

Probabilistic Velocity Estimation for Autonomous Miniature Airships using Thermal Air Flow Sensors

Jörg Müller

Oliver Paul

Wolfram Burgard

Abstract—Recently, autonomous miniature airships have become a growing research field. Whereas airships are attractive as they can move freely in the three-dimensional space, their high-dimensional state space and the restriction to small and lightweight sensors are demanding constraints with respect to self-localization. Furthermore, their complex second-order kinematics makes the estimation of their pose and velocity through dead reckoning odometry difficult and imprecise. In this paper, we consider the problem of estimating the velocity of a miniature blimp with lightweight air flow sensors. We present a probabilistic sensor model that accurately models the uncertainty of the flow sensors and thus allows for robust state estimation using a particle filter. In experiments carried out with a real airship we demonstrate that our method precisely estimates the velocity of the blimp and outperforms the standard velocity estimates of the motion model as applied in many existent autonomous blimp navigation systems.

I. INTRODUCTION

In recent years, autonomous navigation for miniature flying vehicles has become a growing research field. Equipped with sensors and communication devices these robots can fulfill various tasks such as surveillance and exploration. For these tasks, miniature airships, in particular, have desirable properties. Their low power consumption combined with their slow and safe flight behavior make them suitable for indoor operation even in the presence of people. To fulfill their tasks autonomously and in a reliable and practical way, they require precise self-localization. Usually, this is challenging as these platforms are restricted to lightweight and small on-board sensors. In many navigation systems for miniature blimps, a small number of lightweight sensors like sonars or micro electromechanical systems (MEMS) based inertial measurement units (IMUs) were employed [18], [25]. In contrast to camera images, their low-dimensional measurement output can be processed even with limited computational resources. Probabilistic localization systems based on such sensors typically also rely on dead reckoning odometry for velocity estimation. In the context of autonomous blimps, however, the precision of the motion prediction suffers from large errors, even when adaptive motion models [17] are used.

In this paper, we consider the problem of estimating the state of a miniature blimp in indoor environments through probabilistic state estimation with a particle filter. To measure the air speed, our blimp [19], which is shown in Fig. 1, is equipped with two MEMS-based thermal air flow micro-sensors. These smart sensors enable the velocity of media

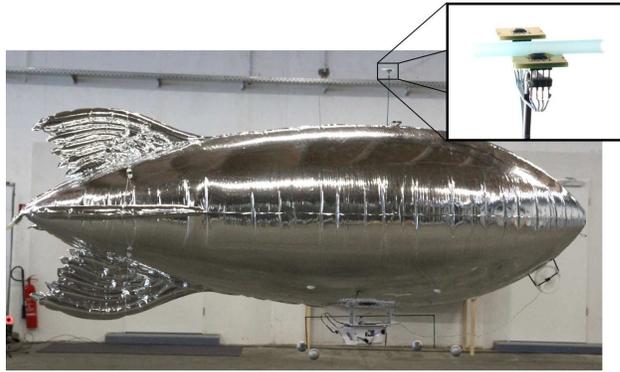


Fig. 1. The robotic indoor blimp [19] is 2.1 m long and has a payload of about 150 grams. It is actuated by three propellers and is equipped with two thermal air flow sensors from Sensirion AG, Stäfa, Switzerland, mounted on a pole on the top of the hull.

sweeping over them to be determined through the detection of on-chip thermal differences. We present and compare two regression techniques for approximating the characteristics of the flow sensors including their measurement uncertainty and propose an approach to use the sensor data for velocity estimation in a particle filter. Thereby, our approach allows for robust state estimation through the modeling of all underlying uncertainties. Furthermore, other sensors like an IMU or sonars can seamlessly be integrated into our approach. In practical experiments we demonstrate that our approach to velocity estimation outperforms the dead reckoning motion models applied in many autonomous blimp navigation systems [10], [13], [27].

This paper is organized as follows. After discussing related work in the following section, we briefly describe Monte Carlo state estimation in Section III and present the applied airship motion model in Section IV. We then introduce our approach to probabilistic modeling of flow sensors in Section V. Finally, in Section VI we demonstrate the capabilities of our approach in experiments with a real robotic blimp.

II. RELATED WORK

A popular application of airspeed sensors on UAVs is the combination with GPS for wind estimation [2]. In this context, approaches to calibrate the scaling of an airspeed sensor have also been developed [1].

Furthermore, several authors considered state estimation or control of robots based on flow sensors. For example, Fei *et al.* [9] and Tokutake *et al.* [22] utilize thermal flow sensors on the wings of small unmanned aircrafts for the detection of flight parameters including the airspeed. Kruusmaa *et al.* [14] determine the optimal position of pressure sensors

on an artificial trout to estimate the velocity using a quadratic regression model. For attitude estimation, Euston *et al.* [8] fuse IMU and airspeed measurements of a UAV.

