Towards Autonomous MAV Exploration in Cluttered Indoor and Outdoor Environments

Korbinian Schmid

DLR, Robotics and Mechatronics Center (RMC)
Perception and Cognition
Mobile Robots (XRotor Group)

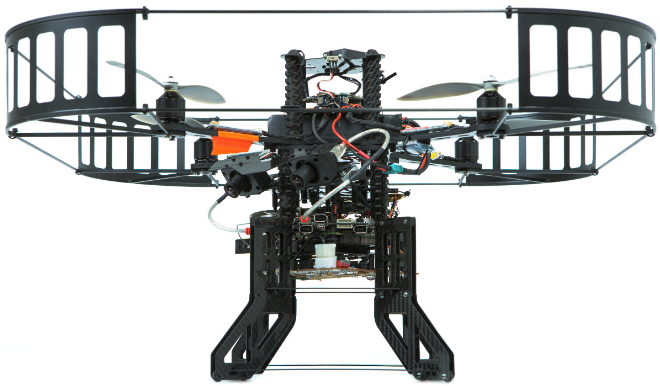
DLR Deutsches Zentrum für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

Motivation

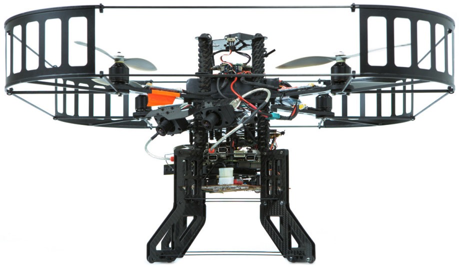
Autonomy?!

- No external navigation aids (GNSS)
- No reliable (high bandwidth, low latency) radio link
  - Full on-board navigation solution

Our systems



Why Multicopter MAVs?



- Small
- Light-weight
- Agile
- Safe
- Cheap
- Easy to fly
- But: limited payload!

Challenges

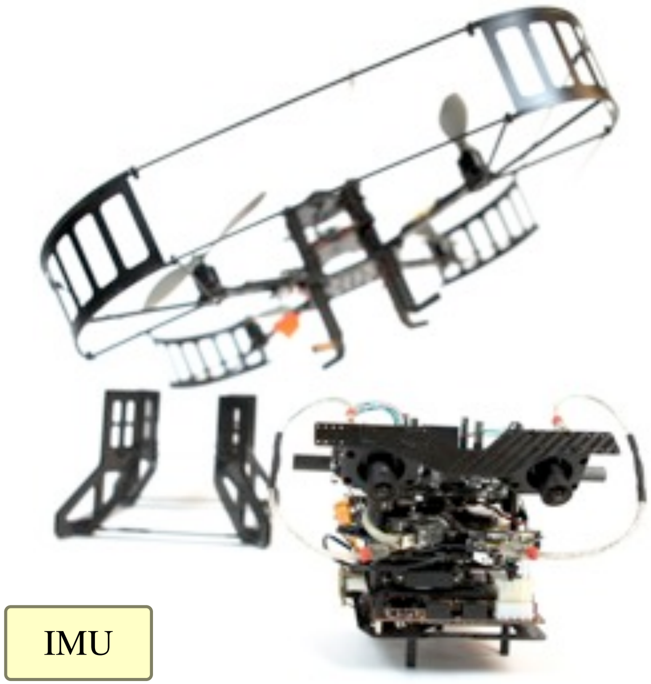
- Limited payload
- Limited computational resources
- Delayed data processing

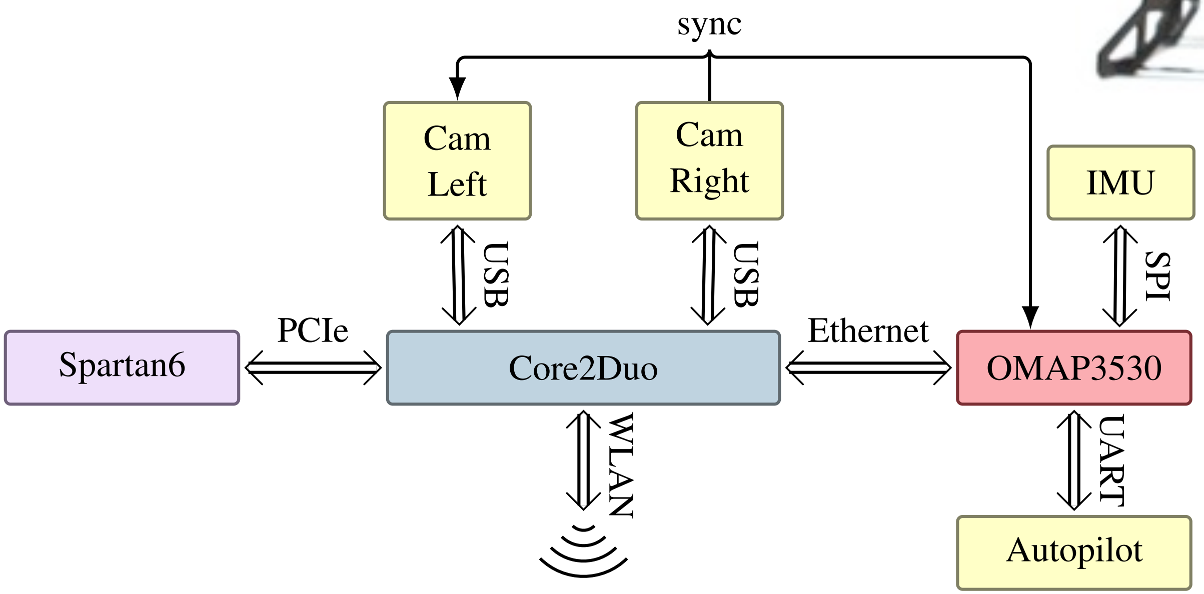
↕

- High system dynamics, inherently unstable
- High data load from exteroceptive sensors
- Computationally complex algorithms (mapping, path planning...)

