<http://kth.diva-portal.org>

This is an author produced version of a paper published in *2011 IEEE International Conference on Robotics and Automation (ICRA)*.

This paper has been peer-reviewed but does not include the final publisher proof-corrections or proceedings pagination.

© 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Citation for the published paper:

Dave Zachariah and Magnus Jansson.

Self-motion and wind velocity estimation for small-scale UAVs

2011 IEEE International Conference on Robotics and Automation (ICRA).

Access to the published version may require subscription.

Published with permission from: IEEE

Self-motion and wind velocity estimation for small-scale UAVs

Dave Zachariah and Magnus Jansson

Abstract—For small-scale Unmanned Aerial Vehicles (UAV) to operate indoor, in urban canyons or other scenarios where signals from global navigation satellite systems are denied or impaired, alternative estimation and control strategies must be applied. In this paper a system is proposed that estimates the self-motion and wind velocity by fusing information from airspeed sensors, an inertial measurement unit (IMU) and a monocular camera. Such estimates can be used in control systems for managing wind disturbances or chemical plume-based tracking strategies. Simulation results indicate that while the inertial dead-reckoning process is subject to drift, the system is capable of separating the self-motion and wind velocity from the airspeed information.

I. INTRODUCTION

The control system of a small-scale UAV requires information of its self-motion as well as of the changing external conditions. When operating in GPS-denied or impaired environments, such as indoor or urban canyons, the UAV has to rely on dead-reckoning systems and/or natural landmarks. Furthermore, changing external conditions may include wind disturbances which affect the performance of autonomous flight control and efficiency of path planning and following [1], [2].

Biological autonomous control systems, in e.g. flying insects, are capable of navigating without absolute position estimates in a fixed Euclidean space nor relying on pre-installed infrastructure but on visual, inertial and other sensors that provide information about their self-motion. In addition, they are capable of finding sources that emit chemical plumes through the air, using the direction of the wind [3]. Such control strategies for small-scale UAVs, cf. [4]–[6], may find useful applications for tracking chemical leaks, fires and other detectable sources.

From a control point of view it is of interest to estimate both the self-motion of the UAV and the direction and speed of the wind. A set of three orthogonal anemometers measure the speed of the air that flows past them in the sensor coordinate frame $\{a\}$, which is the difference $\mathbf{d}^a = \mathbf{v}^a - \mathbf{w}^a \in \mathbb{R}^3$ between the velocity of the UAV and the wind. The estimation problem consists of separating these two signals and resolving them in some common navigation frame $\{n\}$ by fusing information from other sensors, exemplified in Figure 1. The design considerations for placing anemometers on different UAV:s are not dealt with here.

An inertial measurement unit provides estimates of $\mathbf{v}^n$ which rapidly deteriorate through double integration of sen-

Fig. 1. Measuring airspeed in $\{a\}$. Camera and navigation coordinate frames, denoted as $\{c\}$ and $\{n\}$, respectively.

sor noise and thus is insufficient. In [7], a system using a downward-directed monocular camera is proposed and tested in simulations. The images are assumed to be processed by an optical flow algorithm proposed by [8], [9], which extracts an estimated direction vector. The sensor fusion is formulated in a deterministic signal model and solved by a Least-Squares polynomial approximation. The method, however, assumes that the UAV operates parallel over a level ground or a ground modeled by tessellated planes with smoothly varying height. Its robustness and drift properties remain to be explored.

In this paper, assumptions of the geometric structure of the scene are relaxed, exploiting only natural, static points that are detectable using common feature point extraction algorithms [10]–[12]. The proposed system uses a low-cost IMU and forward-directed monocular camera for a visual memory that imposes stochastic, geometric constraints on the estimation problem. Hence given air speed measurements, noisy inertial signals and extracted visual feature points, the estimation of $\mathbf{v}^n$ and $\mathbf{w}^n$ is accomplished by exploiting pairwise epipolar constraints between a current and past image [13]. The method exhibits a certain degree of robustness to feature point mismatches.

Since the system navigates in a horizontal plane perpendicular to the gravitation field, the heading angle error is unobservable and hence subject to slow drift due to integration of gyroscope errors, but is still capable of separating the self-motion and wind velocity from the airspeed data as is indicated in a simulation study below.

