

On Autonomous Indoor Flights: High-Quality Real-Time Localization using Low-Cost Sensors

Juergen Eckert*, Reinhard German* and Falko Dressler†

*Dept. of Computer Science, University of Erlangen, Germany

†Institute of Computer Science, University of Innsbruck, Austria

{juergen.eckert,reinhard.german}@cs.fau.de, falko.dressler@uibk.ac.at

Abstract—Indoor hovering objects such as quadrotors need to be controlled continuously to hold their position in space. In order to support fully autonomous flights of these copters, the necessary position control including all related information transfers have to be provided in a fully decentralized and autonomous manner. We discuss challenges related to flight control based on our Autonomous Localization Framework (ALF), which provides scalable and decentralized localization in GPS-denied areas. Using a sensor network based on the IEEE 802.15.4 communication protocol, continuous position maintenance is feasible but, unfortunately, in no way stable. Therefore, we introduce a low-cost sensor array, which reduces the system dynamics and allows a robust position control of the platform.

I. INTRODUCTION

Controlling Unmanned Aerial Vehicles (UAVs) such as quadrotors using low-cost sensor nodes for indoor localization is still one of the most challenging topics in the field of sensor and actor networks [1]. Besides the necessary coordination among all nodes in a certain communication range, real-time collaboration is needed for identifying rather fast movements of the quadrotors in space due to air circulations and other disturbances. In general, six degrees of freedom need to be controlled continuously for motion in the three-dimensional (3D) space. The most simple fall back mechanism in case of an uncertain position and/or orientation in the field of robotics is to just stop – this, of course, can not be applied during the flight of an UAV. In this paper, we focus on a specific class of UAVs, the so called vertical take-off and landing (VTOL) devices, which are particularly suited for our indoor scenarios. Here, however, not much space for maneuvering is available, which forces the platform to mostly remain in its most unstable condition: hovering. Serious self-caused air turbulences additionally affect the flight stability.

Indoor position controlling approaches for flying objects based on localization systems [2] or Simultaneous Localization and Mapping (SLAM) [3] already exist. However, they either depend on a highly accurate and expensive localization system that needs to be deployed manually; or they need a high amount of (mostly off-board) computing power. In the scope of this paper, we rely on our recently presented ALF framework [4], which, to the best of our knowledge, is the first platform fulfilling all the requirements for a real autonomous indoor flying system, including the necessary sensor deployment and information acquisition. The key idea – exemplarily depicted in Figure 1 – is to rely on a number of low-cost sensor nodes



Fig. 1. Quadrotor hovering over mobile ground sensors (distances reduced)

(attached to mobile robots) that fully autonomously spread in a given area to span a reference grid using Ultrasound (US) distance measurements. The same technology is also available for localizing flying quadrotors or other systems with high precision. We present an architecture that allows the UAV to perform a flight that is well controlled based on the distributed sensors spanning the reference grid. We are not only able to steer the system, but most importantly also to handle turbulences and other random noise.

The key contributions can be summarized as follows:

- We present a fully self-organizing approach for indoor flights of a quadrotor controlled by a set of ground sensors spanning a localization grid. To the best of our knowledge, this is the first approach to perform such autonomous flights only based on simple US sensors.
- In addition, we substantially increase the robustness of the system by adding two different low-cost and light-weighted sensors to the aircraft. The smart combination of all those sensors allow a fault-tolerant and safe flight of the robot.

II. RELATED WORK

Autonomous VTOL aircrafts – especially quadrotors – created substantial research activities in the last years, especially in GPS-denied environments. Our goal is to enable fully autonomous flights of multiple low-cost copters. Even though, expensive equipment like laser distance sensors [5] or an off-

board localization system [2], which need to be deployed manually, have been proven to be suitable for this task, we focused on simple low-cost sensors.

Most commonly, on-board vision based sensors are being used. Stereo cameras can be used to generate a 3D model of the environment surrounding the vehicle. In order to circumvent the need of the considerable on-board computational power to build the world model from the gathered data, two different concepts exist: Huang *et al.* [6] use a RGB-D camera in order not to generate but to measure the depth information; while Bills *et al.* [7] are not building a 3D model. Instead, they rely on perspective cues from a single image. Both light-weighted approaches still require too much computational effort.