Many researchers use optical flow on image sensors, like the low cost devices employed in optical mice, for improved dead reckoning odometry on ground robots [12]. Dille *et al.* [7] additionally apply an online re-calibration of the local linear regression calibration for an optical flow sensor. Likewise, Conroy *et al.* [3] determine the maximum likelihood velocity of a quadrotor using an omni-cam based optical flow sensor. Similar techniques have been applied on micro aerial vehicles such as the palm-sized glider designed by Woods *et al.* [24] inspired by flying insects. Here, the authors utilize an embedded low resolution optical flow sensor for target detection and obstacle avoidance. In addition to an optical flow sensor, some authors employ airspeed sensors for flight stabilization [11] or even ground speed and wind estimation [20].

However, all approaches for state estimation or control of robots based on flow sensors mentioned above apply maximum likelihood state estimation or control based on the calibrated output of one or multiple identical sensors. In contrast, we explicitly model the uncertainty of the measurements and of the motion of the robot for probabilistic state estimation. Thus, our approach can seamlessly integrate arbitrary sensors and takes into account the control signals sent to the actuators of the robot.

III. RECURSIVE MONTE CARLO STATE ESTIMATION

In this paper we consider the problem of estimating the state \mathbf{x} of a robot using Monte Carlo localization [5]. The key idea of this approach is to maintain a probability density $p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ of the state \mathbf{x}_t at time t conditioned on all sensor data $\mathbf{z}_{1:t}$ and control commands $\mathbf{u}_{1:t}$ up to time t . This probability density is calculated recursively using the recursive Bayesian filtering scheme [21]

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \eta_t p(\mathbf{z}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1}, \quad (1)$$

where η_t is a normalizing constant ensuring that $\int p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) d\mathbf{x}_t = 1$. In (1), the term $p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1})$ is the state transition probability and $p(\mathbf{z}_t | \mathbf{x}_t)$ is the measurement probability specified by the motion model and the sensor model, respectively.

In Monte Carlo localization, we approximate the current belief $p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ by a set $\mathcal{M} = \{(\mathbf{x}^{[i]}, w^{[i]})\}_{i \in [1, N]}$ of N particles, each of which corresponds to a state hypothesis $\mathbf{x}^{[i]}$ weighted by the so-called importance weight $w^{[i]}$. Furthermore, we perform the recursive belief update given in (1) according to the following three steps:

- 1) In the *prediction step*, we propagate each particle by drawing a successor state based on the motion model $p(\mathbf{x}_t^{[i]} | \mathbf{u}_t, \mathbf{x}_{t-1}^{[i]})$ given the control command \mathbf{u}_t .
- 2) In the *correction step*, we integrate a new measurement \mathbf{z}_t by assigning a new weight $w^{[i]} \propto p(\mathbf{z}_t | \mathbf{x}_t^{[i]})$ to each particle according to the sensor model.

- 3) In the *resampling step*, we draw a new generation of particles from \mathcal{M} (with replacement) such that each sample in \mathcal{M} is selected with a probability that is proportional to its weight.

IV. MINIATURE AIRSHIP MOTION MODEL

The probabilistic motion model $p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1})$ is the core component of the prediction step of the Monte Carlo state estimation. We first derive a deterministic model by considering the underlying physics of the motion of miniature airships. In particular, our model is based on the work of Zufferey *et al.* [27] and adapted to our type of airship [17]. In a second step, we extend the deterministic model by a statistical identification of the sources of uncertainty.

We define the state of the blimp as

$$\mathbf{x} = [\mathbf{p}^T, \mathbf{q}^T, \mathbf{v}^T, \boldsymbol{\omega}^T]^T \quad (2)$$

consisting of the position $\mathbf{p} = [x, y, z]^T$, the orientation $\mathbf{q} = [q_0, q_1, q_2, q_3]^T$ represented by a unit quaternion [6], the translational velocity $\mathbf{v} = [v_x, v_y, v_z]^T$, and the angular velocity $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$. Additionally, we define the translational acceleration $\mathbf{a} = [a_x, a_y, a_z]^T = \dot{\mathbf{v}}$ and the angular acceleration $\boldsymbol{\alpha} = [\alpha_x, \alpha_y, \alpha_z]^T = \dot{\boldsymbol{\omega}}$. The position and orientation are expressed in the global frame of reference \mathcal{F}_g (with the z -axis pointing upwards). The velocities, accelerations, forces, and torques are expressed in the body-fixed frame of reference \mathcal{F}_b . The origin of the body-fixed frame \mathcal{F}_b is the center of buoyancy of the blimp with the x -axis pointing forward and the z -axis pointing upwards.

