State Estimation for highly dynamic flying Systems using Key Frame Odometry with varying Time Delays

Korbinian Schmid[1], Felix Ruess[2], Michael Suppa[1] and Darius Burschka[2]

Abstract— System state estimation is an essential part for robot navigation and control. A combination of Inertial Navigation Systems (INS) and further exteroceptive sensors such as cameras or laser scanners is widely used. On small robotic systems with limitations in payload, power consumption and computational resources the processing of exteroceptive sensor data often introduces time delays which have to be considered in the sensor data fusion process. These time delays are especially critical in the estimation of system velocity. In this paper we present a state estimation framework fusing an INS with time delayed, relative exteroceptive sensor measurements. We evaluate its performance for a highly dynamic flight system trajectory including a flip. The evolution of velocity and position errors for varying measurement frequencies from 15Hz to 1Hz and time delays up to 1s is shown in Monte Carlo simulations. The filter algorithm with key frame based odometry permits an optimal, local drift free navigation while still being computationally tractable on small onboard computers. Finally, we present the results of the algorithm applied to a real quadrotor by flying from inside a house out through the window.

I. Introduction

Accurate state estimation is essential for mobile robot control. The subclass of Micro Aerial Vehicles (MAV) poses special requirements on state estimation algorithms. MAVs are inherently unstable and have fast system dynamics. For stabilization at least the system orientation and velocity must be estimated. The estimation algorithm has to be computationally efficient as the system payload usable for processing resources is often very limited. Sensor measurements that have to be preprocessed may be time delayed and only available at a low frequency compared to the natural frequency of the system.

Inertial Navigation Systems (INS) were shown to be suitable for system state estimation for MAVs [1]. The system motion is calculated by the strapdown algorithm with sensor inputs from an Inertial Measurement Unit (IMU). Errors in the system states resulting from sensor noise and varying sensor biases are estimated by an extended Kalman Filter (EKF). Therefore further absolute measurement sensors such as GPS as the main position and velocity sensor, a barometer as height sensor and a magnetometer mainly for yaw stabilization are fused in the filter. Nevertheless, these classical INS are only suitable for outdoor applications with GPS coverage.

Several approaches have been shown to replace GPS by other exteroceptive sensors in GPS-denied environments. The

[1]with DLR (German Aerospace Center), Robotics and Mechatronics Center (RMC), Perception and Cognition Münchner Str. 20, 82234 Wessling, Germany

[2]with Technische Universität München, Institute for Computer Science VI Parkring 13, 85748 Garching, Germany
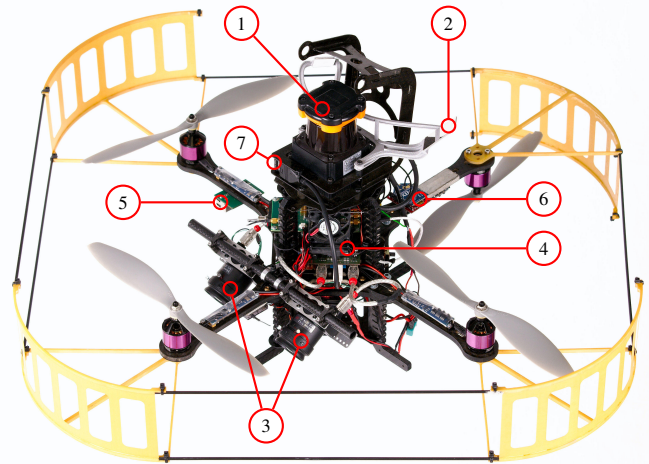

Fig. 1: Experimental quadrotor platform showing (1) laser scanner, (2) mirrors, (3) stereo cameras, (4) a modular computation stack, (5) wired ethernet connection, (6) XBee modem, and (7) WLAN stick. One of the propellers is pointing downwards to improve the view of a front-facing camera (not depicted).

probably most widely used method is the use of external cameras with known locations to track the 6 Dof MAV pose [2]. While this is useful for testing purposes, it can only be used in controlled environments. The more general approach is to mount exteroceptive sensors like cameras or laser scanners on the MAV. Downward looking cameras are used by Herisse et al. [3] to stabilize the MAV position during hover with optical flow and by Eberli et al. [4] to track a known landmark. However, they are limited in terms of dynamics and the need for a known landmark.

To be able to better cope with higher dynamics Achtelik et al. [5] use a fast low level attitude estimation based on IMU measurements, with position and velocity estimation done in a separate filter. A monocular Simultaneous Localization and Mapping (SLAM) algorithm is used to provide position feedback to the second filter in order to compensate for drifts. In [6] all processing is also done onboard and the camera is synchronized with the IMU to provide a common timebase. Also Cheviron et al. [7] have developed an efficient algorithm for fusing inertial and visual measurements.

However, the introduced approaches do not properly compensate for measurement and processing time delays and are only suitable in situations where sensor speed is capable of

observing the system dynamics.

In this contribution we propose a state estimation system for MAVs in GPS denied environments that is stable up to the dynamic limits of the system. Estimates of velocity and attitude as well as gyroscope and accelerometer biases are needed to stabilize the system. Delayed relative position measurements are used for state correction, therefore position is also estimated. We use stochastic cloning [8] to process the relative position and orientation information from a general odometry sensor, in our case an onboard stereo vision odometry system. In contrast to the original stochastic cloning, we leverage previously held key frames as a local reference instead of just using relative measurements between the two latest images. As long as the system navigates in a small area, features used for localization stay visible for some time. It is possible to reuse these features and calculate a delta pose change referencing the initial key frame. In this way, a local drift free navigation is possible. In contrast to sparse SLAM methods [9], where the positions of all features are estimated, we only estimate vehicle positions within the filter.

A further extension is the consideration of sensor time delays resulting from communication delays and data pre-processing time which is of importance for highly dynamic systems with limited processor resources. Our filter framework can directly process time delayed absolute and (overlapping) relative measurements at the time of arrival.

In the simulation part of this paper we demonstrate the efficiency of the proposed system. We simulate a quadrotor using low and highly dynamic trajectories including a flip. We vary the odometry frequency and time delay for different runs in the range of 15Hz to 1Hz and between 5ms to 1s respectively. Results of the state estimation in combination with a controller are shown on a real quadrotor flight from inside a house through a window to the outside. Finally we give a conclusion from the achieved results and an outlook on further research topics.