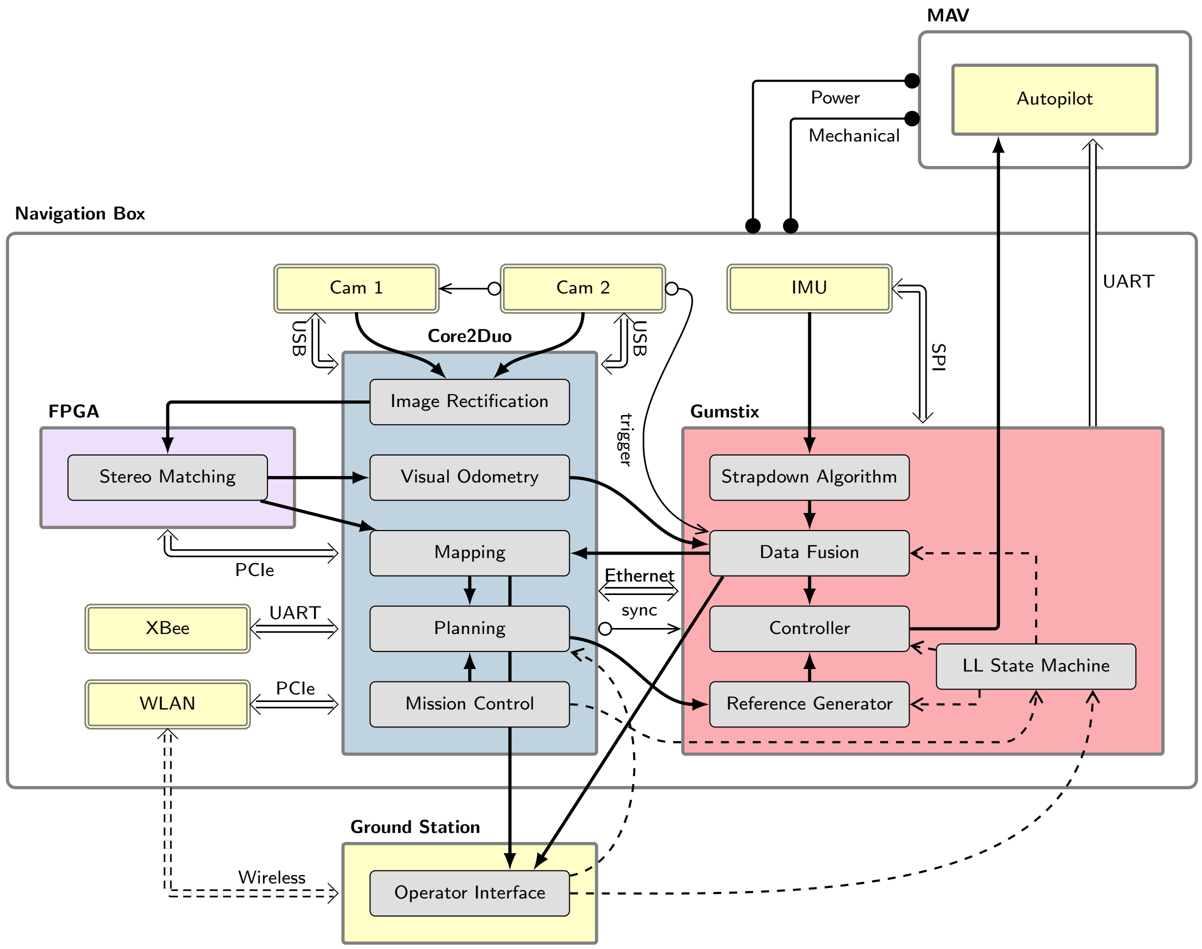
[Automatica, 2010]