II. PROCESS MODEL

The wind and the UAVs navigation states are modeled as a discrete-time process running at the sample rate of the inertial sensors, which provide high-rate measures of the specific force and angular rate, in the order of 100 Hz, in an inertial frame $\{b\}$, here aligned with $\{a\}$ for simplicity.

The authors are with the ACCESS Linnaeus Centre, School of Electrical Engineering, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden. {dave.zachariah, magnus.jansson}@ee.kth.se

TABLE I
NOTATION.

Variables	Description
$\mathbf{v}$	self-motion
$\mathbf{w}$	wind velocity
$\mathbf{d}$	anemometer measurements
$\bar{\mathbf{y}}$	epipole measurements
ψ	stacked INS errors
$\delta\bar{\rho}$	stacked camera position errors
ξ, Ξ	wind process noise and covariance matrix
η, Σ	INS error process noise and covariance matrix
ν_d, Υ	anemometer noise and covariance matrix
$\nu_{\bar{\mathbf{y}}}, \Pi$	epipole noise and covariance matrix

A. Wind velocity

Rather than taking the perilous path of constructing a local model of the wind field, which is notoriously hard to track [14], a simple, first-order vector-autoregressive (AR) model is adopted,

$$\mathbf{w}_{k+1}^n = \Phi_k \mathbf{w}_k^n + \xi_k \in \mathbb{R}^3, \quad (1)$$

with a process noise covariance matrix Ξ_k . For simplicity, $\Phi_k \equiv \mathbf{I}_3$ and Ξ_k is updated adaptively, as described below.

B. Navigation error states

The Inertial Navigation System (INS) mechanization equations provide estimates of the position $\mathbf{p}^n$, velocity $\mathbf{v}^n$ and attitude θ of the system given input signals from the accelerometers $\hat{\mathbf{f}}^b$ and gyroscopes $\hat{\omega}^b$. However, the solution rapidly deteriorates due to the integration and propagation of sensor noise in low-cost IMUs. In addition, a model of the self-motion of the UAV would have to incorporate effects of the wind on the system. These two facts motivate modeling the errors of the INS solution, rather than $(\mathbf{p}^n, \mathbf{v}^n, \theta)$, and estimating them for periodic corrections. The errors during a short period of time can be modeled in a linear fashion,

$$\begin{aligned} \delta\mathbf{p}_{k+1}^n &= \delta\mathbf{p}_k^n + dt\delta\mathbf{v}_k^n \\ \delta\mathbf{v}_{k+1}^n &= \delta\mathbf{v}_k^n + dt[\hat{\mathbf{R}}_{b,k}^n \hat{\mathbf{f}}_k^b]_{\times} \delta\theta_k \\ &\quad + dt\hat{\mathbf{R}}_{b,k}^n \delta\mathbf{f}_k^b + dt\hat{\mathbf{R}}_{b,k}^n \eta_{\mathbf{f},k} \\ \delta\theta_{k+1} &= \delta\theta_k - dt\hat{\mathbf{R}}_{b,k}^n \delta\omega_k^b - dt\hat{\mathbf{R}}_{b,k}^n \eta_{\omega,k}, \end{aligned} \quad (2)$$

where dt is the sample period, $\hat{\mathbf{R}}_{b,k}^n$ is the current estimate of the attitude in rotation matrix form, $[\cdot]_{\times} : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ is the skew-symmetrix matrix representation of a vector in the cross-product operation, η is stochastic zero-mean, temporally white noise arising from the sensors, and $(\delta\mathbf{f}^b, \delta\omega^b)$ are slow-varying accelerometer and gyroscope sensor biases modeled as,

$$\begin{aligned} \delta\mathbf{f}_{k+1}^b &= \delta\mathbf{f}_k^b + dt\eta_{\delta\mathbf{f},k} \\ \delta\omega_{k+1}^b &= \delta\omega_k^b + dt\eta_{\delta\omega,k}, \end{aligned} \quad (3)$$

respectively. The navigation error states can be written compactly in the linear model,

$$\psi_{k+1} = \Lambda_k \psi_k + \Gamma_k \eta_k \in \mathbb{R}^{15}, \quad (4)$$

where augmented noise covariance matrix Σ is determined by the second-order statistical properties of the sensor noise.

Fig. 2. Correlation diagram of states and measurements.