II. SYSTEM STATE ESTIMATION

We use a MEMS IMU with gyroscopes and accelerometers to calculate the system state for highly dynamic movements. The drifting states are stabilized by further sensors: a low frequency key frame based odometry system and an absolute height measurement. A system block diagram is shown in Figure 2.
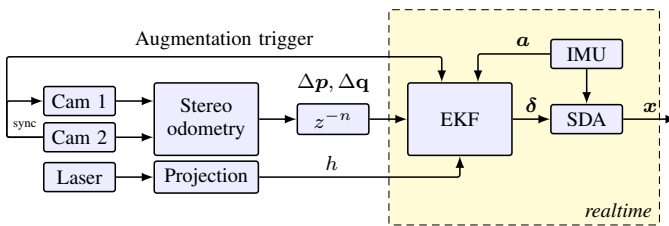

Fig. 2: EKF INS state estimator in feedback configuration with height and time delayed odometry sensor.

The current, direct system state $\boldsymbol{x}$ is calculated by integrating accelerations and angular rates of the IMU in the strap-

down algorithm (SDA) [10]. An indirect feedback Kalman Filter uses measurements, in our case visual odometry and laser height, to estimate the errors rather than the direct state. Hence, it is also known as error state space Kalman Filter. The odometry sensor measurement is assumed to be time delayed. The EKF defines the indirect (error) system state $\boldsymbol{\delta}$. The direct system state is corrected by subtracting the estimated errors after each update step. After correction, the filter state is reset to zero. The decoupling of the error state estimation problem from the full system state observation has several advantages [11]:

- Fast system dynamics tracked by the INS system using a high execution frequency for the SDA. The slower error dynamics can be accurately tracked by a filter running at a lower frequency.
- Time delays in additional sensor measurements have only to be considered in the indirect filter.
- The INS can operate even in the case of a failure of the filter.
- No system model is needed.
- As the system state is corrected after each filter update we can assume small angles in the attitude error, which can be efficiently represented by an error angle vector of size 3.

The EKF and SDA algorithms run on a separate realtime operating system whereas the rest of the sensor processing is carried out on a system without special timing constraints. The realtime system receives a hardware synchronization signal at the exact time of a camera or laser measurement.

A. Strapdown algorithm

The evolution of the direct system state $\boldsymbol{x} \in \mathbb{R}^{16}$ is calculated by the SDA using IMU data. We model a discrete single sensor measurement at time k as:

$$\tilde{m}_k = m_k + b_k + n_k \tag{1}$$

where m_k is the ideal measurement, b_k is a time varying bias and n_k is a zero mean Gaussian random variable. By including the varying sensor biases in the direct system state we define:

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{p}_{ob}^{o,T} & \boldsymbol{v}_{ob}^{n,T} & \boldsymbol{q}_b^{o,T} & \boldsymbol{b}_a^{b,T} & \boldsymbol{b}_\omega^{b,T} \end{pmatrix}^T \tag{2}$$

with the sub-states:

- body position ($\boldsymbol{p}_{ob}^o$) in an earth fixed frame defined by the initial starting position and heading of the system (o-frame)
- body velocity ($\boldsymbol{v}_{ob}^o$) in the o-frame
- body orientation quaternion ($\boldsymbol{q}_b^o$) in respect to the o-frame
- three bias estimates $\boldsymbol{b}_a^b$ for each acceleration sensor axis
- three bias estimates $\boldsymbol{b}_\omega^b$ for the corresponding gyroscope sensor axis

1) Body orientation: IMU gyroscopes measure the body angular rate in the body frame (b-frame) in respect to an earth centered inertial frame (i-frame):

$$\boldsymbol{\omega}_{ib}^b = \boldsymbol{\omega}_{nb}^b + \mathbf{C}_b^{n,T}(\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \tag{3}$$

where $\boldsymbol{\omega}_{nb}^b$ denotes the angular rate of the body with respect to the North East Down (NED) local-tangent-plane navigation frame (n-frame), expressed in the b-frame. $\mathbf{C}_b^{n,T}$ is the Direction Cosine Matrix (DCM) rotating n-frame onto b-frame. As the earth rotation rate $\boldsymbol{\omega}_{ie}^n$ and the transport rate $\boldsymbol{\omega}_{en}^n$ are small compared to the noise of the MEMS gyroscopes, we can assume:

$$\boldsymbol{\omega}_{ie}^n \approx 0 \,, \boldsymbol{\omega}_{en}^n \approx 0 \tag{4}$$

Equation (3) can be approximated as:

$$\boldsymbol{\omega}_{ib}^b \approx \boldsymbol{\omega}_{nb}^b \approx \boldsymbol{\omega}_{ob}^b \tag{5}$$

With the assumption that the angular rate $\boldsymbol{\omega}_{ob,k}^b$ at time step k is constant within the measuring interval from T_{k-1} to T_k we get for the delta orientation vector at time k:

$$\Delta\boldsymbol{\sigma}_{b,k}^{b,k-1} \approx \boldsymbol{\omega}_{ob,k}(T_k - T_{k-1}) \tag{6}$$

We use a quaternion representation with the vector part in the first three and the scaling part in the last component. The delta rotation quaternion between T_{k-1} and T_k is given by:

$$\Delta\mathbf{q}_{b,k}^{b,k-1} = \begin{pmatrix} cos\left(\frac{\Delta\sigma_{b,k}^{b,k-1}}{2}\right) \\ \frac{\Delta\boldsymbol{\sigma}_{b,k}^{b,k-1}}{\Delta\sigma_{b,k}^{b,k-1}} sin\left(\frac{\Delta\sigma_{b,k}^{b,k-1}}{2}\right) \end{pmatrix} \tag{7}$$

where $\Delta\sigma_{b,k}^{b,k-1} = \|\Delta\boldsymbol{\sigma}_{b,k}^{b,k-1}\|$. And so, a new orientation quaternion, representing the rotation from the b to the o-frame for time step k, can be calculated as:

$$\mathbf{q}_{b,k}^o = \mathbf{q}_{b,k-1}^o \Delta\mathbf{q}_{b,k}^{b,k-1} \tag{8}$$