Another common vision-based concept is called ‘optical flow’ using the pattern of apparent motion of objects in relation to the sensor. Again, a standard camera in combination with a fast computing unit can be used as reported by Conroy *et al.* [8]. More important are specialized flow sensors as typically installed in computer mice. The frequently used sensor type (e.g., in [9]) is light-weighted and requires a minimum of computational effort. In contrast to the previous sensing approach, sampling rates of more than 1 kHz are achievable. The only drawback is that VTOL devices tilt during flight, which needs to be compensated.

For non-vision sensing, Roberts *et al.* [10] present the minimal requirements for an autonomous indoor flight. Based on US and infrared, the system can avoid collisions but can not stay at a given position. However, a common knowledge base (e.g., relative positions) is needed in case of a multi robot application. The approach of Chintalapudi *et al.* [11] provides such a knowledge base. A reference grid is autonomously generated using Received Signal Strength Indicator (RSSI) measurements (of WiFi-enabled devices). Besides of the too low accuracy for indoor navigation, the system requires too much computational power for an embedded device.

III. TOWARDS AUTONOMOUS INDOOR FLIGHTS

The key objective of this paper is to provide a position-hold mechanisms for VTOL robots. If a flight on a trajectory or obstacle avoidance is required, it can be done by dynamically adjusting the desired hold position.

A. Physical Model of a Quadrotor

Besides of the actual 3D position $p_c = (\vec{x}, z)^T$; $\vec{x} = (x, y)^T$ of the platform, the rotations of the three axes (pitch, roll $\vec{\Phi} = (\Phi_x, \Phi_y)^T$, and yaw Ψ) are highly important. The forward/back x and right/left y translation is indirectly controlled by pitch Φ_x and roll Φ_y , respectively. Figure 3(a) shows the craft system of a single axis $i \in [x, y]$, where F_t is the thrust of all rotors, m is the total weight of the platform and g is the gravity. (Simplifying assumption: $\Phi_j = 0 : i \neq j$.) For flying without an altitude variation the following equation must hold: $F_t \cdot \cos \Phi_i = mg$. Applying a constant angle Φ_i , different from zero, to an axis i will cause the platform to constantly accelerate \ddot{x}_i into this direction. This reduces the amount of independent degrees to four: pitch Φ_x , roll Φ_y , yaw Ψ , and the altitude z .

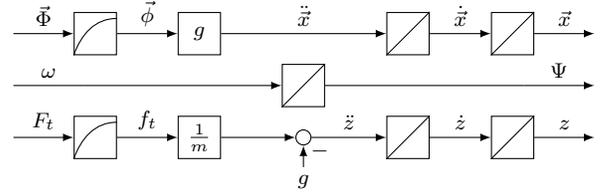


Fig. 2. Quadrotor system model

B. System Model and its Minimal Sensor Requirements

For controlling the platform, it is essential to consider the system behavior. Assuming the platform always hovers (small angles) and by accepting small modeling errors, each degree of freedom can be controlled independently. Thus, we approximated the model of a quadrotor as depicted in Figure 2.

On-board rate integrating gyroscopes can measure the angular velocity ω but not the absolute heading $\Psi = \int_0^t \omega(\tau) d\tau + c_\Psi$ (see Figure 2 middle row). We observed that compass sensors, commonly used in outdoor scenarios, can not robustly determine the offset c_Ψ indoor. A different approach is required that will be explained later in this section. For stabilization, a simple proportional (P) controller can be utilized.

The position \vec{x} is more complex as it needs to be indirectly controlled by tilting $\vec{\Phi}$ (see Figure 2 top row). Due to inertia, the tilting takes some time (approx. 150 ms on our platform). We approximated this using a first order lag element. As previously introduced, the tilting angles $\vec{\Phi}$ introduce an acceleration $\ddot{\vec{x}}$. Even if no acceleration $\ddot{\vec{x}} = 0$ is affecting the platform, it might still have a constant velocity $\dot{\vec{x}} \neq 0$ and thus a continuous drift of the position \vec{x} . A proportional and differential (PD) controller is required to stabilize this unstable system. Standard on-board avionics can not provide accurate position information. Therefore, an external positioning system, our Autonomous Localization Framework (ALF), is required. We mounted two sensing devices onto a single platform. This not only increases the accuracy and availability of the positioning but also the yaw angle Ψ can be computed by evaluating the difference of both positions. The control of the altitude z is similar (see Figure 2 bottom row): it can be done indirectly by the thrust F_t . In addition to the position controller, it also has to compensate the gravity g . Again, a PD controller is required as well as an external sensor. The necessary altitude information z can also be provided by the same positioning system.