C. Visual memory states

The current camera position in $\{n\}$ is $\rho = \mathbf{p}^n + \mathbf{R}_b^n \mathbf{p}_c^b$, where $\mathbf{p}_c^b$ is the offset between inertial frame and camera frame. The position error is linearly related to ψ by

$$\delta\rho = \delta\mathbf{p}^n + [\hat{\mathbf{R}}_b^n \mathbf{p}_c^b]_{\times} \delta\theta \in \mathbb{R}^3. \quad (5)$$

Once recorded, a past view i has no dynamics, therefore its error process model is simply $\delta\rho_{k+1}^{(i)} = \delta\rho_k^{(i)}$. As time progresses the visual overlap and correlation between a past view and the current view grow weaker. Hence, for purposes of tractability it is motivated to maintain a limited buffer of P past views, by continuously removing the oldest view states, which effectively constitutes a short-term visual memory that imposes multiple geometric constraints on the estimation problem. The error states are stacked into a single vector $\delta\bar{\rho} \in \mathbb{R}^{3P}$.

III. MEASUREMENT MODEL

The states in the process models are correlated through measurements of the air speed, $\mathbf{d}^a$, and a set of visual epipolar points, $\bar{\mathbf{y}}$, as illustrated in Figure 2.

A. Anemometers

The three-axis wind speed sensors measure the velocity relative to the surrounding air, which can be written in terms of the INS error and wind states,

$$\begin{aligned} \mathbf{d}_k^a &= \phi_k(\psi_k, \mathbf{w}_k^n) + \nu_{\mathbf{d},k} \in \mathbb{R}^3 \\ &= \mathbf{R}_b^a \hat{\mathbf{R}}_{b,k}^b \mathbf{C}(\delta\theta)(\hat{\mathbf{v}}_k^n - \delta\mathbf{v}_k^n - \mathbf{w}_k^n) + \nu_{\mathbf{d},k}, \end{aligned} \quad (6)$$

where $\mathbf{C}(\cdot)$ is a rotation matrix, $\mathbf{R}_b^a$ is the rotation matrix from frame $\{b\}$ to $\{a\}$, $\hat{\mathbf{v}}_k^n$ is the current INS estimate of the velocity, and $\nu_{\mathbf{d}}$ is the quantization noise with a covariance matrix $\Upsilon = \frac{\Delta_{\text{quant}}^2}{12} \times \mathbf{I}_3$.

B. Visual memory and epipolar points

When a new image is obtained, its feature points π_j are extracted and compared with the feature points π_j^i in view i , producing a set of matched pairs $\{(\pi_j, \pi_j^i)\}_j$. Using the rotation quaternion $\mathbf{q}_{c_i}^c$ from camera frame $\{c_i\}$ to current frame $\{c\}$ the points from view i can be ‘rotation unwrapped’ into $\tilde{\pi}_j^i$, producing a set of matched feature points on parallel image planes $\{(\pi_j, \tilde{\pi}_j^i)\}_j$. The rotation $\mathbf{q}_{c_i}^c$ can be obtained using the information provided by the gyroscopes, which simultaneously update a buffer $\Theta_k = [\mathbf{q}_{c_1}^c \cdots \mathbf{q}_{c_P}^c]$, provided the error growth is moderate between frames.

Each pair of matches $(\pi_j, \tilde{\pi}_j^i)$ generates a line and ideally all of them intersect at the epipolar point $\mathbf{y}^{(i)} \in \mathbb{R}^2$ on the

image plane. By the construction of parallel image planes the epipole is the projection of the translation vector $\Delta \mathbf{p}^{(i)} \in \mathbb{R}^3$ between current and past view i . However, due to a host of error sources in the inertial and visual sensors the actual lines intersect pairwise at various points $\mathbf{c}_j$. The epipole is estimated as the weighted average,

$$\mathbf{y}^{(i)} = \sum_l w_l \mathbf{c}_l.$$

The measurement noise is modeled as an additive zero-mean term, $\boldsymbol{\nu}_y^{(i)}$ with a covariance matrix approximated by

$$\boldsymbol{\Pi}^{(i)} = \sum_l w_l (\mathbf{c}_l - \mathbf{y}^{(i)})(\mathbf{c}_l - \mathbf{y}^{(i)})^*,$$

which takes into account the spread of the intersections. The weights w_l can be set as a measure of the reliability of the intersection $\mathbf{c}_l$, e.g. using distance between the points that generate the lines. The covariance approximation builds in a certain degree of robustness to feature point mismatches and outliers. See [13] for a more detailed description of the epipolar measurement process.

