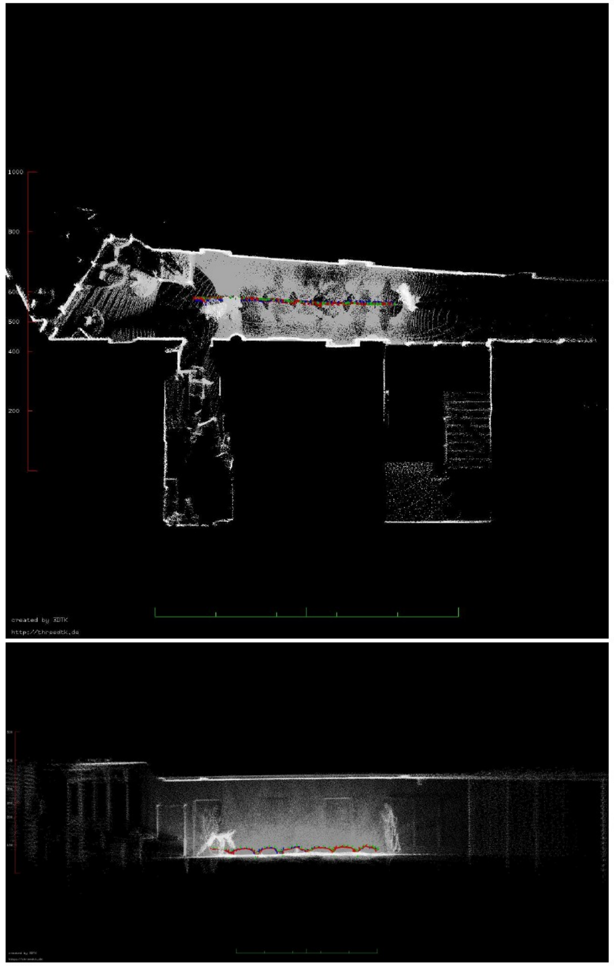
→ LiDAR sensor for visualization
→ Our calibration method is also able to estimate intrinsics between multiple IMUs !