System architecture
Navigation Box Hardware



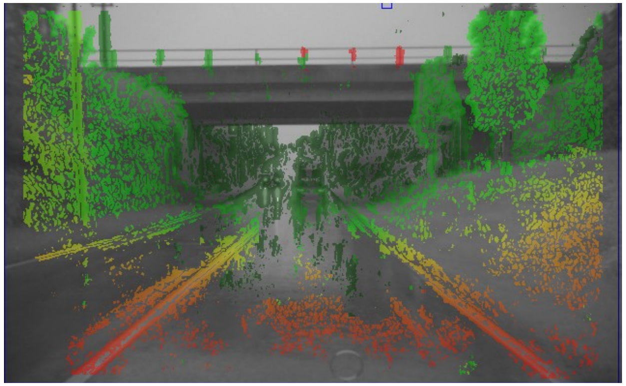


System architecture

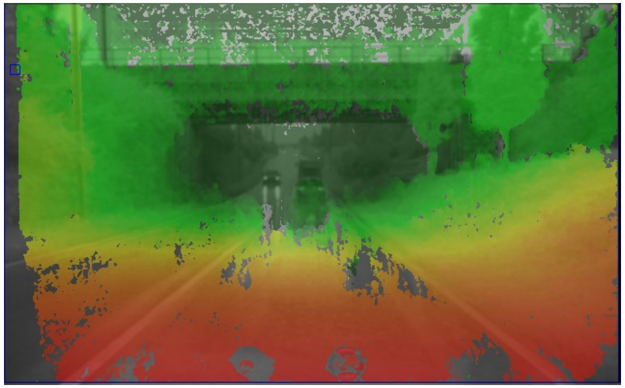


Depth image calculation on FPGA

- Semi Global Matching (SGM) [Hirschmüller, 2008]
- FPGA implementation: [Gehrig et al., 2009]
  - acceleration by parallelization
  - acceleration by pipelining
  - 0.5 MPixel depth images at 14.6 Hz



Correlation based Stereo

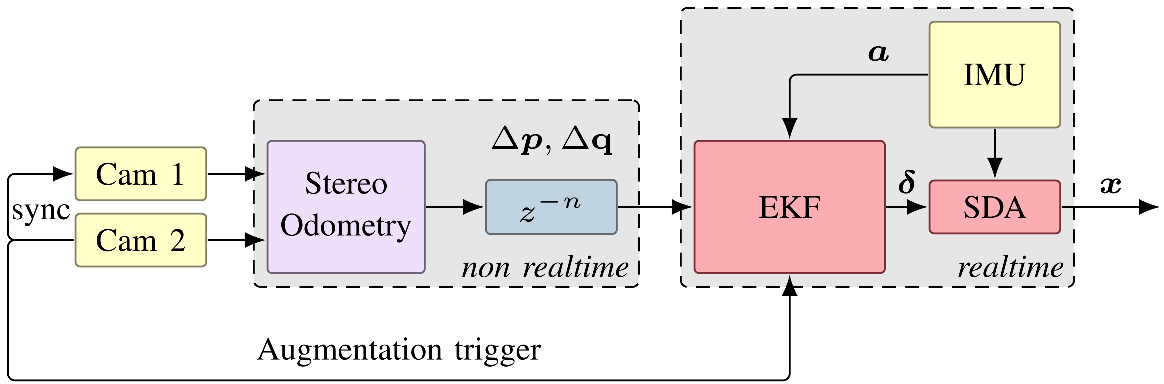


Semi-global matching

Images are a courtesy of Stefan Gehrig, Daimler AG

[Schmid et al. IROS 2012]

INS filter design



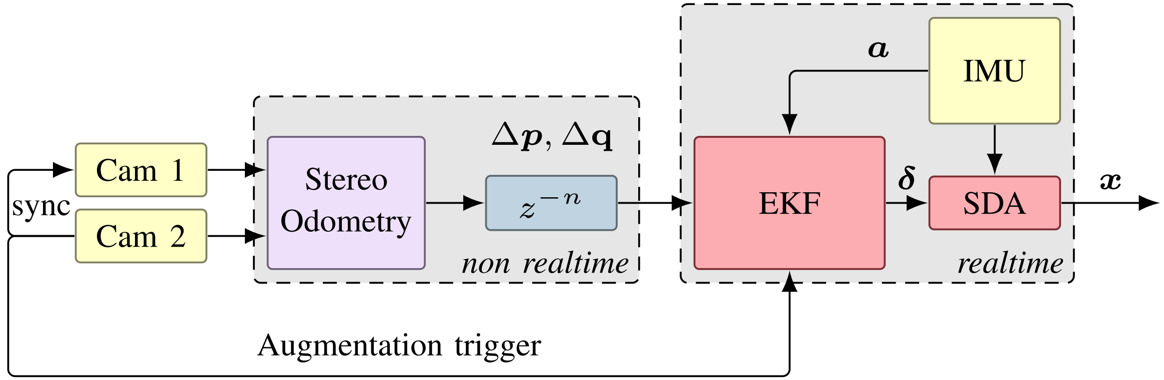
- Synchronization of realtime and non realtime modules by sensor hardware trigger

- Direct system state: $\boldsymbol{x} = \left(\boldsymbol{p}_{EB}^{E,T}\ \boldsymbol{v}_{EB}^{E,T}\ \boldsymbol{q}_{B}^{E,T}\ \boldsymbol{b}_a^T\ \boldsymbol{b}_\omega^T\right)^T \in \mathbb{R}^{16}$
- High rate calculation by „Strap Down Algorithm“ (SDA)

- Indirect system state: $\boldsymbol{\delta} = \left(\delta\boldsymbol{p}^T\ \delta\boldsymbol{v}^T\ \delta\boldsymbol{\sigma}^T\ \delta\boldsymbol{b}_a^T\ \delta\boldsymbol{b}_\omega^T\right)^T \in \mathbb{R}^{15}$
- Estimation by indirect Extended Kalman Filter (EKF)

- Measurement delay compensation by filter state augmentation

Indirect INS EKF advantages



- Separation of fast system dynamics from slow error dynamics
- Considering measurement time delays only in filter
- State calculation robust to filter divergence
- No system model
- Small angle approximation for attitude error (represented by an error angle vector of size 3 vs. quaternion representation of size 4)