The Newton-Euler equation of motion

$$M \begin{bmatrix} \mathbf{a} \\ \boldsymbol{\alpha} \end{bmatrix} = \boldsymbol{\mathfrak{F}}_{\text{external}}(\mathbf{x}) + \boldsymbol{\mathfrak{F}}_{\text{fictitious}}(\mathbf{x}) \quad (3)$$

ouples the acceleration to the force and torque $\boldsymbol{\mathfrak{F}} = \begin{bmatrix} \mathbf{F} \\ \boldsymbol{\tau} \end{bmatrix}$. With respect to \mathcal{F}_b , the inertia matrix

$$M = \begin{bmatrix} m I_{3 \times 3} & -m S(\mathbf{r}_g) \\ m S(\mathbf{r}_g) & J \end{bmatrix} + \text{diag}(k_1 m_{\text{air}}, k_2 m_{\text{air}}, k_2 m_{\text{air}}, 0, k' J_{\text{air}, y}, k' J_{\text{air}, z}) \quad (4)$$

$$=: \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad \text{with} \quad M_{ij} \in \mathbb{R}^{3 \times 3}$$

is composed of the mass of the airship m and its moment of inertia J , the skew symmetric matrix operator

$$S(\mathbf{r}) = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix} \quad (5)$$

and the position of the center of gravity \mathbf{r}_g of the blimp. The air accompanying the airship is taken into account by Lamb's virtual mass coefficients k_1 , k_2 , and k' [15], where m_{air} and J_{air} are the mass and the moment of inertia of the air displaced by the blimp. Here, we exploit rotation symmetry of the hull of the blimp around its x -axis.

The fictitious forces and torques are caused by Coriolis and centripetal effects in the moving frame of reference \mathcal{F}_b . They can be efficiently calculated from the inertia matrix as

$$\begin{aligned} \mathfrak{F}_{\text{fictitious}} &= \begin{bmatrix} O_{3 \times 3} & S(M_{11}\mathbf{v} + M_{12}\boldsymbol{\omega}) \\ S(M_{11}\mathbf{v} + M_{12}\boldsymbol{\omega}) & S(M_{21}\mathbf{v} + M_{22}\boldsymbol{\omega}) \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}, \quad (6) \end{aligned}$$

where $O_{3 \times 3}$ is the zero block matrix [27]. The external forces and torques $\mathfrak{F}_{\text{external}} = \mathfrak{F}_{\text{bg}} + \mathfrak{F}_{\text{D,h}} + \mathfrak{F}_{\text{D,f}} + \mathfrak{F}_{\text{r}}$ consist of the buoyancy and gravity \mathfrak{F}_{bg} , the drag of the hull $\mathfrak{F}_{\text{D,h}}$ and the fins $\mathfrak{F}_{\text{D,f}}$, and the propulsion of the rotors \mathfrak{F}_{r} . The buoyancy and gravity can be calculated jointly as

$$\mathfrak{F}_{\text{bg}} = \begin{bmatrix} \mathbf{F}_b + \mathbf{F}_g \\ \mathbf{r}_g \times \mathbf{F}_g \end{bmatrix} \quad (7)$$

with

$$\begin{bmatrix} 0 \\ \mathbf{F}_b \end{bmatrix} = \bar{\mathbf{q}} \odot \begin{bmatrix} 0 \\ 0 \\ 0 \\ m_{\text{air}} g \end{bmatrix} \odot \mathbf{q} \quad \text{and} \quad \begin{bmatrix} 0 \\ \mathbf{F}_g \end{bmatrix} = \bar{\mathbf{q}} \odot \begin{bmatrix} 0 \\ 0 \\ 0 \\ -mg \end{bmatrix} \odot \mathbf{q},$$

where \odot is the quaternion product and $\bar{\mathbf{q}}$ is the adjoint of \mathbf{q} [6].

In the typical range of operation, our blimp has a Reynolds number $Re \approx 30,000$, so that we can safely drop the viscous resistance term and specify the air drag by the quadratic term. We approximate the drag force and torque of the hull in an uncoupled way as

$$\begin{aligned} \mathfrak{F}_{\text{D,h}} &= [-D_1 v_x |v_x|, -D_2 v_y |v_y|, -D_2 v_z |v_z|, \\ &0, -D' \omega_y |\omega_y|, -D' \omega_z |\omega_z|]^T. \quad (8) \end{aligned}$$