C. Autonomous Localization Framework

Our Autonomous Localization Framework (ALF) has been designed to fulfill the task of providing a zero-effort accurate localization system for autonomously flying robots [4]. However, it is not exclusively restricted to this. The framework provides highly accurate localization features using low-cost sensors in indoor environments.

ALF has been designed as a decentralized system without the need for synchronization or global knowledge and a finite amount of energy. The typical operation scenario is depicted in Figure 1. Sensor nodes need to be capable of detecting and

communicating with their direct neighbors; no special routing information or topology is required. For localization, at least the distance to the neighbors needs to be measurable. As we desire an accurate localization system, we do not rely on rather vague RSSI measurements (although it would be possible [11]). Our ground platform [12] is equipped with an US based Time of Flight (ToF) measuring hardware. It reports distances with an observational error of less than ± 2 cm. A byproduct is a rough estimation of the angle to a neighbor ($\pm 45^\circ$).

The self-localization grid is created fully autonomously according to the following rules: First each node needs to find a “convenient” place, which is a trade-off between area coverage, accuracy, and cost. As no global knowledge exists, each node needs to find this independently from the states of its neighbors. A local decision table based approach is used. Subsequently, the node localizes itself according to the reference grid spanned by the already well placed nodes [13]. Using this initial position, the node joins the grid and enters the maintenance process based on our Advanced Mass-Spring-Relaxation (advMSR) approach [14] to further refine its position estimation. Again, only local information are used.

In this process, as typical for indoor scenarios, many Non Line of Sight (NLOS) measurements may occur. In order not to corrupt the system state, those need to be detected and blacklisted [4]. The algorithm has been designed so that it reaches convergence. At any point in time, nodes are allowed to join or leave the network (e.g. due to a depleted battery). Finally, the nodes and the generated position information are used as a reference grid for providing a localization support for customers [13], such as the autonomously flying copters. The trilateration-based scheme was specially designed to fulfill real-time requirements.

The used localization frequency is 5 Hz. As reported in the literature, under optimal conditions a stabilization of a VTOL device is possible at this frequency [2]. However, the position values have a time lag of up to 195 ms (including measurement, information transport and computation). This in combination with measurement outliers of up to ± 20 cm prevent a stable position-hold function purely based on ALF.

D. Additional Sensors

In order to enable autonomous flights nonetheless, two possibilities exist: (a) reducing the time lag by tweaking/changing the communication channel or (b) reducing the system dynamic of the platform by additional sensors. Here, we focus on the latter one. The problem we want to address here is that low-cost sensors are typically not as accurate, precise, and/or fast as required. To ensure a safe flight, measurement outliers need to be detected and handled. Similar to the approach by Zug and Kaiser [15], we use the construct of smart abstract sensors to improve the readings of the sensors.

1) *Smart Altitude Sensor*: To provide quick altitude measurements, we employ an additional US distance sensor (SRF02 made by Devantech) and synchronized it with ALF so that multiple platforms do not interfere with each other. During the flight of the quadrotor we observed five possible outcomes

for a measurement: (a) the correct distance, (b) temporary overestimations, (c) clear distance overestimations, (d) 0, or (e) no result at all. Exploiting this knowledge, we developed a smart sensor based on Equation 1: The inputs are the measurement \hat{d}_t of the physical sensor as well as the altitude \hat{z}_{pos} of the positioning system (\hat{z}_{pos} is more accurate but not available quickly enough).

$$\hat{z}_t = \begin{cases} \hat{z}_{pos} & \text{if } \hat{d}_t \notin [d_{min}, d_{max}] \\ & \text{(cases c, d or e),} \\ p \cdot \hat{d}_t + (1 - p) \cdot \hat{z}_{t-1}, & \text{if } \hat{d}_t > v_d \text{ (case b),} \\ \hat{d}_t. & \text{else (case a).} \end{cases} \quad (1)$$

Physical limitations such as the maximal climbing velocity v_d and the maximal and minimal altitude d_{max} , d_{min} are observed as well. Based on a number of experiments, we set the value p for the exponential smoothing to 0.1. This provides additional inertia and should be omitted if the reported values are plausible.