Basic (direct) INS EKF with global position updates

- Prediction (state propagation):

$$\hat{\boldsymbol{x}}_{k+1}^{-} = \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_k^{+} + \boldsymbol{B}_k \boldsymbol{u}_k$$
$$\boldsymbol{P}_{k+1}^{-} = \boldsymbol{\Phi}_k \boldsymbol{P}_k^{+} \boldsymbol{\Phi}_k^T + \boldsymbol{G}_k \boldsymbol{Q}_k \boldsymbol{G}_k^T$$

- Update :

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{H_k} \boldsymbol{x}_k + \boldsymbol{n}_{\tilde{y}} = \begin{pmatrix} \boldsymbol{p_{E,B}^E} \\ \boldsymbol{q_B^E} \end{pmatrix} + \boldsymbol{n}_{\tilde{y}}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^{-} \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_k^{-} \boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1}$$
$$\boldsymbol{P}_k^{+} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_k^{-}$$
$$\hat{\boldsymbol{x}}_k^{+} = \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{K}_k (\tilde{\boldsymbol{y}}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^{-})$$

Basic (indirect) INS EKF with global position updates

- Direct system state calculation by Strap Down Algorithm
- EKF:
  - Prediction step (state error propagation):

$$\boldsymbol{P}_{k+1}^{-} = \boldsymbol{\Phi}_k \boldsymbol{P}_k^{+} \boldsymbol{\Phi}_k^T + \boldsymbol{G}_k \boldsymbol{Q}_k \boldsymbol{G}_k^T$$

  - Update :

$$\tilde{\boldsymbol{\delta}}_k = (\tilde{\boldsymbol{y}}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^{-}) + \boldsymbol{n}_{\tilde{y}} = \boldsymbol{H}_k \boldsymbol{\delta}_k + \boldsymbol{n}_{\tilde{y}}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^{-} \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_k^{-} \boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1}$$

$$\boldsymbol{P}_k^{+} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_k^{-}$$

$$\boldsymbol{\delta}_k = \boldsymbol{K}_k \tilde{\boldsymbol{\delta}}_k$$

- Correction of direct state

State augmentation (stochastic cloning)

- General measurement: $\tilde{\boldsymbol{y}}_k = \boldsymbol{H_k}\boldsymbol{x}_k + \boldsymbol{n}_{\tilde{y}}$
- Odometry measurement (direct): $\tilde{\boldsymbol{y}}_k = \boldsymbol{t}_{n_2} - \boldsymbol{t}_{n_1} + \boldsymbol{n}_{\tilde{y}} = \boldsymbol{H}_k \begin{pmatrix} \boldsymbol{x}_{n_1} \\ \boldsymbol{x}_{n_2} \end{pmatrix} + \boldsymbol{n}_{\tilde{y}}$

- „Stochastic cloning“ [Roumeliotis, 2002]
- Augmentation at time of measurement [Schmid et al., 2012]:

Time:	n_1	n_1+1	n_2	n_k
	$\hat{\boldsymbol{x}}_{n_1} = \hat{\boldsymbol{x}}_{n_1}$			
	$\hat{\bar{\boldsymbol{x}}}_{n_1} = \begin{pmatrix} \hat{\boldsymbol{x}}_{n_1} \\ \hat{\boldsymbol{x}}_{n_1} \end{pmatrix}$	$\hat{\bar{\boldsymbol{x}}}_{n_1+1} = \begin{pmatrix} \hat{\boldsymbol{x}}_{n_1} \\ \hat{\boldsymbol{x}}_{n_1+1} \end{pmatrix}$		
			$\hat{\bar{\boldsymbol{x}}}_{n_2} = \begin{pmatrix} \hat{\boldsymbol{x}}_{n_1} \\ \hat{\boldsymbol{x}}_{n_2} \\ \hat{\boldsymbol{x}}_{n_2} \end{pmatrix}$	$\hat{\bar{\boldsymbol{x}}}_k = \begin{pmatrix} \hat{\boldsymbol{x}}_{n_1} \\ \hat{\boldsymbol{x}}_{n_2} \\ \hat{\boldsymbol{x}}_{n_k} \end{pmatrix}$