Here, D_1 , D_2 , and D' are the drag coefficients and we exploit rotation symmetry along the x -axis and neglect the ω_x -component which is dominated by the drag of the fins. Analogously, the drag force of each fin acts at its center \mathbf{r}_f parallel to its normal \mathbf{n}_f and scales with the fin drag coefficient D_f , its area A_f , and the velocity component along the normal:

$$\mathbf{F}_f = -D_f A_f (\mathbf{n}_f \cdot (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_f)) |\mathbf{n}_f \cdot (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_f)| \mathbf{n}_f.$$

The forces and torques for each fin and rotor are

$$\mathfrak{F}_{\text{D,f}} = \begin{bmatrix} \mathbf{F}_f \\ \mathbf{r}_f \times \mathbf{F}_f \end{bmatrix} \quad \text{and} \quad \mathfrak{F}_{\text{r}} = \begin{bmatrix} \mathbf{F}_r \\ \mathbf{r}_r \times \mathbf{F}_r \end{bmatrix}, \quad (9)$$

where \mathbf{F}_r is the rotor force (depending on the current control signal) and \mathbf{r}_r is the rotor position.

Finally, we solve the second-order differential equation (3) through numerical integration assuming constant acceleration during each time step.

We determine those parameters, which are hard to identify individually, from data recorded during operation of the real airship by minimizing the difference

$$\Delta = M^{-1} (\mathfrak{F}_{\text{external}}(\mathbf{x}) + \mathfrak{F}_{\text{fictitious}}(\mathbf{x})) - \begin{bmatrix} \mathbf{a}^* \\ \boldsymbol{\alpha}^* \end{bmatrix} \quad (10)$$

between the accelerations estimated by the motion model and the ground truth accelerations \mathbf{a}^* and $\boldsymbol{\alpha}^*$. From the

sequence of differences Δ we can estimate the covariance of the accelerations calculated by the learned model. This covariance implicitly defines the probabilistic model needed in the prediction step of the particle filter by error propagation through the numerical integration.

V. FLOW SENSOR MODEL

There exist various techniques for measuring air velocity. Whereas cup, windmill, and sonic anemometers are rather heavy and bulky, hot-wire anemometers and thermal mass flow meters can be built in MEMS technology and therefore are suitable even for employment on miniature flying vehicles. Most of these flow sensors have in common that their one-dimensional measurement value $\mathbf{z} \in \mathbb{R}^1$ depends on the air velocity $v_{\mathbf{z}}$ along the measurement axis of the sensor.

We model this measurement principle by assuming the heteroscedastic measurement process

$$\mathbf{z} = h(v_{\mathbf{z}}) + \varepsilon \quad \text{with} \quad \varepsilon \sim \mathcal{N}(0, \sigma(v_{\mathbf{z}})^2), \quad (11)$$

where h is a strictly monotonic increasing function. The noise ε typically depends on the sensor characteristics as well as the air velocity.

In indoor navigation scenarios, we assume the air to be static (no wind) and the sensor to be placed at a sufficient distance from the hull so that the influence of the surrounding air accompanying the blimp [15] can be neglected. Depending on the translational and rotational velocity of the blimp, the velocity of the position $\mathbf{r}_{\mathbf{z}}$ where the sensor is rigidly mounted is $\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_{\mathbf{z}}$. Hence, the velocity component of the sensor along its measurement axis $\mathbf{n}_{\mathbf{z}}$ is

$$v_{\mathbf{z}}(\mathbf{x}) = (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_{\mathbf{z}}) \cdot \mathbf{n}_{\mathbf{z}}. \quad (12)$$

According to (11) the probabilistic measurement model used for Monte Carlo state estimation is defined by the Gaussian distribution

$$p(\mathbf{z} | \mathbf{x}) = \mathcal{N}(h(v_{\mathbf{z}}(\mathbf{x})), \sigma(v_{\mathbf{z}}(\mathbf{x}))^2). \quad (13)$$

For the implementation of the model described above, we need a function approximating $h: \mathbb{R} \rightarrow \mathbb{R}$ and $\sigma: \mathbb{R} \rightarrow \mathbb{R}$ from a set of training data $\{(x_i, y_i)\}_{i \in [1, n]}$. Each training data point contains the velocity $v_{\mathbf{z}}$ of the sensor relative to the air in x and the measured flow value \mathbf{z} in y . According to our model, all points are assumed to be generated from $y_i = h(x_i) + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma(x_i)^2)$. In the following, we present a non-parametric and a parametric regression approach. As shown in Fig. 2, both are suitable for our regression analysis problem.