Projecting the epipole on the image plane using a pinhole camera model yields the relation,

$$\mathbf{y}^{(i)} = \bar{\mathbf{A}}([0 \ 0 \ 1] \Delta \mathbf{p}^{(i)})^{-1} \Delta \mathbf{p}^{(i)} + \boldsymbol{\nu}_y^{(i)}, \quad (7)$$

where $\bar{\mathbf{A}} \in \mathbb{R}^{2 \times 3}$ is a truncated camera calibration matrix, $\boldsymbol{\nu}_y$ is measurement noise and the translation vector between view i and the current view expressed in $\{c\}$ is given by

$$\begin{aligned} \Delta \mathbf{p}^{(i)} &= \mathbf{R}_b^c \mathbf{R}_n^b (\boldsymbol{\rho}^{(i)} - \boldsymbol{\rho}) \\ &= \mathbf{R}_b^c \hat{\mathbf{R}}_n^b \mathbf{C}(\delta \boldsymbol{\theta}) \\ &\cdot [\hat{\boldsymbol{\rho}}^{(i)} - \hat{\mathbf{p}}^n + \delta \mathbf{p}^n - \delta \boldsymbol{\rho}^{(i)} - \mathbf{C}^\top(\delta \boldsymbol{\theta}) \hat{\mathbf{R}}_b^n \mathbf{p}_c^b]. \end{aligned} \quad (8)$$

The epipole can only be measured in motion which is not parallel to the image plane, and provides no information of the motion component colinear to $\Delta \mathbf{p}^{(i)}$.

Stacking computed epipoles and the corresponding pose error states as vectors $\bar{\mathbf{y}} = \text{vec}([\mathbf{y}^{(1)} \dots \mathbf{y}^{(P)}])$ and $\delta \bar{\boldsymbol{\rho}} = \text{vec}([\delta \boldsymbol{\rho}^{(1)} \dots \delta \boldsymbol{\rho}^{(P)}])$, respectively, yields the compact measurement equation, in terms of the error states,

$$\bar{\mathbf{y}}_k = \boldsymbol{\sigma}_k(\boldsymbol{\psi}_k, \delta \bar{\boldsymbol{\rho}}_k) + \boldsymbol{\nu}_{\bar{\mathbf{y}},k}, \quad (9)$$

where the stacked noise vector $\boldsymbol{\nu}_{\bar{\mathbf{y}},k}$ has a block diagonal covariance matrix, using the direct sum $\boldsymbol{\Pi}_k = \boldsymbol{\Pi}_k^{(1)} \oplus \dots \oplus \boldsymbol{\Pi}_k^{(P)} \in \mathbb{R}^{2P \times 2P}$.

IV. STATE ESTIMATION

The overall estimation framework fuses the information of the sensors as illustrated in Figure 3. Combining all states into a single vector $\mathbf{x}_k \triangleq [(\mathbf{w}_k^n)^\top \ \boldsymbol{\psi}_k^\top \ \delta \bar{\boldsymbol{\rho}}_k^\top]^\top$, the joint process model can be written as,

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{n}_k \in \mathbb{R}^{18+3P}, \quad (10)$$

where $\mathbf{F}_k \triangleq \boldsymbol{\Phi}_k \oplus \boldsymbol{\Lambda}_k \oplus \mathbf{I}_{3P}$,

$$\mathbf{G}_k \triangleq \begin{pmatrix} \mathbf{I}_3 \oplus \boldsymbol{\Gamma}_k \\ \mathbf{0}_{3P \times 15} \end{pmatrix} \in \mathbb{R}^{(18+3P) \times 15},$$

Fig. 3. Estimation framework.

and $\mathbf{n}_k \triangleq [\boldsymbol{\xi}_k^\top \ \boldsymbol{\eta}_k^\top]^\top \in \mathbb{R}^{15}$ is the joint process noise vector with covariance matrix $\mathbf{Q}_k \triangleq \boldsymbol{\Xi}_k \oplus \boldsymbol{\Sigma}$. This sparse structure can be exploited in fast implementations of the signal processor described below.