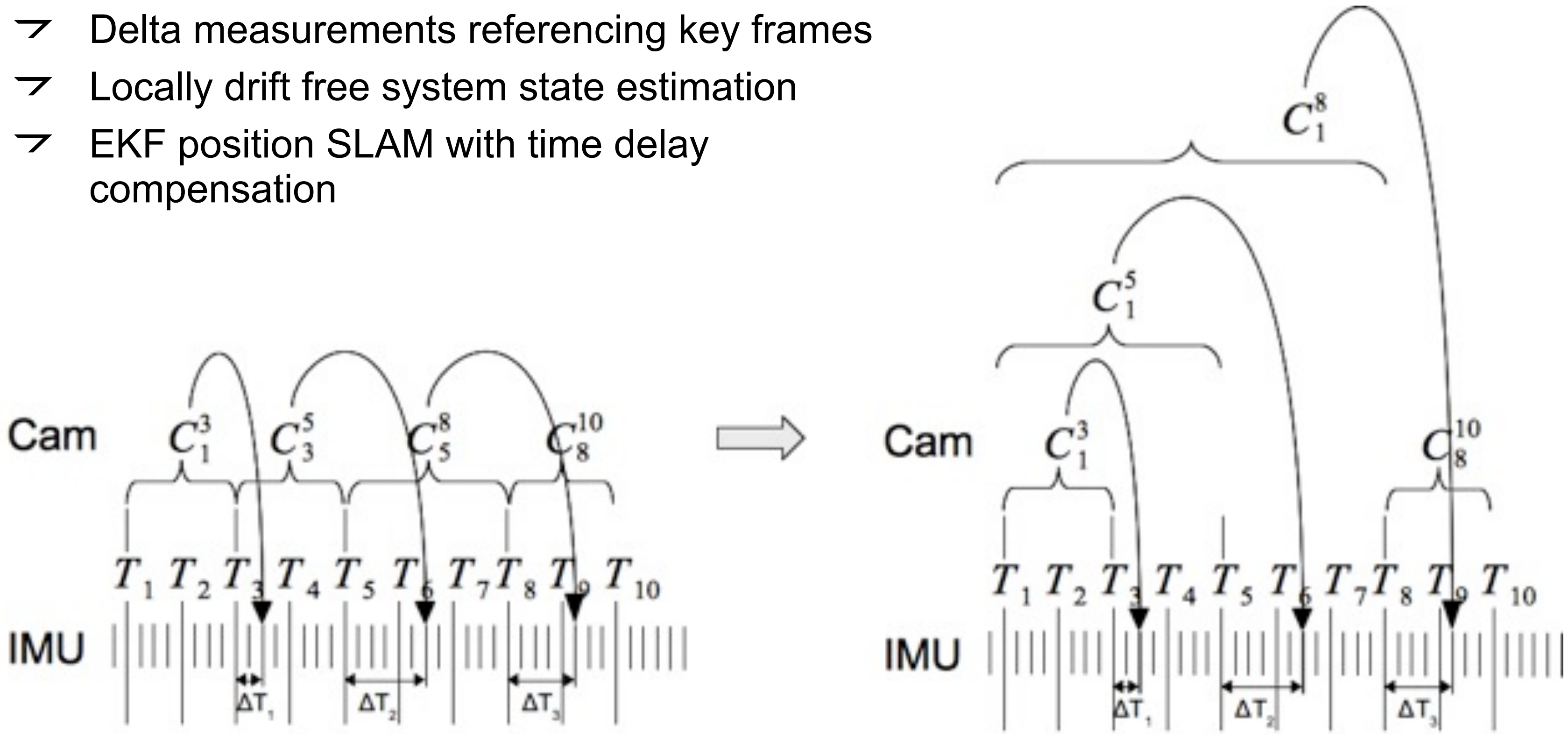
INS EKF with relative time delayed measurements

- EKF:
  - State augmentation at exact time of measurement:
    - Saving direct system state
    - Cloning of indirect filter state
  - Prediction step as basic INS EKF
  - Update:
    - Calculate delta pose from saved direct states
    - Calculate error residual from measurement
    - Standard EKF update referencing cloned indirect states in filter
  - Correction of direct states

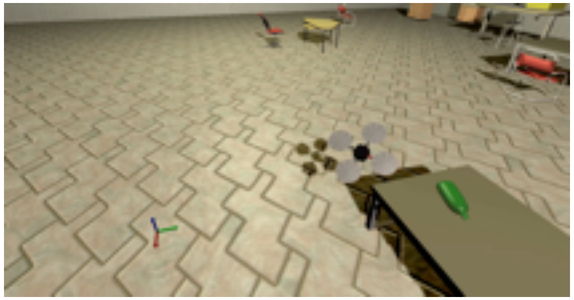
- Instant processing of arriving (time delayed) measurements

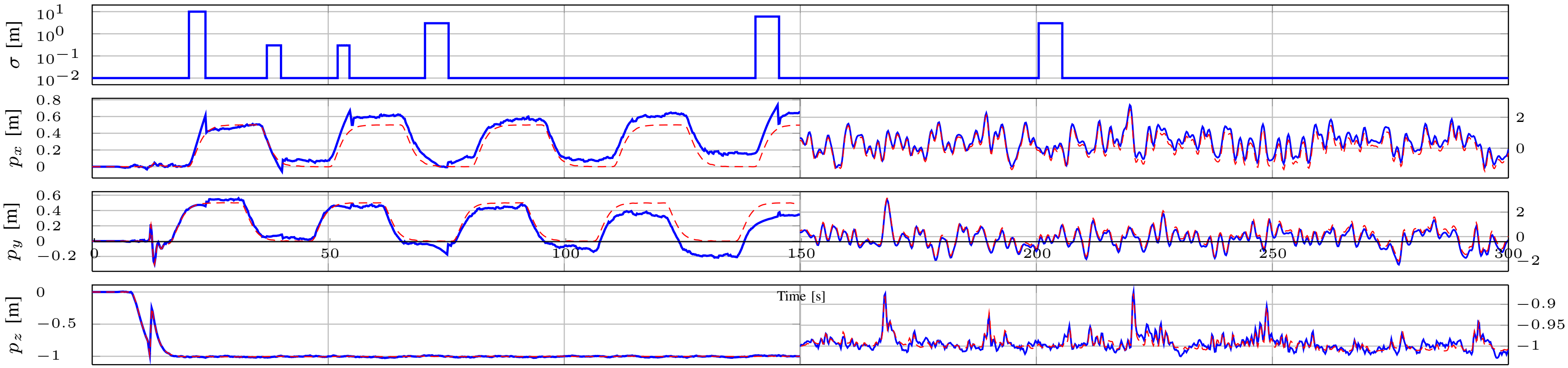
Key frame based stereo odometry

- Delta measurements referencing key frames
- Locally drift free system state estimation
- EKF position SLAM with time delay compensation