2) *Smart Position and Yaw Sensor*: This smart sensor is based upon the customer localization of ALF and returns verified 3D positions. As previously introduced, two sensing arrays are mounted onto the platform (at maximum distance in between). The platform acts like two separate sequential customers and, therefore, has two instances of the customer localization running. Based on the two resulting positions p_0 and p_1 and the assumption that the platform moved continuously during the time between the two measurements (preferable did not move at all), a yaw angle of the platform can be computed. The resulting abstract sensor uses the positions p_0 and p_1 as an input. It further knows the time Δt between the individual US measurements, and it knows the last accepted positions $p_{0,db}$ and $p_{1,db}$.

The complete procedure is depicted in Algorithm 1, where $\angle : \mathbb{R}^3 \rightarrow \Psi$ computes the horizontal angle (yaw) of a 3D vector. If one position estimation fails, the sensor tries to reconstruct this information based on previous position information and the movement of the other position, assuming a previously stabilized constant movement and yaw angle. In the very unlikely event that two positioning failures occur at once, the sensor reports this to the system.

3) *Smart Velocity Sensor*: The key idea is not to increase the position monitoring frequency of the platform but to observe

Algorithm 1 Smart Position and Yaw Sensor

```

if  $p_0$  is available and  $p_1$  is unavailable then
     $p_1 = p_{1,db} + \dot{p}_0 \cdot 2\Delta t$  # reconstruct  $p_1$ 
else if  $p_0$  is unavailable and  $p_1$  is available then
     $p_0 = p_{0,db} + \dot{p}_1 \cdot 2\Delta t$  # reconstruct  $p_0$ 
else if  $p_0$  is unavailable and  $p_1$  is unavailable then
    return failure # problem
end if
 $(\hat{x}_{pos}, \hat{z}_{pos})^T = \frac{1}{2}(p_0 + p_1)$ 
 $\hat{\Psi} = \angle(p_0 + \dot{p}_0\Delta t - p_1)$ 
return  $\hat{x}_{pos}, \hat{\Psi}, \hat{z}_{pos}$ 

```

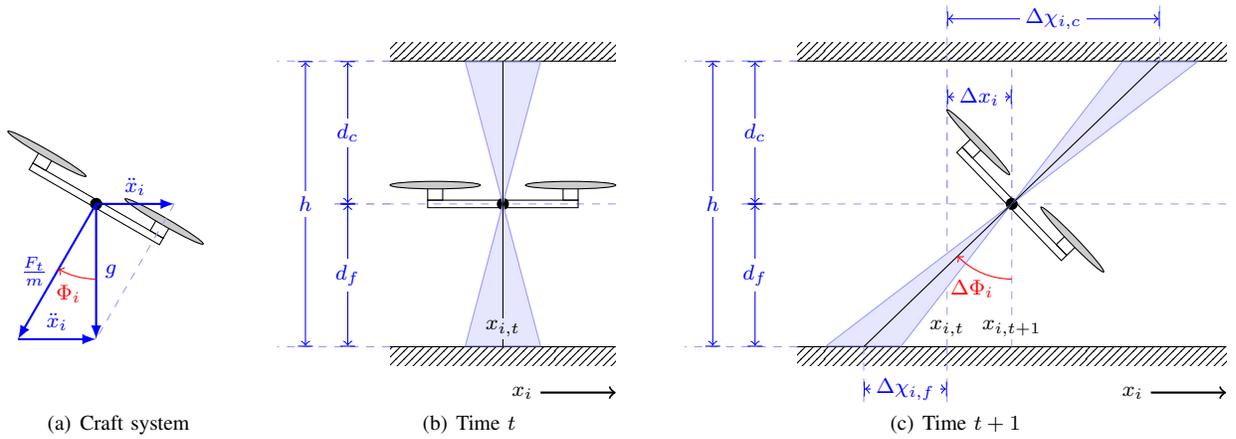


Fig. 3. Quadrotor movement

and control the velocity $\dot{\vec{x}}$ as an important system state. This reduces the chain length of the system model by one integrator, thus, it becomes much easier to stabilize the system.