The combined measurement equation of (6) and (7) is

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \boldsymbol{\nu}_k. \quad (11)$$

Depending on what sensor data is available at time instant k , $\mathbf{h}_k(\cdot)$ is the concatenation of $\boldsymbol{\sigma}_k(\cdot)$ and $\boldsymbol{\phi}_k(\cdot)$, the noise $\boldsymbol{\nu}_k$ of $\boldsymbol{\nu}_{d,k}$ and $\boldsymbol{\nu}_{\bar{\mathbf{y}},k}$, and the covariance matrix $\mathbf{R}_k$ of $\boldsymbol{\Pi}_k$ and $\boldsymbol{\Upsilon}$.

Together (10) and (11) form a state space model with a linear and sparse process model and a nonlinear measurement model. The estimator is restricted to be recursive and linear in the observations. Using the Mean Square Error criterion, the proposed estimator is a hybrid Sigma-Point Kalman filter, with a linear and efficient time update and a measurement update which approximates first and second-order statistics using the Sigma-Point transformation [15], [16]. The entire estimation procedure is summarized in Algorithm 1. Decimation of view error states in the error covariance matrix $\mathbf{S}_k$ is done by simply removing its corresponding entries. Augmentation of a new view error state is done by

$$\mathbf{S}'_k = \begin{pmatrix} \mathbf{S}_k & \mathbf{S}_k \mathbf{T}_k^* \\ \mathbf{T}_k \mathbf{S}_k & \mathbf{T}_k \mathbf{S}_k \mathbf{T}_k^* \end{pmatrix},$$

where $\mathbf{T}_k \triangleq [\mathbf{0}_3 \ \mathbf{I}_3 \ \mathbf{0}_3 \ [\hat{\mathbf{R}}_{b,k}^n \mathbf{p}_c^b]^\times \ \mathbf{0}_{3 \times (6+3P)}]$. In practice the filter is implemented in square-root form which offers numerical stability [17], [18].

The covariance matrix of the wind process noise is adaptively updated using a sliding window of estimated innovations, $\boldsymbol{\Xi}_l = \frac{1}{W}(\mathbf{e}_{d,l} \mathbf{e}_{d,l}^* - \mathbf{e}_{d,l-W} \mathbf{e}_{d,l-W}^*) + \boldsymbol{\Xi}_{l-1}$, where $\mathbf{e}_{d,l} = \hat{\mathbf{w}}_l^n - \hat{\mathbf{w}}_l^{n,-}$ is the difference between filtered and predicted wind estimates. Hence if predictions deteriorate over time, the uncertainty captured by $\boldsymbol{\Xi}_l$ increases.

V. SIMULATIONS

The estimator is tested using simulations, which facilitate comparisons with generated reference wind signals.