2) Position: With assumption (4) and a local operation area the differential equation for the body position in the world frame is approximated by:

$$\dot{\boldsymbol{x}}_{ob}^o \approx \boldsymbol{C}_n^o \boldsymbol{v}_{ob}^n \approx \boldsymbol{v}_{ob}^o \tag{9}$$

with $\boldsymbol{C}_n^o$ rotating the n-frame on the o-frame. We find the approximated differential equation of the body speed in the o-frame as:

$$\dot{\boldsymbol{v}}_{ob}^o \approx \boldsymbol{C}_b^o \boldsymbol{a}_{ib}^b + \boldsymbol{g}^o \tag{10}$$

where $\boldsymbol{C}_b^o$ is the DCM corresponding to $\boldsymbol{q}_b^o$, $\boldsymbol{a}_{ib}^b$ the accelerations measured by the IMU and $\boldsymbol{g}^o$ the gravity vector in the o-frame. Under the assumption of a constant acceleration between T_{k-1} and T_k the difference equation for the body speed at time k can be found as:

$$\Delta\boldsymbol{v}_{ob,k}^o \approx \mathbf{C}_{b,k-1}^o \left(\boldsymbol{a}_{ib}^b + \frac{1}{2}\Delta\boldsymbol{\sigma}_{b,k}^{b,k-1} \times \boldsymbol{a}_{ib}^b\right)(T_k - T_{k-1}) \tag{11}$$

B. Indirect Kalman Filter

The error state vector of our filter ($\boldsymbol{\delta}$) includes the errors in position ($\boldsymbol{\delta}_p^o$), velocity ($\boldsymbol{\delta}_v^o$) and attitude ($\boldsymbol{\delta}_\psi^o$) in the o-frame as well as the errors of the biases of the accelerometers ($\boldsymbol{\delta}_{b_a}^b$) and the gyroscopes ($\boldsymbol{\delta}_{b_\omega}^b$) of the INS in the b-frame. Our filter error-state vector becomes:

$$\boldsymbol{\delta} = \begin{pmatrix} \boldsymbol{\delta}_p^{o,T} & \boldsymbol{\delta}_v^{o,T} & \boldsymbol{\delta}_\psi^{o,T} & \boldsymbol{\delta}_{b_a}^{b,T} & \boldsymbol{\delta}_{b_\omega}^{b,T} \end{pmatrix}^T \,. \tag{12}$$

The uncertainties in the error propagation for translation and rotation are modeled as additive zero-mean, white Gaussian noise (AWGN). The acceleration and gyroscope biases are modeled as random walk processes driven by AWGN. The noise vector $\boldsymbol{n}_s$ has the spectral density $\boldsymbol{Q}$.

$$\mathbf{Q} = \text{diag}(\mathbf{Q}_a, \mathbf{Q}_\omega, \mathbf{Q}_{b_a}, \mathbf{Q}_{b_\omega}) \,. \tag{13}$$

where $\text{diag}(\boldsymbol{X_1}, \boldsymbol{X_2})$ is a diagonal matrix with $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ on the diagonal and

$$\mathbf{Q}_s = \mathrm{E}\left[\boldsymbol{n_s}\boldsymbol{n_s}^T\right] \mid s \in \{a, \omega, b_a, b_\omega\} \,. \tag{14}$$

where $\boldsymbol{E}[\boldsymbol{n}]$ is the mean of the stochastic variable $\boldsymbol{n}$. The linearized continuous-time transition can be modeled as [10]:

$$\begin{aligned} \dot{\boldsymbol{\delta}} =& \mathbf{F}\,\boldsymbol{\delta} + \mathbf{G}\,\boldsymbol{n} \\ =& \begin{pmatrix} \mathbf{O}_{3x3} & \mathbf{I}_3 & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} \\ \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & -\lfloor\boldsymbol{a}_{ib}^o\rfloor & -\mathbf{C}_b^o & \mathbf{O}_{3x3} \\ \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & -\mathbf{C}_b^o \\ \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} \\ \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} \end{pmatrix} \boldsymbol{\delta} \\ &+ \begin{pmatrix} \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} \\ \mathbf{C}_b^o & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} \\ \mathbf{O}_{3x3} & \mathbf{C}_b^o & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} \\ \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{I}_3 & \mathbf{O}_{3x3} \\ \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{O}_{3x3} & \mathbf{I}_3 \end{pmatrix} \boldsymbol{n} \end{aligned} \tag{15}$$

where $\lfloor\boldsymbol{a}\rfloor$ is a 3×3 skew matrix such that $\lfloor\boldsymbol{a}\rfloor\,\boldsymbol{b}$ is the cross product of $\boldsymbol{a}$ and $\boldsymbol{b}$. In the following $\boldsymbol{\Phi}_k, \boldsymbol{G}_k$ and $\boldsymbol{Q}_k$ represent the discretizations at time k of $\boldsymbol{F}, \boldsymbol{G}$ and $\boldsymbol{Q}$ respectively.

1) State augmentation: To process relative state measurements in an optimal way we use state augmentation by stochastic cloning [8]. At the arrival of a relative measurement a new relative measurement starts. The end of the preceding relative measurement is the start of the new one. The state at the end of a relative measurement is augmented to the state vector so that it can be referenced to by the following measurement. This concept can be generalized for multiple delta position measurements as shown in [12].

Nevertheless, for highly dynamic systems the time span between the real end of a relative measurement and the arrival of the measurement data can be unacceptably high due to communication and processing time delays. To compensate for these delays and get a non delayed, optimal state estimate the state at the end of a relative measurement is augmented when a hardware trigger signals the real end of the measurement. When the delayed relative measurement arrives it can reference the start and end states of the measurements and correct all states including the augmentations. The filter framework can directly process different time delayed (overlapping relative) measuremtents at the time of arrival. As the state estimation is running at the current time, an optimal estimate of velocity is available at any time. This is of great importance for stable system control.

Furthermore, it is possible, depending on the available processor resources, to hold augmented states in the filter

instead of deleting it directly after the arrival of the measurement data. If a relative state sensor is able to recognize a measurement start or end point that is augmented in the filter, the proposed navigation filter is equivalent to an indirect EKF position SLAM but can compensate for measurement time delays.

For a general representation of filter augmentation we use in the following the term *main state* for the estimated states at the current time and *augmented states* for the rest of the state vector. As mentioned in [12] it is not necessary to augment the whole *main state* for the processing of relative measurements but only the parts a relative measurement refers to. In general, state augmentation and removal for the direct and indirect states at time k can be written as:

$$\bar{\boldsymbol{x}}_k = \boldsymbol{S}_k \boldsymbol{x}_k \qquad (16)$$

$$\bar{\boldsymbol{\delta}}_k = \boldsymbol{S}_k \boldsymbol{\delta}_k \qquad (17)$$

where $\boldsymbol{S}_k$ is a state selection matrix with dimension $(n + a_k + a) \times (n + a_k)$ with n as the size of the *main state* vector, a_k as the number of initially *augmented states* at time k and a as the number of states to augment to or remove from the state vector.