A. Local Linear Regression

In general, local regression [23] computes a weighted average of the training function values y_i giving a higher weight to those points near the requested value x . In our approach we apply the Gaussian kernel

$$w(x, x') = \frac{1}{\sqrt{2\pi} l} \exp\left(-\frac{(x - x')^2}{2l^2}\right) \quad (14)$$

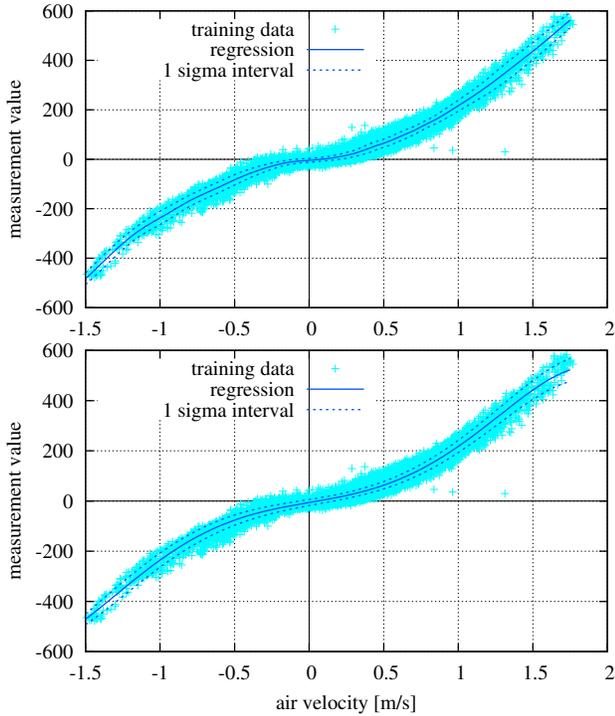


Fig. 2. The local linear regression (top) and the polynomial regression (bottom) on the tube sensor training data generated from about 20 min of operation. The regression on the measurement noise is represented by the 1σ interval.

with bandwidth l for local weighing. For a compact representation we define

$$W(x) = \frac{1}{\sum_{i=1}^n w(x, x_i)} \text{diag}(w(x, x_1), \dots, w(x, x_n)),$$

$$X = \begin{bmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_n \end{bmatrix}, \text{ and } Y = [y_1 \quad \dots \quad y_n].$$

In local linear regression [23], each function value is computed from a linear regression

$$f(x) = A(x) [1, x]^T, \quad (15)$$

where the coefficient matrix A minimizes the locally weighted sum of squared errors

$$(Y - AX)^T W(x) (Y - AX). \quad (16)$$

In the linear, one-dimensional case it is $A = [a_0, a_1]$. Minimizing (16) gives the weighted least squares estimator

$$\hat{A}(x) = Y W(x) X^T (X W(x) X^T)^{-1} \quad (17)$$

and finally the estimated function value $\hat{f}(x) = \hat{A}(x) [1, x]^T$.

For our flow sensor model we extend the local linear regression by an estimate of the covariance σ^2 . In the training stage, we calculate $\varepsilon_i = y_i - \hat{f}(x_i)$ for each training data point. Based on these values, we estimate $\sigma(x)^2$ as the local constant regression [23] on ε^2

$$\hat{\sigma}(x)^2 = \frac{\sum_{i=1}^n w(x, x_i) \varepsilon_i^2}{\sum_{i=1}^n w(x, x_i)}. \quad (18)$$

B. Polynomial Regression

An alternative technique, which is less flexible but usually more efficient than local regression, is the polynomial regression. For a compact representation of the polynomial function

$$f(x) = \sum_{d=0}^p a_d x^d = A_p [1, x^1, \dots, x^p]^T \quad (19)$$

of degree p we define the regression parameter $A_p = [a_0, \dots, a_p]$ and

$$X_p = \begin{bmatrix} 1 & \dots & 1 \\ x_1^1 & \dots & x_n^1 \\ \vdots & & \vdots \\ x_1^p & \dots & x_n^p \end{bmatrix}. \quad (20)$$

Minimizing the squared sum of estimation errors

$$(Y - A_p X_p)^T (Y - A_p X_p) \quad (21)$$

on the training data gives the polynomial least squares estimator [23]

$$\hat{A}_p = Y X_p^T (X_p X_p^T)^{-1} \quad (22)$$

and finally the estimate $\hat{f}(x) = \hat{A}_p [1, x^1, \dots, x^p]^T$.

Here, we estimate the covariance $\sigma(x)^2$ by another polynomial regression

$$\hat{\sigma}(x)^2 = \hat{A}'_p [1, x^1, \dots, x^p]^T \quad (23)$$

on ε^2 with $\varepsilon_i = y_i - \hat{f}(x_i)$ and

$$\hat{A}'_p = [\varepsilon_1^2, \dots, \varepsilon_n^2] X_p^T (X_p X_p^T)^{-1}. \quad (24)$$

As shown in Fig. 2, both regression techniques are suitable for our flow sensor model.