Results

Using a single off-centered IMU

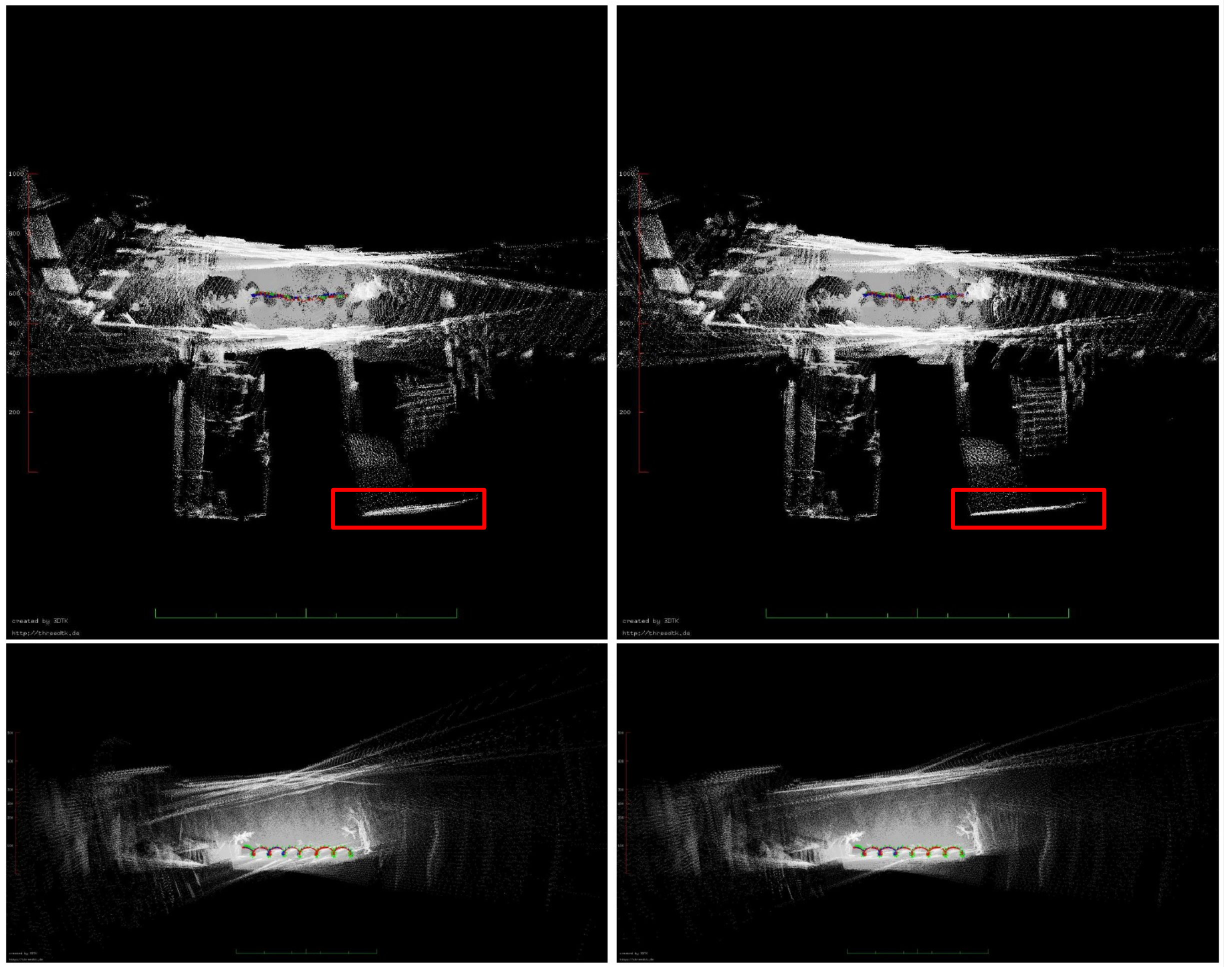


Reference (scan matching)