Algorithm 1 Estimation using Sigma-Point Kalman Filter, with weights $\{w^c\}$, $\{w^m\}$ and η . Number of states $N = 18 + 3P$.

```

1: Initialize  $\hat{\mathbf{x}}_0^-$ ,  $\mathbf{P}_0^-$ 
2: for  $k = 0, \dots$  do
3:   Update INS equations and  $\Theta_k$ 
4:   if new image  $\exists$  then
5:     Extract feature points  $\{\pi_j\}$ 
6:     Match and rotation unwrap past feature points  $\{\pi_j^i\}$ 
7:     Compute epipoles  $\mathbf{y}^{(i)}$  and  $\Pi_k$  using  $\{(\pi_j, \check{\pi}_j^i)\}_j$ 
8:   end if
9:   if  $\{\mathbf{y}^{(i)}\} \neq \emptyset \vee \mathbf{d}_k^a \exists$  then
10:    %Generate sigma points:
11:     $\mathcal{X}_{l,k}^- = \hat{\mathbf{x}}_k^- \pm \eta \cdot \text{col}[\mathbf{P}_k^{-1/2}]_l$ 
12:    if  $\{\mathbf{y}^{(i)}\} \neq \emptyset$  then
13:      %Generate prediction
14:       $\mathcal{Y}_{l,k}^- = \sigma_k(\mathcal{X}_{l,k}^-)$ ,  $\hat{\mathbf{y}}_k^- = \sum_{l=0}^{2N} w_l^m \mathcal{Y}_{l,k}^-$ 
15:    end if
16:    if  $\mathbf{d}_k^a \exists$  then
17:      %Generate prediction
18:       $\mathcal{D}_{l,k}^- = \phi_k(\mathcal{X}_{l,k}^-)$ ,  $\hat{\mathbf{d}}_k^- = \sum_{l=0}^{2N} w_l^m \mathcal{D}_{l,k}^-$ 
19:    end if
20:    Form augmented vectors  $\mathbf{z}_k, \hat{\mathbf{z}}_k^-, \mathcal{Z}_{l,k}^-$ 
21:    Form noise covariance matrix  $\mathbf{R}_k = \Pi_k \oplus \Upsilon$ 
22:    %Measurement update:
23:     $\mathbf{R}_{e,k} = \sum_{l=0}^{2N} w_l^c (\mathcal{Z}_{l,k}^- - \hat{\mathbf{z}}_k^-)(\mathcal{Z}_{l,k}^- - \hat{\mathbf{z}}_k^-)^* + \mathbf{R}_k$ 
24:     $\mathbf{R}_{\mathbf{x},e,k} = \sum_{l=0}^{2N} w_l^c (\mathcal{X}_{l,k}^- - \hat{\mathbf{x}}_k^-)(\mathcal{Z}_{l,k}^- - \hat{\mathbf{z}}_k^-)^*$ 
25:     $\mathbf{K}_k = \mathbf{R}_{\mathbf{x},e,k} \mathbf{R}_{e,k}^{-1}$ 
26:     $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k^-)$ 
27:     $\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^*$ 
28:    Use error states of  $\hat{\mathbf{x}}_k$  for correction
29:    Reset error states of  $\hat{\mathbf{x}}_k$ 
30:     $\mathbf{s}_k := \hat{\mathbf{x}}_k$ ,  $\mathbf{S}_k := \mathbf{P}_k$ 
31:  else
32:     $\mathbf{s}_k := \hat{\mathbf{x}}_k^-$ ,  $\mathbf{S}_k := \mathbf{P}_k^-$ 
33:  end if
34:  if new image  $\exists$  then
35:    Record view position  $\hat{\rho}^{(i)}$ 
36:    Decimate and augment view orientations in  $\Theta_k$ 
37:    Decimate and augment view error states in  $\mathbf{s}_k$ 
38:    Update  $\mathbf{S}_k$  correspondingly
39:  end if
40:  %Time update:
41:   $\hat{\mathbf{x}}_{k+1}^- = \mathbf{F}_k \mathbf{s}_k$ 
42:   $\mathbf{P}_{k+1}^- = \mathbf{F}_k \mathbf{S}_k \mathbf{F}_k^* + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^*$ 
43: end for

```

A. Wind generation

The wind was generated according to the vector-AR(L) process,

$$\mathbf{w}_{k+1}^n = \sum_{l=0}^{L-1} \mathbf{A}_l^s \mathbf{w}_{k-l}^n + \boldsymbol{\xi}_k,$$

with process noise covariance matrix $\boldsymbol{\Xi}^s$, where the model parameters $(\{\mathbf{A}_l^s\}_{l=0}^{L-1}, \boldsymbol{\Xi}^s) \in \mathcal{A}$ are chosen randomly at every time instant k according to a Markov-chain with states $s \in \{1, \dots, |\mathcal{A}|\}$ and transition probabilities $p_{i|j} \equiv \lambda_p, i \neq j$. This allows for a non-stationary signal with erratic changes in its characteristics, e.g. gusts.

Here the number of states were set to $|\mathcal{A}| = 4$ and the maximum lag $L = 3$. The parameters for the x and y -

Fig. 4. A realization of wind process $\mathbf{w}^n$ and hidden state s .

Fig. 5. 66 second trajectory and 200 feature points.

dimensions were set by fitting their values to different, short segments of wind data recorded with a two-axis anemometer at 60 Hz, using the covariance method [19]. On the basis of the fact that at low altitudes wind flows in the z -dimension are small, the parameters were set to small and arbitrary values. The state transition probability was set to $\lambda_p = 6 \times 10^{-4}$ for a wind process sampled at 50 Hz. An example is given in Figure 4.

B. Trajectory and signals

A user-defined trajectory is set in frame $\{n\}$, as illustrated in Figure 5 with an average speed of 1 m/s, i.e. of the same order as the wind. Trajectories were chosen to include turns; if the camera moves head along a linear path the epipolar measurements provide no information about the velocity along this dimension, which becomes unobservable and its estimate diverges. The attitude, and hence angular rate, of the UAV is determined by a simple model which aligns it to the velocity vector. Continuous acceleration is ensured using cubic spline interpolation to yield ideal IMU signals sampled at 100 Hz. Biases and stochastic noise are added as constants and additive white Gaussian noise processes for each sample, with parameters given in Table II. With these settings the error of the INS velocity estimates after 33 seconds (half-way) of the given trajectory is about $[6 \cdot 10^{-2}, 6 \cdot 10^{-1}]^\top$ m/s for $[\hat{v}_x^n, \hat{v}_y^n, \hat{v}_z^n]$.