A common approach is to use a two dimensional optical flow sensor [16]. These sensors are very low-cost, low-power, and light-weight (e.g., ADNS-3080 by Agilent). They are typically also installed in computer mice. Usually this sensor is facing downwards to the floor and controls the velocity $\dot{\vec{x}}$ on the forward/back x and left/right y axes. The altitude \hat{z} , which is already available from the smart altitude sensor, is needed to calculate the velocity. Furthermore, the pitch and roll angle $\vec{\Phi}$ of the robot are needed, which are more difficult to estimate. As the platform tilts to move and the sensor is rigidly mounted to the frame, it detects a relatively high movement into the inverse direction of the actual movement (see Figure 3(c)). This needs to be compensated using knowledge of the current slant. However, it is very time critical and the angles need to be very accurate, especially for large altitudes.

As a novel concept, we exploit a unique feature of indoor flights: Rooms not only have floors but also ceilings. Thus, we use one down and one up facing sensor to circumvent the need of the inclination information.

Figure 3(b) depicts one axis of a quadrotor hovering at time t . The current tilt $\Phi_{i,t}$ usually needs to be known, but the absolute value is not important. It additionally knows the distances to the floor $d_f = \hat{z}$ and to the ceiling d_c . If the room height h is known, $d_c = h - d_f$ needs not to be measured. In Figure 3(c), the quadrotor has a different tilt $\Phi_{i,t+1}$ and has also moved a distance Δx_i . The up and the down facing sensors observe a position change $\Delta \chi_{i,c}$ and $\Delta \chi_{i,f}$, respectively.

The common approach is now just to use the down facing sensor, which might report a movement into the negative direction as depicted in Figure 3(c). The movement per step (and therefore the velocity $\dot{\vec{x}}$) can be calculated as:

$$\Delta \vec{x} = (\Delta \vec{\chi}_f + \Delta \vec{\Phi} C_\Phi) \cdot d_f \cdot C_x, \quad (2)$$

where $\Delta \vec{\Phi} = \vec{\Phi}_{t+1} - \vec{\Phi}_t$; C_Φ and C_x are system constants.

Our primary motivation was that, first, accurate tilt information in general is very hard to obtain on such dynamic systems.

Either it is simply not available, as on low-cost coaxial-copters (requires additional hardware), or the accuracy/sampling rate is too low. Secondly, the inclination compensation $\Delta \vec{\Phi} C_\Phi$ is independent of the altitude.

Exploiting the down and up facing sensors (combining Equation 2 for each sensor) gives us the angle free estimation for the platform movement:

$$\Delta \vec{x} = (\Delta \vec{\chi}_f + \Delta \vec{\chi}_c) \cdot \frac{d_f d_c}{d_f + d_c} \cdot C_x. \quad (3)$$

This allows control frequencies far beyond of 1 kHz. In case of an undetectable surface (no contrasts) on one sensor, the fall back to the others in combination with the standard angle based approach (Equation 2) provides a secondary safety stage.

E. Overall Control System

A system overview including the information and control flow is depicted in Figure 4. The reference position $p_r = (\vec{x}_r, z_r)^T$ and the yaw angle Ψ_r are required input parameters. The smart sensors (depicted in red blocks) are physically connected to the frame of the platform. The blue blocks indicate the individual controllers. As the platform can instantaneously move in any direction, it is essential to know and to stabilize the rotation $\hat{\Psi}$, whereas to satisfy the desired heading Ψ_r is less important for a stable flight of the quadrotor. For the altitude correction, a PD controller has been used. To minimize the stationary error $\lim_{t \rightarrow \infty} (z_r - \hat{z})$, we added a constant value $F_g \approx mg$ (force required for hovering). *Nota bene*: Due to the high phase shift of the system model an integral part in the controller instead of F_g results in an instable system.