VI. EXPERIMENTAL EVALUATION

We evaluated our approach in extensive experiments with a real robotic blimp in a large indoor environment of about $20 \times 12 \text{ m}^2$ with a vertical space of 5 m. Our blimp, shown in Fig. 1, is 2.1 m long. The payload of about 150 grams is used for sensors and a Gumstix computer communicating via WiFi with a laptop computer.

Additionally, we equipped our blimp with two SDP600 differential pressure sensors¹ from Sensirion AG, Stäfa, Switzerland, operated here as thermal flow sensors. Although various other suitable miniature flow sensors are available [4], [26], we chose the Sensirion sensors because they have several desirable properties. They are fully developed, have a weight below 1 gram, a very low power consumption, react quickly to changes in the gas flow, and their integrated evaluation circuitry can be controlled via the I²C interface available on the Gumstix computer of the blimp. In our first experiments we placed the sensors, similarly to Fei *et al.* [9] and Tokutake *et al.* [22], directly onto the hull of the blimp. However, in these setups the measurements showed

¹The data sheet is available at http://www.sensirion.com/en/pdf/product-information/Datasheet.SDP600series.differential_pressure_sensor.pdf.

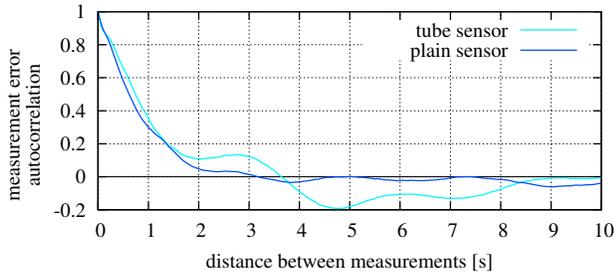


Fig. 3. The autocorrelation of measurement errors for both flow sensors with respect to the local linear regression model.

a clear influence of the surrounding air accompanying the blimp which is hard to model. Even when placing the sensors on a pole at a distance of 10 cm from the hull, the velocity estimates provided by our particle filter approach were worse than those obtained with the plain motion model. Thus, we finally mounted the flow sensors on a pole at a distance of 20 cm from the hull (see Fig. 1). Because the estimation of the forward velocity suffered from large errors in our previous localization systems [17], [18], we placed the sensors on top of the blimp with the measurement axis pointing forward. As we expected to reduce the air turbulences in the vicinity of the thermal elements of the sensor in this way, we mounted a short tube onto one of the sensors. This sensor will be called “tube sensor” throughout our experiments.

In the run-up to the experiments, we learned the parameters of our motion model from about 20 min of manually operated flight observed by a MotionAnalysis motion capture (MoCap) system with eight Raptor-E cameras. We obtained precise velocity and acceleration ground truth data from a quadratic regression on the reference trajectory estimated by the MoCap. From the same reference trajectory data (including 24,196 measurements of each sensor at 20 Hz) we trained the flow sensor models as shown in Fig. 2.

In our implementation of the Monte Carlo state estimation we apply low-variance resampling [21] and skip the resampling step as long as the effective number of particles [16] is greater than half the number of particles. For the reason of efficiency, we restricted the considered points in the local linear regression to those having a significant weight. We chose the bandwidth $l = 0.1$ of the Gaussian kernel and the polynomial degree $p = 5$.

We evaluated our approach in terms of the state estimation error in an experiment of about 12 min of manual operation during which we acquired ground truth states from the MoCap system. Throughout this experiment, the operator carried out many different maneuvers including strong accelerations and rotations which also caused rocking movements of the blimp. We evaluated the quality of the velocity estimation as the root mean square error (RMSE) of the estimated forward velocity v_x with respect to the actual (“ground truth”) value v_x^* from the MoCap system.

The Bayesian filtering scheme, on which our state estimation approach is based, relies on the Markov assumption [21] that the measurement noise of consecutive measurements to

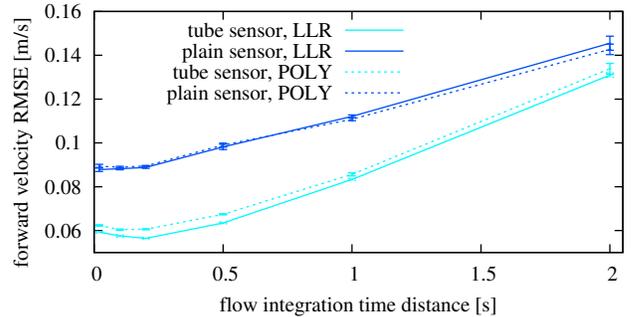


Fig. 4. The average forward velocity RMS error of the different flow sensors together with the local linear regression (LLR) and the polynomial regression (POLY) compared for different time distances between two integrations of sensor measurements into the particle filter. The error bars indicate the 95% confidence intervals over ten runs.