To augment a part of the *main state* to the state vector in between the *main state* and the already *augmented states*, $\boldsymbol{S}_k$ can be written as:

$$\boldsymbol{S}_k = \begin{pmatrix} \boldsymbol{I}_{n \times n} & \boldsymbol{0}_{n \times a_k} \\ \bar{\boldsymbol{I}}_{a \times n} & \boldsymbol{0}_{a \times a_k} \\ \boldsymbol{0}_{a_k \times n} & \boldsymbol{I}_{a_k \times a_k} \end{pmatrix} \qquad (18)$$

where $\boldsymbol{I}_{i \times i}$ is the $i \times i$ identity matrix, $\bar{\boldsymbol{I}}_{i \times j}$ is an identity matrix containing only the rows that correspond to states that should be augmented and and $\boldsymbol{0}_{a_k \times n}$ is the $a_k \times n$ zero matrix.

To remove an augmentation from the state vector, $\boldsymbol{S}_k$ can be written as:

$$\boldsymbol{S}_k = \bar{\boldsymbol{I}}_{(n + a_k + a) \times (n + a_k)} \qquad (19)$$

where a is negative and $\bar{\boldsymbol{I}}$ is an identity matrix of size $(n + a_k) \times (n + a_k)$ with the rows removed that correspond to a state that should be removed.

With this notation the augmented/de-augmented state covariance matrix can be written as:

$$\boldsymbol{E}[\bar{\boldsymbol{\delta}}_k \bar{\boldsymbol{\delta}}_k^T] = \boldsymbol{E}[\boldsymbol{S}_k \boldsymbol{\delta}_k \boldsymbol{\delta}_k^T \boldsymbol{S}_k^T] \qquad (20)$$

2) Prediction: As we have an error state space filter representation only the filter covariance is involved in the prediction step. The augmented error propagation matrix $(\boldsymbol{\Phi}_{aug,k})$ and the noise propagation matrix $(\boldsymbol{G}_{aug,k})$ are defined as:

$$\boldsymbol{\Phi}_{aug,k} = \text{diag}(\boldsymbol{\Phi}_k, \boldsymbol{I}_{a_k^+ \times a_k^+}) \qquad (21)$$

$$\boldsymbol{G}_{aug,k} = \begin{pmatrix} \boldsymbol{G}_k \\ \boldsymbol{0}_{a_k^+ \times n} \end{pmatrix} \qquad (22)$$

where a_k^+ is the number of *augmented states* at the time of prediction k. The filter covariance prediction can be realized by the standard Kalman Filter prediction step:

$$\boldsymbol{P}_{k+1}^- = \boldsymbol{\Phi}_{aug,k} \boldsymbol{P}_k^+ \boldsymbol{\Phi}_{aug,k}^T + \boldsymbol{G}_{aug,k} \boldsymbol{Q}_k \boldsymbol{G}_{aug,k}^T \qquad (23)$$

where $\boldsymbol{P}_{k+1}$ is the a priori error state covariance at time step $k+1$ and $\boldsymbol{P}_k^+$ the a posteriori covariance matrix at time step k.

The special form of $\boldsymbol{\Phi}_{aug,k}$ and $\boldsymbol{G}_{aug,k}$ can be exploited in the filter implementation to get a prediction step with a complexity rising only linearly with the number of *augmented states*.

3) Update: The augmented filter update is realized as standard EKF update:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \qquad (24)$$

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_k^- \qquad (25)$$

$$\boldsymbol{\delta} = \boldsymbol{K}_k \boldsymbol{y}_k \qquad (26)$$

where $\boldsymbol{K}_k$ is the Kalman Filter gain, $\boldsymbol{H}_k$ the measurement matrix, $\boldsymbol{R}_k$ the measurement noise covariance matrix and $\boldsymbol{y}_k$ the measurement residual.

The measurement matrix $\boldsymbol{H}_k$ is of dimension $m \times (n + a_k^+)$ where m is the number of sensor measurements. The first n colums of the $\boldsymbol{H}_k$ matrix correspond to the *main state*. For time delayed, relative measurements the colums of $\boldsymbol{H}_k$ corresponding to the referenced *augmented states* are filled with the relative measurement matrices.

The absolute height measurement is assumed to be not time delayed. The measurement equation is given by:

$$\tilde{z} - \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}_{ob}^o = - \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \boldsymbol{\delta}_{\boldsymbol{p}} + n_{\tilde{z}} \qquad (27)$$

where $\tilde{z}$ is the height measurement in the o-frame and $n_{\tilde{z}}$ the AWGN of the measurement with variance $R = \boldsymbol{E}[n_{\tilde{z}} n_{\tilde{z}}^T]$.

We use a pseudo measurement to exploit the gravity vector for roll and pitch stabilization [10]. The measurement equation is given by:

$$\tilde{\boldsymbol{a}}_{ib}^b + \boldsymbol{C}_b^{o,T} \boldsymbol{g}^o = \boldsymbol{C}_b^{o,T} \begin{pmatrix} 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{\delta}_{\boldsymbol{\psi}}^o + \boldsymbol{n}_{\tilde{\boldsymbol{a}}} \qquad (28)$$

where $\boldsymbol{g}^o = (0\ 0\ g)^T$ is the gravity vector in the o-frame and $\boldsymbol{n}_{\tilde{\boldsymbol{a}}}$ the AWGN of the measurement with variance $R = \boldsymbol{E}[\boldsymbol{n}_{\tilde{\boldsymbol{a}}} \boldsymbol{n}_{\tilde{\boldsymbol{a}}}^T]$.

A general odometry sensor provides noisy position and orientation changes between two points in time T_1 and T_2.

$$\Delta \tilde{\boldsymbol{x}}_{T1}^{T2} = \begin{pmatrix} \tilde{\boldsymbol{p}}_{T2}^o - \tilde{\boldsymbol{p}}_{T1}^o \\ \tilde{\boldsymbol{q}}_{b,T1}^{o,-1} \tilde{\boldsymbol{q}}_{b,T2}^o \end{pmatrix} = \begin{pmatrix} \Delta \tilde{\boldsymbol{p}}_{T1,T2}^o \\ \Delta \tilde{\boldsymbol{q}}_{T2}^{T1} \end{pmatrix} + \boldsymbol{n}_{\Delta \tilde{\boldsymbol{x}}} \qquad (29)$$

where $\boldsymbol{n}_{\Delta \tilde{\boldsymbol{x}}}$ is a AWGN random variable modeling the sensor noise. Position and orientation states at time Tx are augmented to process the measurement. The augmented substates can be written as:

$$\boldsymbol{\delta}_{aug,Tx} = \begin{pmatrix} \boldsymbol{\delta}_{\boldsymbol{p},Tx}^{o,T} & \boldsymbol{\delta}_{\boldsymbol{\psi},Tx}^{o,T} \end{pmatrix}^T \qquad (30)$$

$$\boldsymbol{x}_{aug,Tx} = \begin{pmatrix} \boldsymbol{p}_{Tx}^{o,T} & \boldsymbol{q}_{b,Tx}^{o,T} \end{pmatrix}^T \qquad (31)$$