Static feature points are added to the scene and the camera

TABLE II
IMU SENSOR NOISE PARAMETERS.

	Bias	Std.
Acc. [m/s ²]	2×10^{-2}	6×10^{-3}
Gyro. [rad./s]	2×10^{-3}	3×10^{-3}

is simulated using the pinhole camera model. Visible feature points in $\{n\}$ are projected on the image plane at a rate of 10 Hz and rounded to integers as coordinates on a grid of 752×480 pixels. The origins of the camera and inertial frames were offset by 5×10^{-2} m. The camera calibration matrix containing the intrinsic camera parameters with focal length $f = 5 \times 10^{-3}$ m and pixel width $\Delta_x = \Delta_y = 6 \times 10^{-6}$ m.

The output from the air speed sensors were computed according to the model, with quantization step size $\Delta_{\text{quant}} = 0.1$ m/s.

C. Estimation setup

In relative navigation the initial position and heading estimates are correct by definition, but the roll ϕ and pitch θ estimates contain errors. The standard deviations in the filter were set to 1° .

The maximum length of the view buffer was set to $P_{\text{max}} = 10$ frames, corresponding to a visual memory of 1 second and the window size used for Ξ_k was $W = 30$. The weights in the Sigma-Point filter are set as $w_l^c = w_l^m = \frac{1}{2(N+\lambda)}$ for $l = 1, \dots, 2N$. For $l = 0$, $w_0^c = \frac{\lambda}{N+\lambda} + (1 - \alpha^2 + \beta)$ and $w_0^m = \frac{\lambda}{N+\lambda}$. Here $\lambda \equiv \alpha^2(N + \kappa) - N$, with parameters $\alpha = 0.1$, $\beta = 2$ and $\kappa = 0$ that also determine the spread of the sigma points through the weight $\eta \equiv \sqrt{N + \lambda}$.

Since image processing is prone to outliers the system must mitigate such effects. The weights $\{w_j\}$ used for computing the epipole are set according to a threshold on the minimum distance between the points that generate the pair j of intersecting lines $\gamma_j \triangleq \min(\|\pi_k - \check{\pi}_k\|_2, \|\pi_l - \check{\pi}_l\|_2)$. For $\gamma_j < 10$ pixels the weights are set to zero and the remaining ones are equal and sum up to unity. In addition unreliable intersections are rejected completely by rejecting pairs of lines with slopes that differ by less than 30%.

D. Results

The estimation errors are evaluated in 100 Monte Carlo simulations, each generating a realization of the IMU sensor noise processes, for the 66 second trajectory and wind process. The results are shown in Figures 6 and 7, where the average values are given in colored lines and estimated Root Mean Square Error (RMSE) bounds in dashed lines. Figure 8 shows the same statistics overlayed but in an alternative parametrization of the vectors $\mathbf{w}^n$ and $\mathbf{v}^n$ in terms of magnitude, and pitch and heading angles. The 3σ -levels, extracted from the filter's error covariance matrix, in dotted blue lines can be understood as a representation of the uncertainty of the filter.

The oscillating velocity errors along the x - and y -dimensions during the trajectory are a consequence of the

Fig. 6. Error statistics for $\hat{\mathbf{w}}^n$.

Fig. 7. Error statistics for $\hat{\mathbf{v}}^n$.

unobservability of heading errors in inertial dead-reckoning in the horizontal plane perpendicular to the gravitation field. The effect is visible in Figure 8, where the vector norms and pitch angles are consistently tracked during the time period but the heading angles are equally biased for both estimates. This is further illustrated in Figure 9, where ψ drifts and the Sigma-Point approximation used by the signal processor is able to capture the second-order error statistics of roll and pitch but not heading. In other words, the estimates of the velocity of the UAV and the wind are subject to the same slow-drifting heading error, but the filter is capable of separating them equally in the airspeed data.

For comparison with the 8 second trajectory simulated in [7], Table III is included which shows the estimated RMSE over the entire trajectory and the results are of the same order of magnitude.