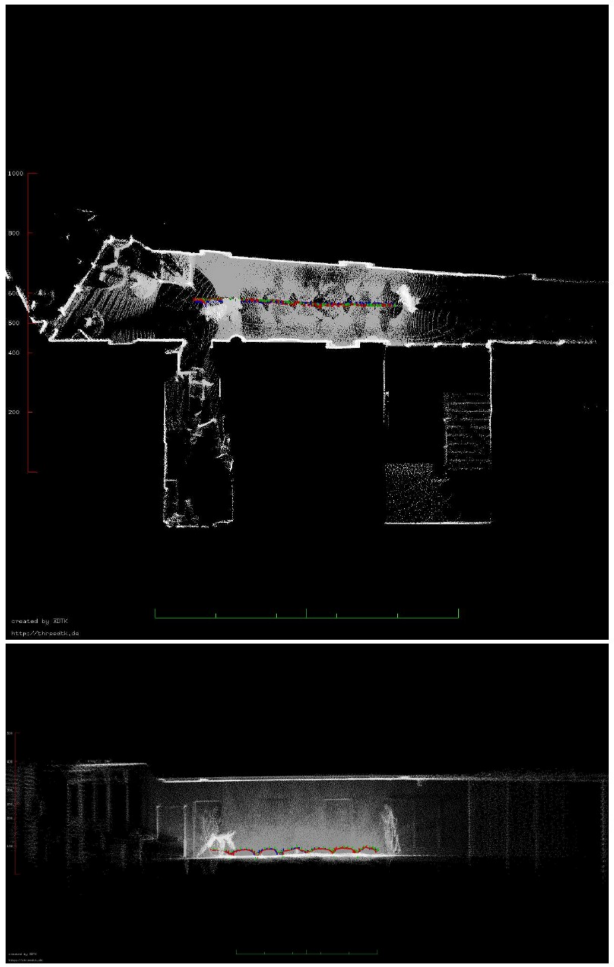


Results

Using two symmetrically placed, off-centered IMUs



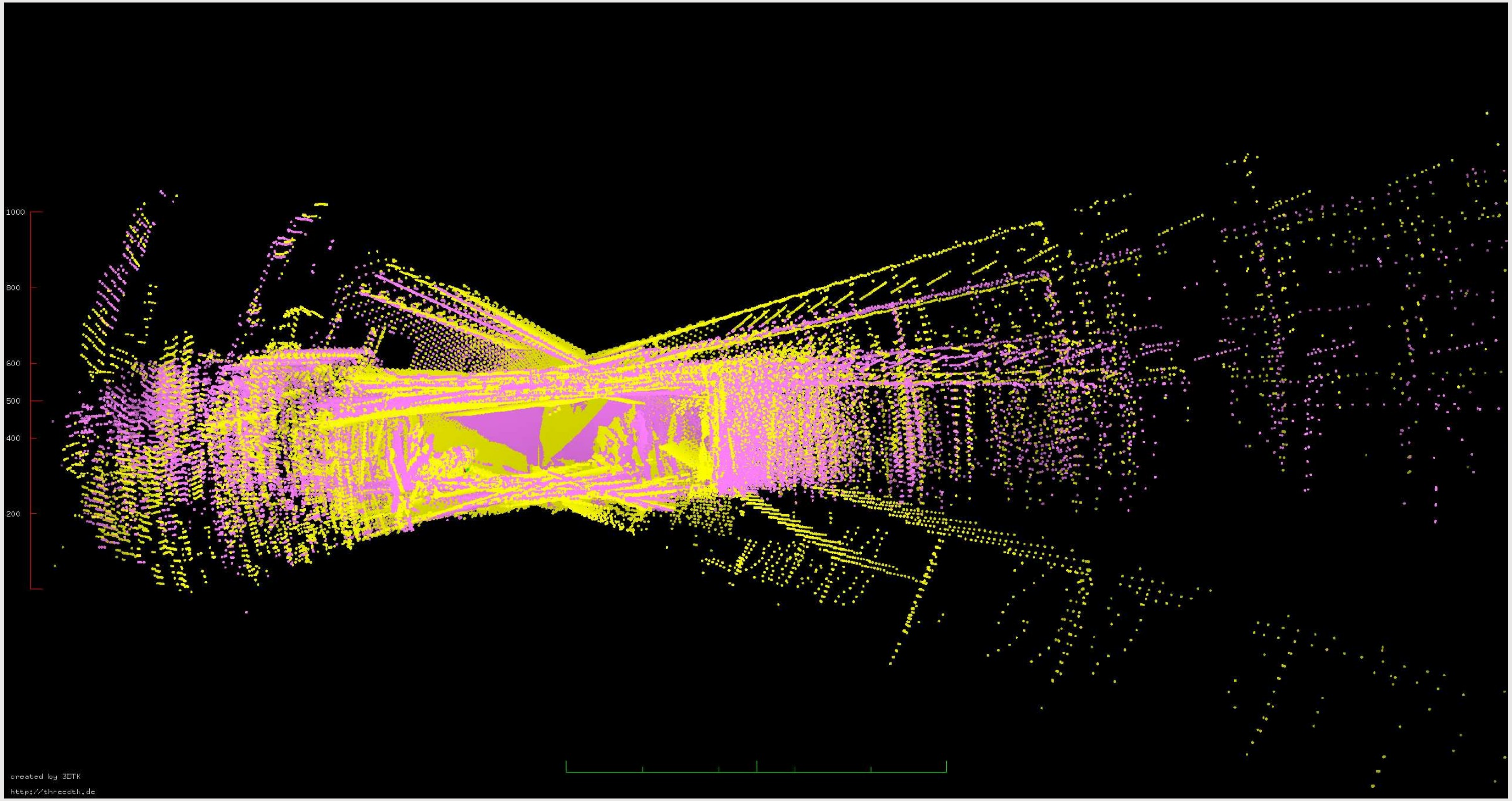
Reference (scan matching)



Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Results

Calibrated (with software compensation) vs Uncalibrated (only hardware compensation)



Conclusions

- Compensating tangential- and centripetal-accelerations significantly **improves attitude estimation**
- This benefits downstream tasks, such as LiDAR-inertial odometry
- Systems that benefit the most: **rotate fast**, or have a **large base-to-IMU lever-arm**
  - e.g. spherical systems, in-orbit spacecraft / satellites, drones, bio-inspired robots, ...
- We provide a robust **offline calibration** framework that estimates
  - **Intrinsic** parameters (bias, scale, misalignment)
  - **Extrinsic** parameters (base-to-IMU vector)
  - For **single IMU or multiple IMUs**
  - Based on the “multi-position approach”, **without external equipment**, suitable for in-field calibration
- We provide a robust **online compensation** method, carefully handling the derivative term using **DoG**
- Few to none state-of-the-art attitude filters care about doing this
- Mathematical details (derivation, optimization problem, selecting the kernel) in our paper!

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Future Work

- More testing on different types of systems
- More quantitative evaluation using ground truth attitudes
- Additional estimation of linear acceleration due to rolling on spherical system
- Tight integration into a model-based filter, e.g., Kalman

Thank you for the attention !
Any questions ?

Paper

Calibration code

Sphere code