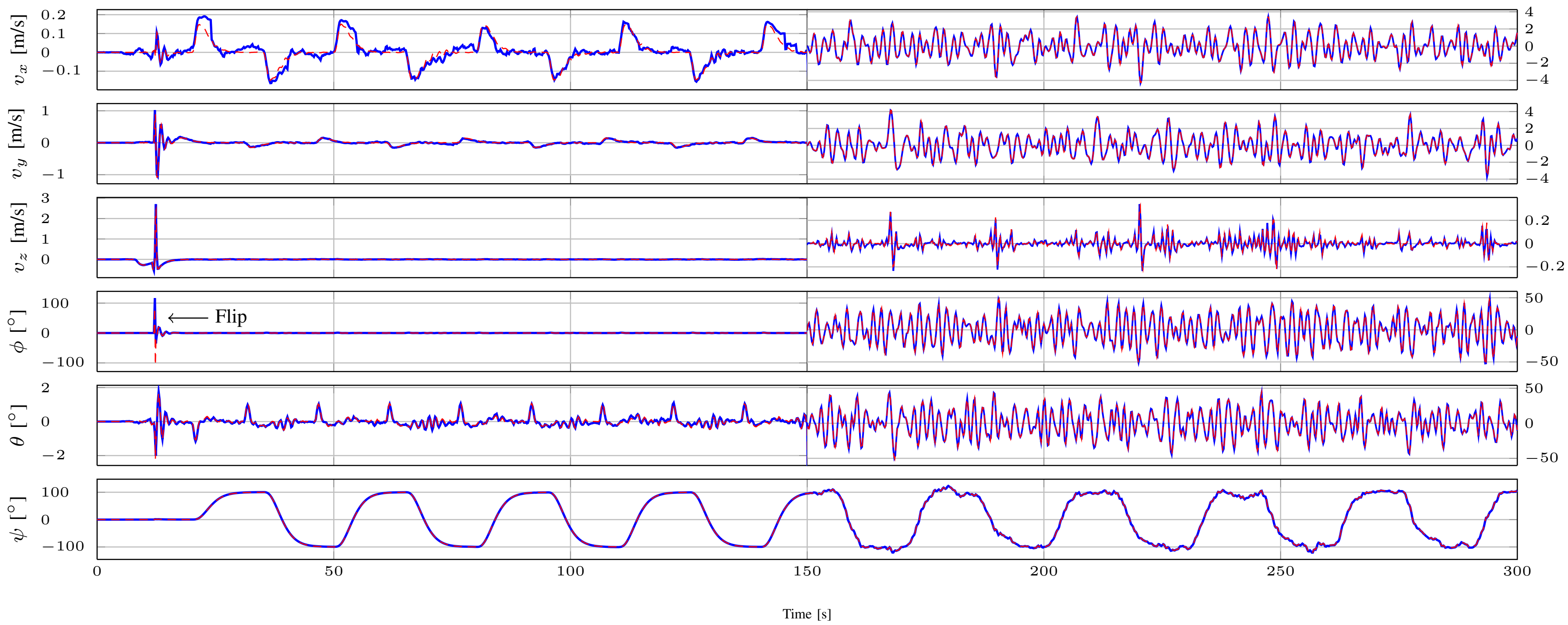
Simulation (Trajectory)





Simulation (velocity, attitude)

- Velocity up to 4m/s
- Roll/Pitch angles up to 50deg



Simulation: FPGA acceleration vs. RMSE

- RMSE for delay variation:
  - const for position
  - linear for velocity
- Acceleration by parallelization
  - higher frequency
  - lower latency

exponential improvement

- RMSE for frequency variation:
  - exponential for position
  - exponential for velocity
- Acceleration by pipelining
  - higher frequency
  - constant latency

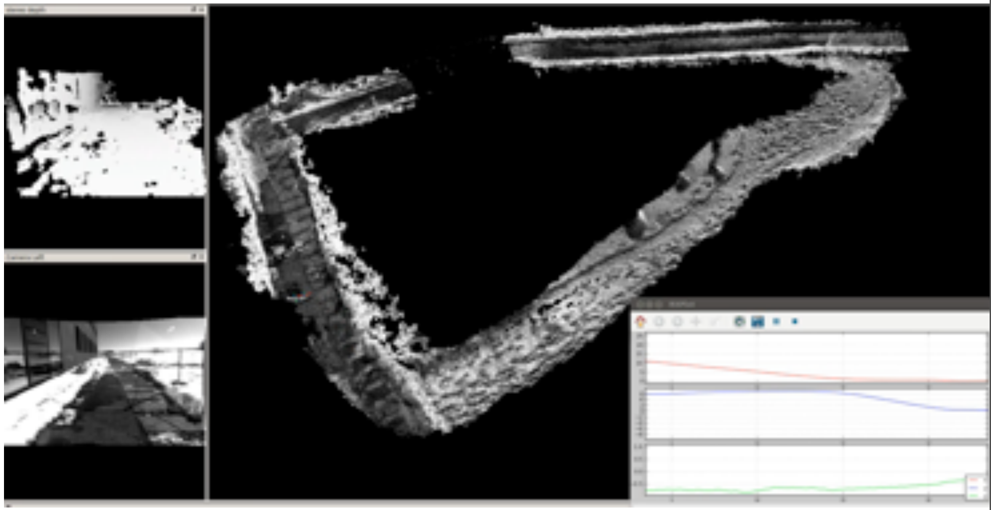
exponential improvement

- Estimator properties fit well acceleration by FPGA

[K. Schmid and H. Hirschmüller ICRA 2013]