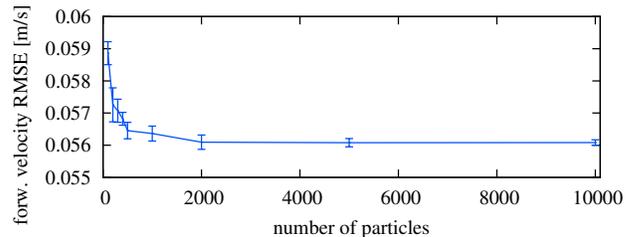


Fig. 5. The average forward velocity RMS error for different numbers of particles in our filter with the tube flow sensor and the local linear regression model. The error bars indicate the 95% confidence intervals over ten runs.

be conditionally independent. To validate this with respect to our flow sensor model we evaluated the autocorrelation of the measurement noise depending on the time lag (see Fig. 3). Despite of the high level of correlation even at a one-second time lag, the results of our approach showed its optimal performance when a flow measurement is integrated into the belief of the filter every 0.2 seconds (see Fig. 4). This is caused by the fact that the filter benefits from the information of more frequent measurements even if there is a limited correlation which is not modeled. In our experiments, the tube sensor significantly outperforms the plain sensor and the local linear regression model seems to be slightly better than the polynomial regression model, at least for the tube sensor.

Fig. 5 shows the robustness of our approach to the Monte

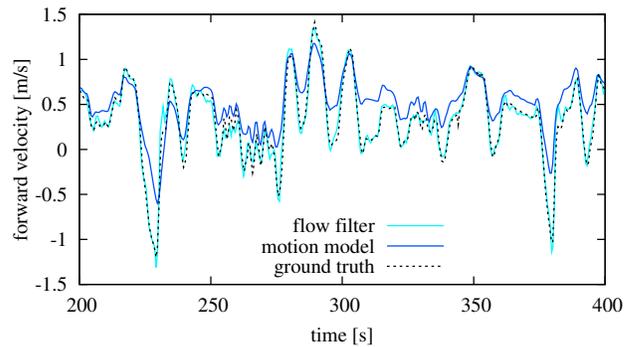


Fig. 6. An exemplary extract of the forward velocity estimates of our filter with the tube flow sensor with the local linear regression model compared to the motion model estimates and the ground truth data.

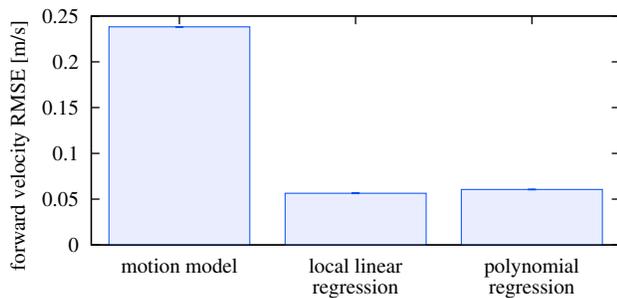


Fig. 7. The average forward velocity RMS error of the estimates of our filter with the tube flow sensor compared to the motion model estimates. The error bars indicate the 95% confidence intervals over ten runs.

Carlo approximation with a limited number of particles. As a trade-off between precision and runtime, 500 particles seem to be the best choice. Furthermore, we compared the forward velocity estimates of our approach with both flow sensor models to those of the motion model without any sensor data fusion. As shown in the accompanying video and Fig. 6 and 7, both sensor models significantly outperform the plain motion model.

VII. CONCLUSIONS

In this paper, we presented an approach to probabilistic velocity estimation for miniature indoor airships using thermal air flow sensors. In contrast to other approaches, we explicitly consider the measurement uncertainties of the thermal flow sensors and probabilistically fuse their measurements with the prediction calculated by the airship motion model in a particle filter. Additionally, our approach allows to seamlessly integrate other sensors such as sonars or an IMU for localization. We compared two regression methods for sensor calibration including uncertainties. Both proved to be suitable for probabilistic velocity estimation. In experiments with a real blimp operating in a large indoor environment, we demonstrated that our approach precisely estimates the velocity of the blimp and outperforms the velocity estimates of a standard motion model for miniature airships. In future work we would like to extend our model to two-dimensional flow sensors [4] and exploit the capabilities of our approach for autonomous navigation and closed-loop control of the blimp in combination with additional sensor technologies.

ACKNOWLEDGMENTS

This work has partly been supported by the German Research Foundation (DFG) within the Research Training Group 1103. The authors would like to thank Jan Peters for valuable suggestions and fruitful discussions. Sensirion AG is gracefully acknowledged for providing flow sensor chips.