The estimated delta pose between $T1$ and $T2$ is:

$$\Delta \boldsymbol{x}_{T1}^{T2} = \begin{pmatrix} \boldsymbol{p}_{T2}^{o} - \boldsymbol{p}_{T1}^{o} \\ \boldsymbol{q}_{b,T1}^{o,-1} \boldsymbol{q}_{b,T2}^{o} \end{pmatrix} = \begin{pmatrix} \Delta \boldsymbol{p}_{T1,T2}^{o} \\ \Delta \boldsymbol{q}_{T2}^{T1} \end{pmatrix} \quad (32)$$

As in [13], the measurement equation can be derived as:

$$\Delta \tilde{\boldsymbol{x}}_{T1}^{T2} \ominus \Delta \boldsymbol{x}_{T1}^{T2} = \begin{pmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_{aug,T1} \\ \boldsymbol{\delta}_{aug,T2} \end{pmatrix} + \boldsymbol{n}_{\Delta \tilde{\boldsymbol{x}}} \quad (33)$$

with the measurement submatrices as:

$$\boldsymbol{H}_1 = \begin{pmatrix} \boldsymbol{C}_{T1}^{o,T} & \boldsymbol{C}_{T1}^{o,T} \lfloor \Delta \boldsymbol{p}_{T1,T2}^{o} \rfloor \\ \boldsymbol{0}_{3 \times 3} & \boldsymbol{C}_{T1}^{o,T} \end{pmatrix}$$

$$\boldsymbol{H}_2 = \begin{pmatrix} -\boldsymbol{C}_{T1}^{o,T} & \boldsymbol{0}_{3 \times 3} \\ \boldsymbol{0}_{3 \times 3} & -\boldsymbol{C}_{T1}^{o,T} \end{pmatrix}$$

and the definition of $\ominus$ as:

$$\Delta \tilde{\boldsymbol{x}}_{T1}^{T2} \ominus \Delta \boldsymbol{x}_{T1}^{T2} =$$

$$\begin{pmatrix} \Delta \tilde{\boldsymbol{p}}_{T1,T2}^{o} - \Delta \boldsymbol{p}_{T1,T2}^{o} \\ \left(\mathrm{diag}(2\ 2\ 2) \quad \boldsymbol{0}_{3 \times 1} \right) \frac{\Delta \tilde{\boldsymbol{q}}_{T2}^{T1} \Delta \boldsymbol{q}_{T2}^{T1,-1}}{\| \Delta \tilde{\boldsymbol{q}}_{T2}^{T1} \Delta \boldsymbol{q}_{T2}^{T1,-1} \|} \end{pmatrix}^T$$

which is a substraction of the position part and for the attitude part the delta rotation between the two delta orientations, approximated by the error angle vector.

The measurement noise $\boldsymbol{n}_{\Delta \tilde{\boldsymbol{x}}}$ is again characterized by its covariance matrix $\boldsymbol{R}_{\Delta \tilde{\boldsymbol{x}}} = \boldsymbol{E}[\boldsymbol{n}_{\Delta \tilde{\boldsymbol{x}}} \boldsymbol{n}_{\Delta \tilde{\boldsymbol{x}}}^T]$ calculated by the odometry algorithm.

III. SIMULATION

The performance of the implemented filter is demonstrated in a simulation. We modeled our sensor system shown in Figure 2.

A. Sensor simulation

We implemented an IMU simulation according to [14], simulating a sensor value with additive white gaussian noise (AWGN), a varying bias driven by AWGN and a constant bias. The standard deviation for the angular and velocity random walk driving AWGN can be found from the Allan deviation (AD) plot of the simulated reference IMU (AD16367):

$$\sigma_{RW} = \frac{RW}{\delta t} \quad (34)$$

where RW is the AD plot value at $\tau = 1$ and δt the simulated sensor sampling time. The standard deviation of the bias instability driving AWGN can be found as:

$$\sigma_{BS} = \sqrt{\frac{\delta t}{\tau_{BS}}} BS \quad (35)$$

where BS is the minimum in the AD plot with corresponding time value τ_{BS}. The corresponding noise values from the data sheet are listed in Table (I).

The simulated stereo odometry system calculates the delta position and orientation quaternion between two points in time from a simulated trajectory disturbed by AWGN. The base noise parameters (Table I) are varied by a factor of 100 to simulate periods with a low number of vision features. We limited the odometry system to hold only one key frame

Parameter	Value
IMU Gyroscopes RW	5.2e-4 rad/s
IMU Gyroscopes BS at $\tau_{bs} = 100s$	2.1e-4 rad/s
IMU Accelerometers RW	3.5e-3 m/s²
IMU Accelerometers BS at $\tau_{bs} = 30s$	2.0e-3 m/s²
Stereo odometry position noise σ_{pos}	0.01 m
Stereo odometry attitude noise σ_{att}	0.02 rad
Stereo odometry key frame hold	1 s

TABLE I: Sensor simulation parameters.

at a time. During this time the delta position measurement refers to the same position. The simulated data including the measurement covariance is delayed by a variable measurement delay. At the beginning and the end of a measurement a start and stop trigger signal is generated.

Our height sensor is modeled according to Equation (27)

B. Results

We generated a trajectory flown by a simulated quadrotor model starting with a flip around the roll axis (at $t = 12$s) followed by a low dynamic flight (from $t = 16$s to $t = 150$s) and finished by a highly dynamic flight (from $t = 150$s to $t = 300$s). The flight path and the corresponding estimates for a typical parameter set of our onboard stereo odometry system ($f = 3$Hz and $d = 320$ms) is shown in Figure 3. During the flip we measure accelerations of about $3g$, in the fast dynamic phase we have $1g$ peaks measured in the horizontal plane. Roll and pitch angles go up to $50°$. The velocity reaches about 4m/s.

Figure 4 shows the estimation errors for position and velocity with their corresponding estimated 3σ bounds. The covariances show peaks on the x and y axes for the stereo odometry drop outs. The total position error rises slowly whereas the corresponding velocity error is bound to 0.05m/s. The absolute height measurement limits the z position and velocity errors.