For the position control, we cascaded a PID and a P controller. At first, the position error $(\vec{x}_r - \hat{\vec{x}})$ needs to be rotated by Θ into the platform coordinate system (so that x and y separately affect pitch and roll). This error is subsequently transformed into a reference velocity $\dot{\vec{x}}_r$ using a very limited P controller. The following PID controller for the velocity has a very slow integral part. This eliminates the velocity error $\lim_{t \rightarrow \infty} (\dot{\vec{x}}_r - \dot{\hat{\vec{x}}}) = 0$. Thus, systematic tilting errors (due to measurement errors) can be compensated. We observed a

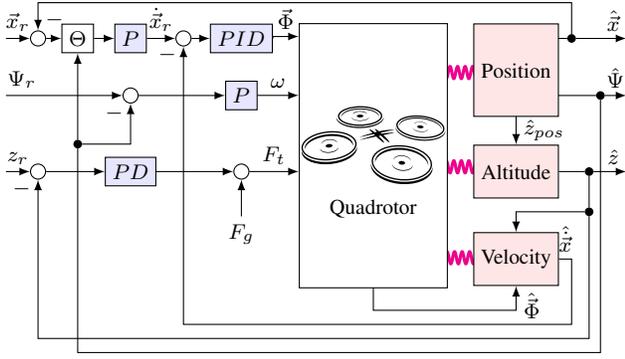


Fig. 4. System overview

significant improvement in terms of position stability compared to a system without the integral part. The actual parameters for all controller are hardware dependent and have been determined empirically.

For the whole system, we used a SunSpot sensor node running a Java virtual machine without any real-time extensions. Still, the system turned out to be stable at a control frequency of 10 Hz (and 5 Hz for position information).

IV. EVALUATION

The platform we extended in this study is a standard “MikroKopter M2” system (depicted in Figure 1). It can easily be controlled using a serial interface and can carry more than 200 g of additional weight. For safety and stability reasons, we added a styrofoam protector.

A. Velocity Estimation

In order to assess the performance of the optical tilting compensation for the velocity sensor, we conducted the following experiment. The platform is mounted to a fixed position ($\vec{x} = 0$) at an altitude of $d_f = 1$ m, $d_c = 2$ m. Different artificial rotation noises were introduced and the results of the various sensors were captured.

Figure 5(a) depicts an exemplary test run of a low frequency followed by a higher one. For improved visualization a single axis is plotted. The upper half of the graph depicts the angular velocity $\hat{\Phi}$ measured by the platform. The lower half depicts the resulting tilt compensated velocity $\hat{\vec{x}}$ of the two separate units as well as of the combined sensor. For comparison, the sampling rate of the combined sensor is significantly reduced to the rate of the individual units, which additionally require the relatively slowly provided tilting information $\hat{\Phi}$. It can be seen that because of the altitude difference $2d_f = d_c$ the ceiling sensor has a larger amplitude than the floor sensor. Obviously, the accuracy of the sensor also depends on the altitude. The signs of the curves differ, too. This is because the two sensors report movements into opposite directions. The oscillations of the platform can, slightly time shifted, also be seen in the outcome of the separate sensors. This time lag between the platform on-board gyroscope and the optical sensor is very hard to compensate on our system. The tilting information has a

different kind of noise (due to vibrations), which consequently adds additional noise to the outcome. However, the combined sensor does not need to deal with those problems and always outperforms the separate ones.

B. Positioning Sensing Time Lag

The accuracy of the self-localization framework ALF and its real-time capabilities have already been reported in [4], [13]. In this paper, we aim to address the time lag between triggering a measurement and receiving enough measurements to be able to conclude to a position \hat{p}_{pos} . The minimization of this time lag is of utmost importance for an unstable system like a hovering quadrotor. When a customer wants to get a position, it triggers an active US measurement. The sound pulse is received by the ground nodes. This duration (ToF measurement; in our case up to 25 ms) needs to be reported back to compute the position (the computational time is limited to 20 ms).

The used SunSpots use an IEEE 802.15.4 compatible Chipcon CC2420 interface for wireless communication. This protocol has not been designed for low-latency transmissions. Even though real-time extension have been presented in the literature [17], we intended to stay to the default setup to show the general applicability of ALF. Recently, we demonstrated that a custom agent based application layer protocol can collect the information in time [13]. This protocol performs significantly better than a simple broadcast based approach. In order not to reduce the localization frequency for the UAV by acting as two disjoint customers, we made two adjustments: First, an agent is able to carry two different slots at once. Secondly, the US sampling rate has been doubled. Thus, US measurements take place every 100 ms and an agent is sent out every 200 ms to collect the past two measurements.