VI. CONCLUSIONS

We have presented a self-motion and wind velocity estimation procedure for small-scale UAVs, which fuses visual and inertial information with airspeed information. While the system is subject to the inherent limitations of inertial dead-reckoning, which causes the heading to drift, it is capable of

Fig. 8. Estimation errors of alternative parametrization of $\hat{\mathbf{w}}^n$ (in color) and $\hat{\mathbf{v}}^n$ (in black). RMSE bounds dashed.

Fig. 9. Error statistics for attitude $\hat{\boldsymbol{\theta}}$.

separating the estimates equally in the airspeed data. If absolute heading is crucial, a magnetometer can be integrated. The system can find applications in various control strategies operating in scenarios with wind disturbances, for tracking chemical plumes, and areas where global satellite navigation systems cannot be used.

ACKNOWLEDGMENT

The authors wish to thank Dr. Adam Rutkowski at the U.S. Air Force Research Laboratory for providing recorded wind data.

TABLE III
ERROR STATISTICS FOR 100 MONTE CARLO SIMULATIONS.

	RMSE	Mean	Std.
δw_x^n	3.251	-2.395	2.198
δw_y^n	3.954	-3.475	1.888
δw_z^n	0.901	-0.858	0.275
δv_x^n	1.704	0.355	1.666
δv_y^n	1.540	-0.653	1.395
δv_z^n	0.526	-0.467	0.243

REFERENCES

- [1] M. Kumon, I. Mizumoto, Z. Iwai, and M. Nagata, "Wind estimation by unmanned air vehicle with delta wing," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 1896–1901, 18–22 April, 2005.
- [2] F. Ruffier and N. Franceschini, "Optic flow regulation: the key to aircraft automatic guidance," *Robotics and Autonomous Systems*, vol. 50, pp. 177–194, March, 2005.
- [3] M. Willis, "Chemical plume tracking behaviour in animals and mobile robots," *Navigation*, vol. 55, pp. 127–135, Summer 2008.
- [4] S. Edwards, A. Rutkowski, R. Quinn, and M. Willis, "Moth-inspired plume tracking strategies in three-dimensions," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 1669–1674, 18–22 April 2005.
- [5] A. Rutkowski, M. Willis, and R. Quinn, "Simulated odor tracking in a plane normal to the wind direction," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 2047–2052, 15–19 May 2006.
- [6] A. Rutkowski, R. Quinn, and M. Willis, "A sensor fusion approach to odor source localization inspired by the pheromone tracking behavior of moths," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 4873–4878, 10–14 April 2007.
- [7] A. Rutkowski, R. Quinn, and M. Willis, "Biologically inspired self-motion estimation using the fusion of airspeed and optical flow," in *American Control Conference*, 14–16 June 2006.
- [8] M. Srinivasan, "An image-interpolation technique for the computation of optic flow and egomotion," *Biological Cybernetics*, vol. 71, pp. 401–415, 1994.
- [9] M. Nagle and M. Srinivasan, "Structure from motion: determining the range and orientation of surfaces by image interpolation," *J. Optical Society of America A*, vol. 13, pp. 25–31, Jan. 1996.
- [10] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conference*, pp. 147–152, 31 August–2 September 1988.
- [11] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Computer Vision*, vol. 2, pp. 1150–1157 vol.2, 1999.
- [12] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.
- [13] D. Zachariah and M. Jansson, "Camera-aided inertial navigation using epipolar points," in *Proc. IEEE/ION Position Location and Navigation Symposium (PLANS)*, pp. 303–309, May 4–6 2010.
- [14] P. Ailliot, V. Monbet, and M. Prevosto, "An autoregressive model with time-varying coefficients for wind fields," *Environmetrics*, vol. 17, pp. 107–117, Mar. 2006.
- [15] S. Julier and J. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Proc. Signal Processing, Sensor Fusion, and Target Recognition VI*, pp. 182–193, Apr. 1997.
- [16] J. Candy, *Bayesian Signal Processing—Classical, Modern, and Particle Filtering Methods*. Wiley & Sons, 2009.
- [17] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.
- [18] R. Van der Merwe and E. Wan, "The square-root unscented kalman filter for state and parameter-estimation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 3461–3464, May 2001.
- [19] M. Hayes, *Statistical Digital Signal Processing and Modeling*. Wiley & Sons, 1996.