We varied the frequency of the stereo odometry measurement from $f = 15$Hz to $f = 1$Hz as well as the delays of the measurement arrival from $d = \frac{1}{f} - dt$ to $d = dt$ with a sampling time of $dt = 5$ms. For each dataset 20 Monte Carlo simulations were performed. For control applications we are especially interested in the velocity estimate in the body frame. In the absence of an absolute measurement of the yaw angle we transformed the ground truth velocities from the simulated world frame to the estimated world frame. In this way the velocities are directly comparable. The dependency of the mean of the root mean square errors (RMSE) over all runs for position and velocity is shown in Figures 5 and 6.

The velocity RMSE plot shows that the accuracy of the velocity estimate depends strongly on the frequency of the measurement update. With the increase of information coming with a higher frequency the filter can estimate velocity errors more accurately.

Furthermore, the velocity errors rise linearly with time delays. Velocities are propagated by the SDA algorithm and only indirectly measured by position updates. With measurement time delays, the velocity error rises from the end time of a measurement until its arrival. The smaller the
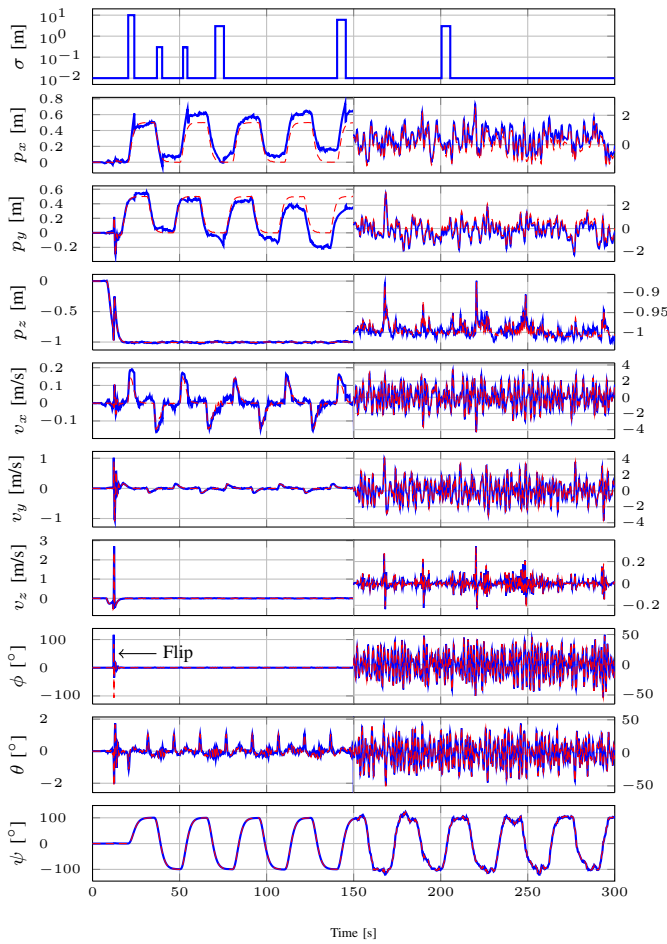
Fig. 3: Simulated flight with stereo odometry parameters $f = 3$Hz, $d = 320$ms and key frame hold of 1s. Plots from top to bottom: Standard deviation of stereo odometry position measurement; Reference position ($p_{x,y,z}$) in red, estimated position in blue; Reference velocity ($v_{x,y,z}$) in red, estimated velocity in blue; Reference angles (roll,pitch,yaw) in red, estimated angles in blue. The trajectory includes a flip ($t = 12s$), a low dynamic passage ($t = 12s$ to $t = 150s$) and a highly dynamic passage ($t = 150s$ to $t = 300s$) with velocities of up to 4m/s and accelerations of up to $3g$ resulting from roll and pitch angles of up to $50°$.

Fig. 4: Simulated flight with stereo odometry parameters $f = 3$Hz, $d = 320$ms and key frame hold of 1s. Plots from top to bottom: Standard deviation of stereo odometry position measurement; Errors for position ($e_{p,x,y,z}$) and velocity ($v_{e,x,y,z}$) in red and the corresponding estimated 3σ bounds in red. The estimate covariances rise during phases of bad vision measurements. The x,y position error rises slowly, the z position error is bound by the absolute hight measurement. The velocity errors are bound by the position measurements.

delay, the smaller the time the velocity error can rise before an indirect update arrives.

Again for the position RMSE, the error rises with lower measurement updates. The position RMSE is almost constant for varying measurement delays. In contrast to velocities, position is measured directly. The error of the measurement is small compared to the propagated error from double integrating accelerations. The constant key frame hold time of 1s has the effect of an absolute position measurement for the period of the key frame. Therefore, the effect of time delays on the position estimate is small.
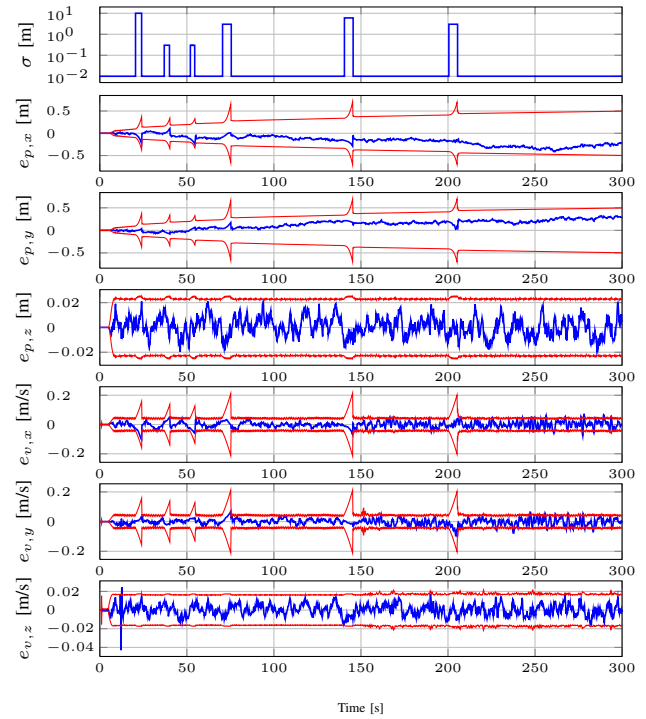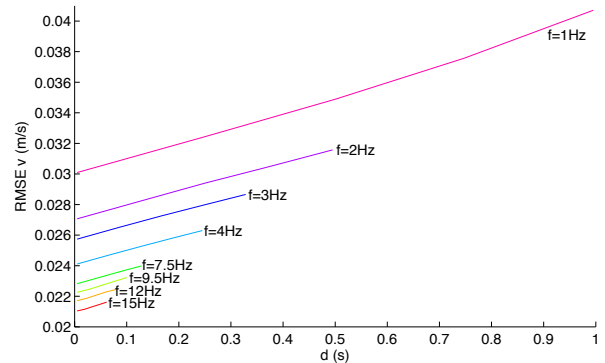