In a second experiment, we placed three to twelve ground nodes around one localization customer, i.e., our quadrotor. We measured the required time for collecting enough measurement tuples to robustly determine the customer’s position. The results are shown in Figure 5(b) in form of boxplots indicating the median and the 25% and 75% quartiles. Three is the minimum number of reference nodes required for trilateration. The red rhombi are showing the mean values. The red dashed line shows the upper threshold of a time slot, which must not be exceeded by all means. It can be seen that the slot threshold of 200 ms is never crossed and that the duration significantly decreases for five and more nodes. This is because we stop the collection process prematurely, because five tuples are enough to robustly compute a position [13]. On the average, it takes 100 ms to 160 ms to complete the data collection.

C. Position-Hold

In the final experiment, we evaluated the position-hold ability of the overall system. We measured the ground truth using a fast optical positioning system. Figure 5(c) depicts the drift over a 2 min hovering flight (12 000 samples) of the four controlled degrees of freedom. The precision of the setup is better than ± 2 cm. However, due to the very limited sensing range of the additional system, we cannot come up with an accuracy in

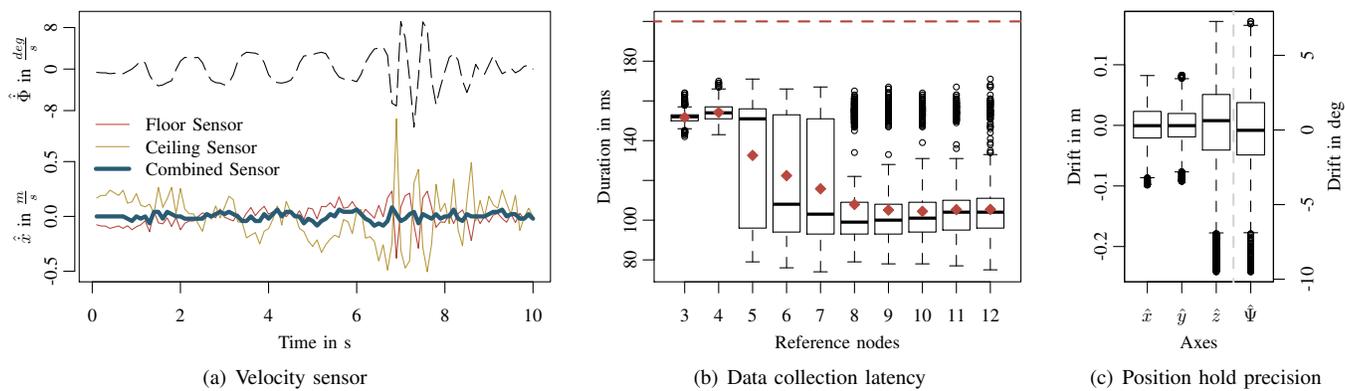


Fig. 5. Measurements

relation to the ground nodes. Thus, the depicted boxplots have been normalized to the mean value of the corresponding axis.

In this experiment, it can be seen that the translation of the position \vec{x} is below ± 10 cm. This value strongly depends on the altitude (here approx. 80 cm) and on the environmental surface contrast. The experiment was conducted in an office building. The floor is a monochrome gray carpet with a few wires on top. The ceiling is plain white with attached fluorescent tubes. Both surfaces are of low contrast. Better results can be obtained using tiles and/or wooden ceilings.

The reason for the drift in the altitude z (beside of measurement uncertainties) lays within the system model and its simplifications (see Figure 2): First, a double integrator chain is very difficult to control. For a more precise behavior, a higher sampling rate (or more accurate sensors) would be required. Secondly, the four degrees of freedom are not fully independent. As the platform compensates the drift, the thrust of the rotors is slightly redirected into a non-vertical direction (see Figure 3(a)). This reduces the vertical thrust component – the platform is descending. To counteract this, a systematic thrust compensation (based on $\Delta\vec{\Phi}$) can be implemented.