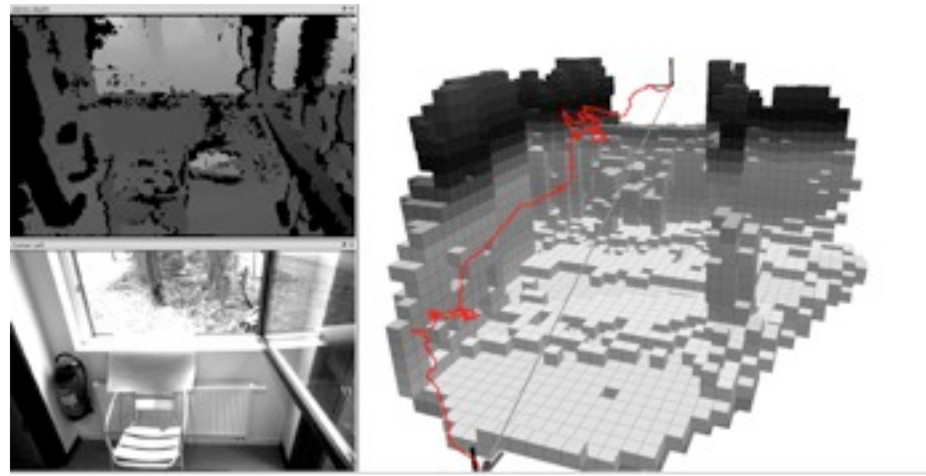
Robustness test

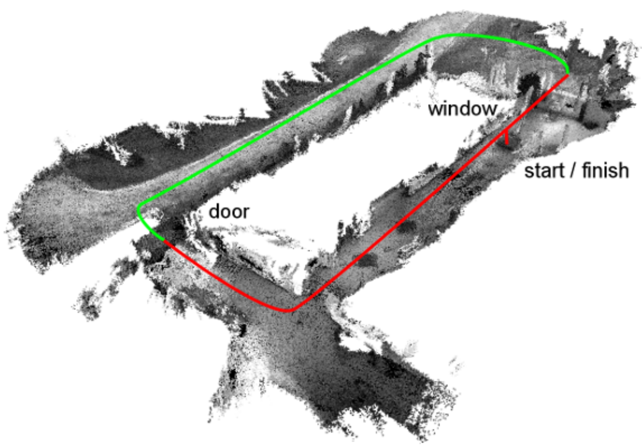
- 70 m trajectory
- Ground truth by tachymeter
- 5 s forced vision drop out with translational motion
- 1 s forced vision drop out with rotational motion
- Estimation error < 1.2 m
- Odometry error < 25.9 m
- Results comparable to runs without vision drop outs

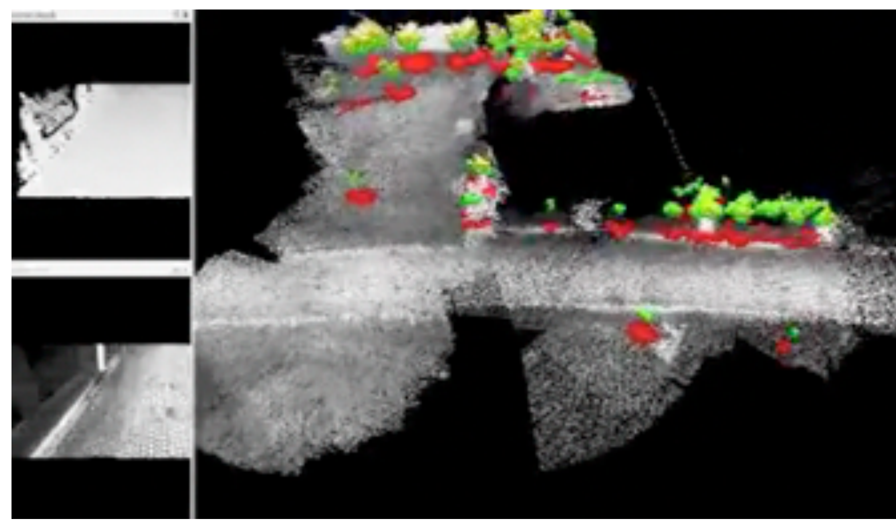


Mixed indoor/outdoor exploration

- Autonomous indoor/outdoor flight of 60m
- Mapping resolution: 0.1m
- Leaving through a window
- Returning through door