Fig. 5: Root mean square error of velocity estimate using a general odometry sensor with varying frequencies and measurement delays (mean of 20 Monte Carlo simulations).
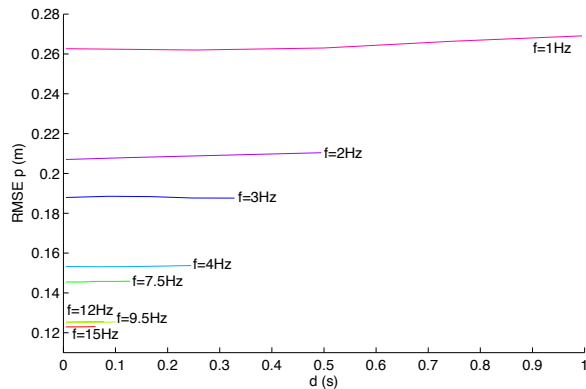
Fig. 6: Root mean square error of position estimate using a general odometry sensor with varying frequencies and measurement delays (mean of 20 Monte Carlo simulations).

IV. Experiments

We demonstrate the estimation system on a quadrotor by flying from inside a house through a window to the outside. A video of the experiment can be found at [15]. Additional to an IMU, the quadrotor is equipped with stereo cameras connected to a Atom processor board (1.6GHz) running a stereo key frame odometry algorithm [12] with a frequency of approximately 3Hz. A key frame is hold as long as it has enough features in common with the current image. Otherwise, the penultimate image is used as new key frame. With this strategy, the system can hover without position drift. In the current baseline configuration the cameras are too close to the ground to have enough image overlap while the quadrotor is still very close on the ground. Therefore, we use a Hokuyo UTM-30LX laser range finder as alternative odometry sensor. The laser measurements are projected onto the ground plane. We run a ICP based laser scan matcher [16] on the projected range measurements to get a delta position and yaw measurement. Some laser beams are reflected to the ground by mirrors to measure height. The system provides 2 Gumstix processor boards (OMAP3530, 720MHz) for sensor fusion, control and laser data processing.

The ground plane projection of laser beams is only valid if the surrounding objects have vertical surfaces which is usually only the case in artificial environments. In other situations (outdoors) the calculated delta movement and its corresponding measurement covariance is invalid and must be ignored. We observed a strong rise in the measurement covariance in these situations. Therefore, we switch between stereo and laser odometry, depending on the measurement covariance.

An external vision tracking system suitable for outdoor environments was not available and GPS is too inaccurate in this case, therefore figure 7 shows the commanded trajectory and the corresponding position estimate. The state estimates for position and velocity as well as laser and stereo odometry covariances are shown in Figure 8.
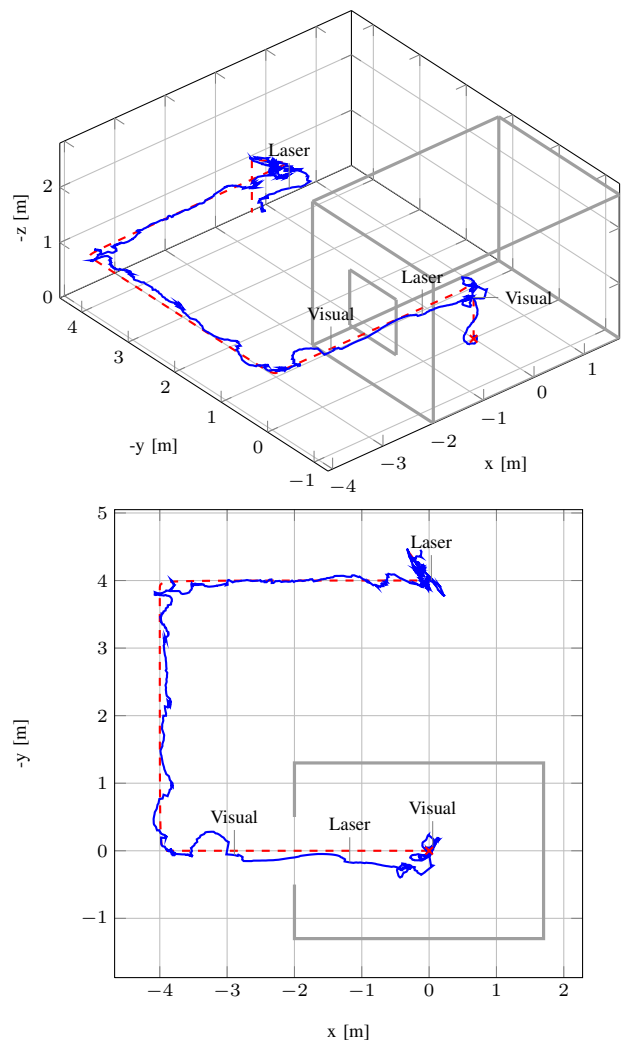
While standing on the ground, only laser odometry is

Fig. 7: Flown path estimated in the experiment. The dashed red line shows the reference trajectory, while the solid line shows the estimated path. The house outline is shown in gray. Locations where switching between visual and laser odometry occurs are also indicated. A video of the experiment can be found at [15].

available. Therefore, the quadrotor starts inside the house using laser odometry. At a height of 70cm, stereo odometry becomes available with a smaller measurement noise than that of the laser odometry. The quadrotor hovers using visual odometry.