V. CONCLUSION

In this paper, we presented a low-cost sensor extension to enable indoor flights of VTOL devices. Based on noisy measurements of the raw sensor values, we show how the necessary information that allows a safe maneuvering can be derived.

For optical flow sensors, which are usually the core sensor for autonomous indoor flights, it is essential to have an environment of high contrast. By adding a second up facing flow sensor, the failure probability is not only significantly reduced but the approach also makes the inclination knowledge unnecessary (for the velocity sensor).

The combined system performance is robust in terms of error and failure. It is still fully functional, for a short period of time, if one of the three smart sensors fails.

REFERENCES

- [1] J. Rullan-Lara, S. Salazar, and R. Lozano, "UAV real-time location using a Wireless Sensor Network," in *WPNC 2011*, Dresden, Germany, April 2011, pp. 18–23.
- [2] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus, "Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz," in *IEEE ICRA 2007*. Roma, Italy: IEEE, April 2007, pp. 361–366.
- [3] S. Hochdorfer and C. Schlegel, "6 DoF SLAM using a ToF Camera: The Challenge of a Continuously Growing Number of Landmarks," in *IEEE/RSJ IROS 2010*. Taipei, Taiwan: IEEE, October 2010, pp. 3981–3986.
- [4] J. Eckert, R. German, and F. Dressler, "ALF: An Autonomous Localization Framework for Self-Localization in Indoor Environments," in *IEEE/ACM DCSS 2011*. Barcelona, Spain: IEEE, June 2011, pp. 1–8.
- [5] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *IEEE ICRA 2009*, Kobe, Japan, May 2009, pp. 2878 – 2883.
- [6] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera," in *ISRR 2011*, Flagstaff, AZ, August 2011, pp. 1–16.
- [7] C. Bills, J. Chen, and A. Saxena, "Autonomous MAV Flight in Indoor Environments using Single Image Perspective Cues," in *IEEE ICRA 2011*. Shanghai, China: IEEE, May 2011, pp. 5776–5783.
- [8] J. Conroy, G. Gremillion, B. Ranganathan, and J. Humbert, "Implementation of Wide-field Integration of Optic Flow for Autonomous Quadrotor Navigation," *Autonomous Robots*, vol. 27, no. 3, pp. 189–198, August 2009.
- [9] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart, "MAV Navigation through Indoor Corridors Using Optical Flow," in *IEEE ICRA 2010*. Anchorage, AK: IEEE, May 2010, pp. 3361–3368.
- [10] J. F. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano, "Quadrotor Using Minimal Sensing for Autonomous Indoor Flight," in *EMAV 2007*, Toulouse, France, September 2007, pp. 1–8.
- [11] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor Localization Without the Pain," in *ACM MobiCom 2010*. Chicago, IL: ACM, September 2010, pp. 173–184.
- [12] J. Eckert, K. Koeker, P. Caliebe, F. Dressler, and R. German, "Self-localization Capable Mobile Sensor Nodes," in *IEEE TePRA 2009*. Woburn, MA: IEEE, November 2009, pp. 224–229.
- [13] J. Eckert, R. German, and F. Dressler, "An Indoor Localization Framework for Four-rotor Flying Robots Using Low-power Sensor Nodes," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 2, pp. 336–344, February 2011.
- [14] J. Eckert, F. Villanueva, R. German, and F. Dressler, "Distributed Mass-Spring-Relaxation for Anchor-free Self-localization in Sensor and Actor Networks," in *IEEE ICCCN 2011*. Maui, HI: IEEE, July/August 2011, pp. 1–8.
- [15] S. Zug and J. Kaiser, "An Approach Towards Smart Fault-Tolerant Sensors," in *IEEE ROSE 2009*, Lecco, Italy, November 2009, pp. 35–40.
- [16] S. Lange, N. Sunderhauf, and P. Protzel, "A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments," in *IEEE ICRA 2009*, Kobe, Japan, June 2009, pp. 1–6.
- [17] F. Chen, R. German, and F. Dressler, "Towards IEEE 802.15.4e: A Study of Performance Aspects," in *IEEE PERCOM 2010, IQ2S Workshop*. Mannheim, Germany: IEEE, March 2010, pp. 68–73.