REFERENCES

- [1] H. Chao and Y.Q. Chen. Surface wind profile measurement using multiple small unmanned aerial vehicles. In *Proc. of the American Control Conf.*, 2010.
- [2] A. Cho, J. Kim, S. Lee, and C. Kee. Wind estimation and airspeed calibration using a UAV with a single-antenna GPS receiver and pitot tube. *IEEE Transactions on Aerospace and Electronic Systems*, 47(1), 2011.

- [3] J. Conroy, G. Gremillion, B. Ranganathan, and J.S. Humpert. Implementation of wide-field integration of optic flow for autonomous quadrotor navigation. *Journal of Autonomous Robots*, 27, 2009.
- [4] A.S. Cubukcu, E. Zernickel, U. Buerklin, and G.A. Urban. A 2d thermal flow sensor with sub-mW power consumption. *Sensors and Actuators A: Physical*, 163:449–456, 2010.
- [5] F. Dellaert, D. Fox, W. Burgard, and W. Thrun. Monte carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1999.
- [6] J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Technical report, Stanford University, 2006.
- [7] M. Dille, B. Grocholsky, and S. Siingh. Outdoor downward-facing optical flow odometry with commodity sensors. In *Proc. of the Int. Conf. on Field and Service Robots (FSR)*, 2009.
- [8] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel. A complementary filter for attitude estimation of a fixed-wing UAV. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008.
- [9] H. Fei, R. Zhu, Z. Zhou, and J. Wang. Aircraft flight parameter detection based on a neural network using multiple hot-film flow speed sensors. *Smart Materials and Structures*, 16, 2007.
- [10] S.B.V. Gomes and J.G. Jr. Ramos. Airship dynamic modeling for autonomous operation. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1998.
- [11] J. Keshavan and J.S. Humpert. MAV stability augmentation using weighted outputs from distributed hair sensor arrays. In *Proc. of the American Control Conference*, 2010.
- [12] S. Kim and S. Lee. Robust velocity estimation of an omnidirectional mobile robot using a polygonal array of optical mice. *Int. Journal of Control, Automation, and Systems*, 6(5), 2008.
- [13] J. Ko, D.J. Klein, D. Fox, and D. Haehnel. GP-UKF: Unscented kalman filters with gaussian process prediction and observation models. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007.
- [14] M. Kruusmaa, G. Toming, J. Salumäe, J. Ješov, and A. Ernits. Swimming speed control and on-board flow sensing of an artificial trout. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.
- [15] H. Lamb. The inertia coefficients of an ellipsoid moving in fluid. Technical Report Reports and Memoranda, No. 623, British Aeronautical Research Committee, October 1918.
- [16] J.S. Liu. Metropolisized independent sampling with comparisons to rejection sampling and importance sampling. *Statist. Comput.*, 6, 1996.
- [17] J. Müller, C. Gonsior, and W. Burgard. Improved monte carlo localization of autonomous robots through simultaneous estimation of motion model parameters. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.
- [18] J. Müller, A. Rottmann, and W. Burgard. A probabilistic sonar sensor model for robust localization of a small-size blimp in indoor environments using a particle filter. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.
- [19] A. Rottmann, M. Sippel, T. Ziterell, W. Burgard, L. Reindl, and C. Scholl. Towards an experimental autonomous blimp platform. In *Proc. of the European Conf. on Mobile Robots (ECMR)*, 2007.
- [20] A.J. Rutkowski, M.M. Miller, R.D. Quinn, and M.A. Willis. Egomotion estimation with optic flow and air velocity sensors. *Biological Cybernetics*, 104(6), 2011.
- [21] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [22] H. Tokutake, S. Sunada, and J. Fujinaga. Attitude control of a small UAV using a flow sensor system. *Journal of System Design and Dynamics*, 5(1), 2011.
- [23] L. Wasserman. *All of Nonparametric Statistics*. Springer, 2006.
- [24] R.J. Woods, S. Avadhanula, E. Steltz, M. Seeman, J. Entwistle, A. Bachrach, G. Barrows, S. Sanders, and R.S. Fearing. An autonomous palm-sized gliding micro air vehicle. *IEEE Robotics & Automation Magazine*, 14(2), 2007.
- [25] G. Wyeth and I. Barron. An autonomous blimp. In *Proc. of the IEEE Int. Conf. on Field and Service Robots (FSR)*, 1997.
- [26] Y. Yang, S. Pandya, J. Chen, J. Engel, N. Chen, and C. Liu. Micro-machined hot-wire boundary layer flow imaging array. In *Proc. of CANEUS Conf.*, 2006.
- [27] J.C. Zufferey, A. Guanella, A. Beyeler, and D. Floreano. Flying over the reality gap: From simulated to real indoor airships. *Autonomous Robots*, 21(3), 2006.