When the quadrotor starts moving towards the window, the measurement noise of the stereo odometry rises due to motion blur caused by weak lighting conditions inside the house. Laser odometry is used for the flight through the window. Outside the house, stereo odometry measurements become significantly better than laser odometry measurements. The rest of the flight is conducted by stereo odometry. In the landing phase, stereo odometry becomes unavailable again and laser odometry is used for stabilization.
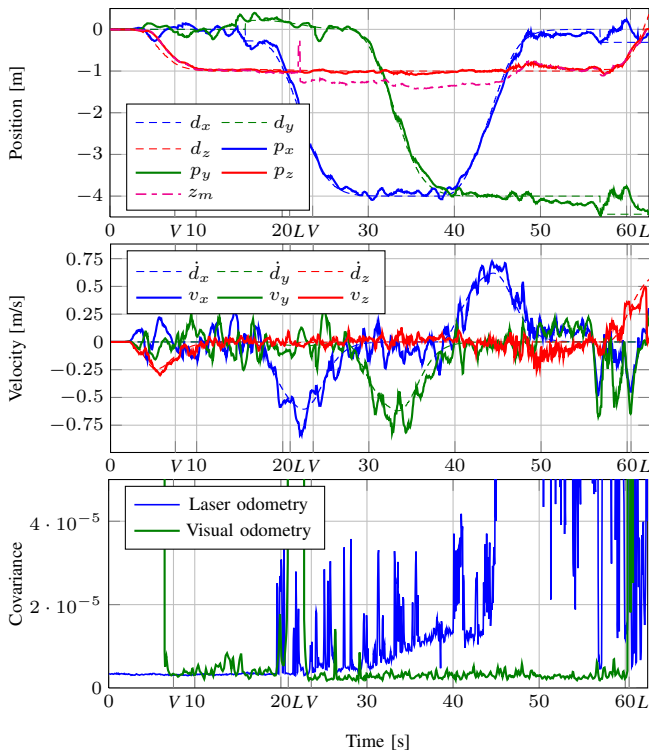
Fig. 8: Flight from inside the house out through the window, located at $x = 2$m. The upper plot shows the reference position (x_d, y_d, z_d), estimated position ($\boldsymbol{p}$) and raw laser height measurement ($\tilde{z}$). The middle plot shows the reference velocity (v_{xd}, v_{yd},v_{zd}) and estimated velocity ($\boldsymbol{v}$). The bottom plot shows the magnitude of laser and visual odometry covariances. Also shown on the time axis are indicators when the system has switched to visual (V) or laser (L) odometry.

V. CONCLUSION

In this contribution we introduced an efficient and robust EKF based INS system. With the proposed state augmentation mechanism, measurements of varying time delayed odometry sensors with key frame support, like a visual odometry system, can be efficiently processed. In this way an accurate and stable estimation of velocities is achieved, especially important for the stabilization of inherently unstable, highly dynamic systems like flying robots. With key frames considered in the filter algorithm a locally drift-free navigation is possible. The number of states augmented to the filter can be increased to result in an INS EKF position SLAM system considering long measurement time delays.

We simulated a quadrotor flight with a fast and slow dynamic trajectory including a flip to show the effect of measurement frequency and time delays. We implemented an odometry sensor holding a key frame for 1s to generate state estimates in 800 Monte Carlo simulations. It was shown that rising time delays result in a linear increase of velocity errors while there is almost no effect on position errors. The measurement frequency has a strong effect on velocity and position errors.

Finally, we tested the system conducting a flight, starting

inside a house and flying out through the window.

We will use the state estimation for highly dynamic, autonomous flight in unknown environments without the aid of external tracking. The size of the drift-free navigated area is limited by the number of key frames. Since the number of onboard-processable key frames is limited due to processor resources, an optimal strategy for dropping unused key frames out of the filter must be found.

REFERENCES

[1] M. George and S. Sukkarieh, "Tightly coupled ins/gps with bias estimation for uav applications," in *Australiasian Conference on Robotics and Automation (ACRA) 2005*, 2005. [Online]. Available: http://prism2.mem.drexel.edu/~vefa/research/HeliExtLoadStabil/relevantPapers/imu/george.pdf

[2] D. Mellinger, N. Michael, and V. Kumar, "Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors," in *Int Symposium on Experimental Robotics*, 2010, pp. 27–27. [Online]. Available: http://www.seas.upenn.edu/~dmel/mellingerISER2010.pdf

[3] B. Herisse, F.-X. Russotto, T. Hamel, and R. Mahony, "Hovering flight and vertical landing control of a VTOL Unmanned Aerial Vehicle using optical flow," *2008 IEEERSJ IROS*, pp. 801–806, 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4650731

[4] D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart, "Vision Based Position Control for MAVs Using One Single Circular Landmark," *Journal of Intelligent Robotic Systems*, vol. 61, no. 1-4, pp. 495–512, 2010. [Online]. Available: http://www.springerlink.com/index/10.1007/s10846-010-9494-8

[5] M. Achtelik, S. Weiss, and R. Siegwart, "Onboard IMU and monocular vision based control for MAVs in unknown in-and outdoor environments," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3056–3063. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5980343

[6] L. Meier and P. Tanskanen, "Pixhawk: A system for autonomous flight using onboard computer vision," *(ICRA), 2011 IEEE*, pp. 2992–2997, 2011. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5980229

[7] T. Cheviron, T. Hamel, R. Mahony, and G. Baldwin, "Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of UAV," *ICRA, 2007*, vol. 92223, no. April 2007, pp. 10–14, 2007. [Online]. Available: http://scholar.google.co.kr/scholar?q=robust+nonlinear+fusion+of+inertial+and+visual+data+for+position&hl=ko&lr=#0

[8] S. Roumeliotis and J. Burdick, "Stochastic cloning: a generalized framework for processing relative state measurements," in *ICRA, 2002*, vol. 2, no. 1. Ieee, 2002, pp. 1788–1795. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1014801

[9] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *PAMI, IEEE Transactions on*, vol. 29, no. 6, pp. 1052 –1067, June 2007.

[10] J. Wendel, *Integrierte Navigationssysteme. Sensordatenfusion, GPS und Inertiale Navigation: Sensordaten, GPS und Inertiale Navigation*. Oldenbourg, 2007.

[11] P. S. Maybeck, *Stochastic models, estimation, and control*, ser. Mathematics in Science and Engineering, R. Bellman, Ed. Academic Press, 1979, vol. 1, no. 2.

[12] A. Stelzer, H. Hirschmuller, and M. Görner, "Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain," *The International Journal of Robotics Research*, Feb. 2012. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364911435161

[13] S. I. Roumeliotis, "A Kalman filter for processing 3-D relative pose measurements," California Institute of Technology, Tech. Rep., 2002. [Online]. Available: http://robotics.caltech.edu/~stergios/tech_reports/relative_3d_kf.pdf

[14] O. Woodman, "An introduction to inertial navigation," University of Cambridge, Cambridge, Tech. Rep. 696, 2007.

[15] http://www.dlr.de/rm/desktopdefault.aspx/tabid-7903.

[16] A. Censi, "An ICP variant using a point-to-line metric," in *2008 IEEE International Conference on Robotics and Automation*. Ieee, 2008, pp. 19–25. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4543181