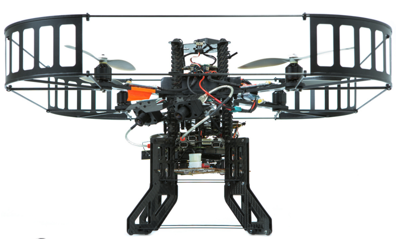


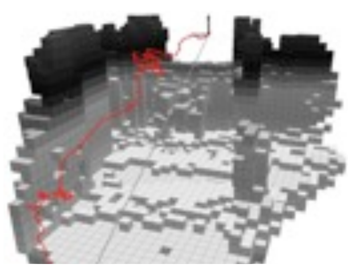


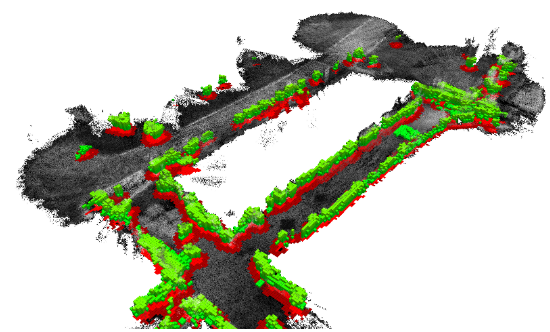


Conclusion

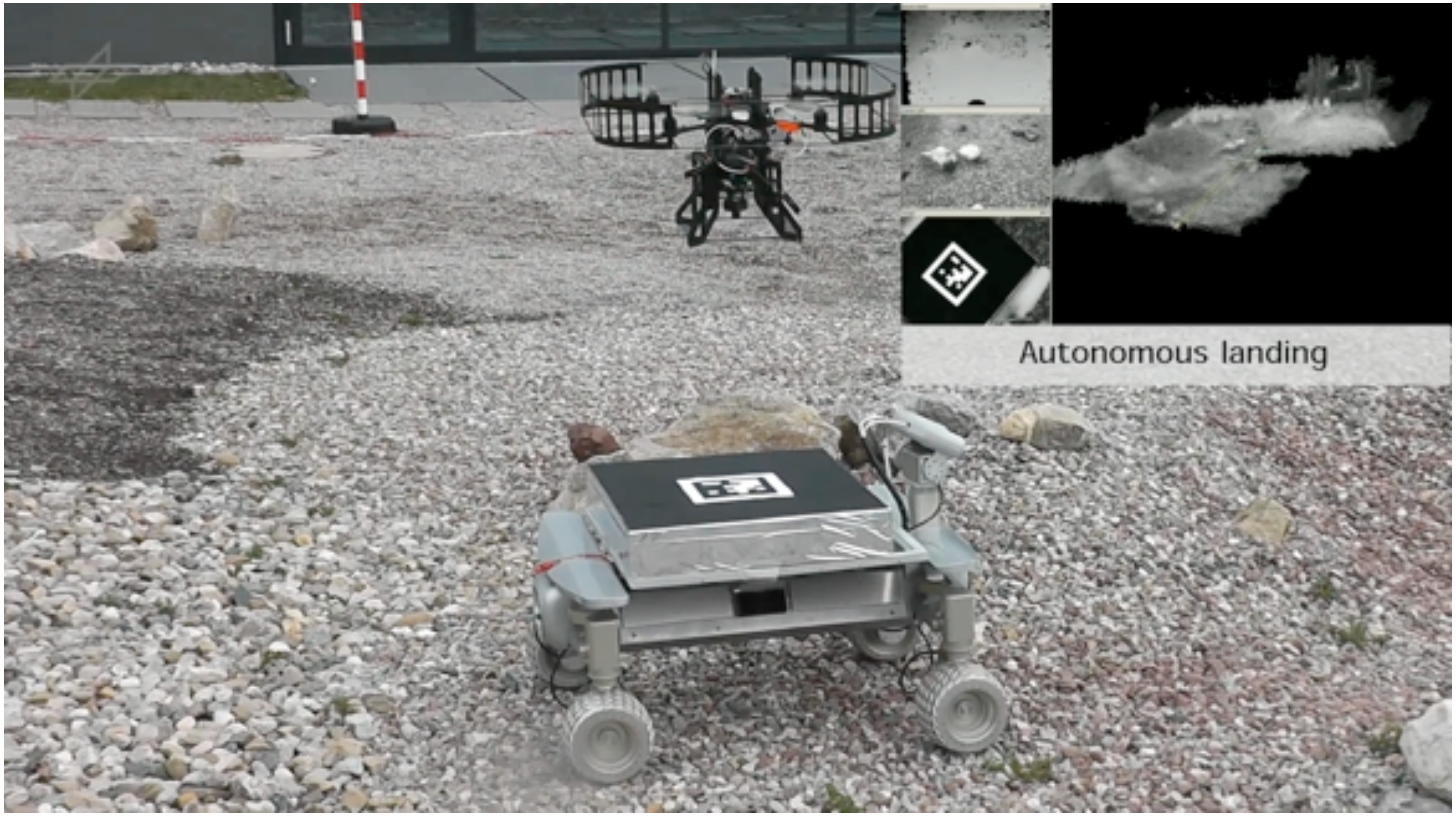
- Multicopters for SAR and disaster management scenarios
- System concept for autonomous MAVs:
  - FPGA based stereo image processing
  - Key frame based stereo odometry
  - INS fusion with delay compensation by EKF
  - Mapping, path planning, mission control
- System state estimation improvement by FPGA acceleration
- Robust navigation concept for indoor/outdoor exploration







What‘s next?



Multicopters in SAR and disaster management scenarios

The RMC XRotor team

Thank you for your attention! Questions?

http://mobilerobots.dlr.de/systems/multicopters.
Or google: DLR XRotor