

Low-Cost Visual Tracking of a Landing Place and Hovering Flight Control with a Microcontroller

Karl E. Wenzel, Paul Rosset and Andreas Zell

the date of receipt and acceptance should be inserted later

Abstract The growth of civil and military use has recently promoted the development of unmanned miniature aerial vehicles dedicated to surveillance tasks. These flying vehicles are often capable of carrying only a few dozen grams of payload. To achieve autonomy for this kind of aircraft novel sensors are required, which need to cope with strictly limited onboard processing power. One of the key aspects in autonomous behaviour is target tracking. Our visual tracking approach differs from other methods by not using expensive cameras but a Wii remote camera, i.e. commodity consumer hardware. The system works without stationary sensors and all processing is done with an onboard microcontroller. The only assumptions are a good roll and pitch attitude estimation, provided by an inertial measurement unit and a stationary pattern of four infrared spots on the target or the landing spot. This paper details experiments for hovering above a landing place, but tracking a slowly moving target is also possible.

1 Introduction

There has been great interest in unmanned aerial vehicles (UAVs) over the past decade. Valavanis constitutes in [18] that UAVs should become smaller and smarter. As tracking is one of the key features for autonomous flying robots, lightweight and low-cost tracking solutions are required. In this paper, we show how a cheap infrared (IR) consumer electronic camera can be used as main sensor for stable flight control. The control algorithm is running on an onboard microcontroller. This technique paves the way for low-cost target tracking, automatic starting and landing and position estimation of other robots for miniature flying robots (MFRs). Actually, this paper focuses on hovering a miniature quadrotor in a defined position over a landing place.

The camera, the primary sensor in our configuration, is part of the Wii remote controller (informally known as the Wiimote), distributed by Nintendo (Fig. 1). It is capable of tracking infrared blobs and the integrated circuit provides the pixel position of each

Karl E. Wenzel is a PhD student and Paul Rosset is a student with the Chair of Computer Architecture, headed by professor Andreas Zell, in the Department of Computer Science, University of Tübingen, Sand 1, 72076 Tübingen, Germany {karl.e.wenzel, andreas.zell}@uni-tuebingen.de, paul.rosset@student.uni-tuebingen.de

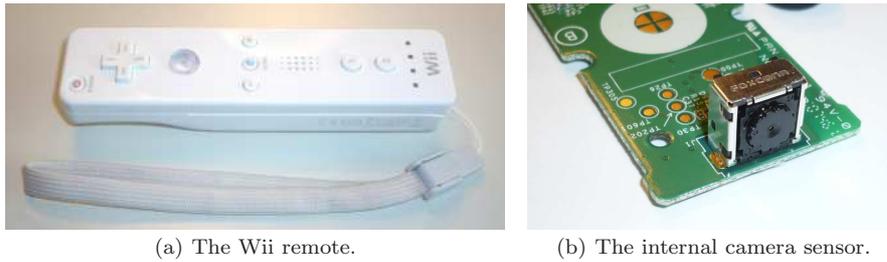


Fig. 1 The sensor of the Wii remote.

tracked blob at a high frequency.

The key idea of our approach is to track a pattern of infrared spots located at the landing place, or target, by looking downwards with a fixed camera in quasi stationary flight. The design of the pattern is a T-shape with known dimensions. So the system is able to calculate the distance, or z position and yaw angle (ψ) directly from geometric information. Assuming that the inertial measurement unit (IMU) of the aircraft provides accurate roll and pitch angles, the x and y positions can also be estimated. These four degrees of freedom are the inputs for the control loop to hover above the target. The aircraft we used is a X3D-BL Hummingbird quadrotor helicopter distributed by Ascending Technologies¹. In our experiments, the aircraft hovers indoor at a target height between 50 cm and 1 m above the landing pad. High level control behaviours could take control of the aircraft from this position. The system is capable to hover at a height of 50 cm with a root mean square deviation of 1.7 cm.

As Bouabdallah et al. demonstrated in [2], a classical PID controller has the ability to control a quadrotor in the presence of minor perturbations. That is why the control loop could be designed as a simple algorithm running on an onboard microcontroller at a high frequency.

2 Related Work

Onboard vision-based tracking and navigation have mostly been done on vehicles of a significant size and weight using comparatively expensive, industrial cameras and high performance processors. The work by Shakernia et al. [15], [16] focuses on the control of a large, single rotor helicopter (overall length 3.6 m), where the position and velocity to a planar landing pad is estimated by a vision system. Nordberg et al. [11] present vision methods for motion estimation, navigation and tracking. Their helicopter is approximately 2×1 meters and has a maximum payload of 30 kg. Saripalli et al. [13] introduce a vision-based autonomous landing algorithm, Hrabar shows path planning and stereo-based obstacle avoidance for a comparable aircraft in [8]. Frew et al. [3] present a vision-based road-following algorithm for a small, autonomous fixed-wing aircraft. This kind of aircrafts are capable of carrying personal computers with modern Gigahertz processors and accurate stereo cameras. Smaller aircrafts have to deal with very limited payload capacity. Hence, other solutions have to be found.

Roberts et al. [12] describe a low-cost flight control system for a small outdoor heli-

¹ <http://www.asctec.de/>

copter, where all processing is performed onboard. For a comparable solution on miniature UAVs, as the Hummingbird quadcopter, the system has to be even smaller and lighter. Most MFR tracking systems are based on stationary cameras or processing is done on a ground station.

Kemp [9] developed a visual control method by using a sub-miniature onboard camera of only 9 g weight. Guenard et al. [4] present a visual servo control and Herisse et al. [7] describe a hovering flight and vertical landing control using optical flow. Calculations are transferred to the ground station in these projects, which allows to control a miniature quadcopter, but leads to restrictions in autonomy.

Gurdan et al. [5] did flight experiments by using an external motion capture system. They were able to follow a trajectory with a maximum deviation of 10 cm from the desired position. Watanabe et al. [19] propose an assistant and training system for controlling unmanned helicopters in a flight field monitored by stationary cameras. Controlling UAVs without an onboard camera, but tracked by a motion capture system, leads to accurate positioning at a high control frequency, as cameras of a high quality and fast computers can be used. However, they can only operate in a few cubic meters, which is not practicable for most autonomous operations.

Mak et al. [10] describe a six degrees of freedom visual tracking system for a miniature helicopter using only three onboard LEDs and a single on-ground camera. Hay et al. [6] explain how optical tracking can be done using two Wiimotes and achieve excellent results with a minimum on hardware costs. These solutions deal with a minimum of hardware demands, but still depend on a permanent connection to the ground station when used for UAV control.

Our approach is different from those described above, as we are using the IR camera from a Wiimote onboard. The approach is only dependent on four IR diodes as external landmarks, which can easily be integrated in a landing pad and can be regarded as landing strip lights.

3 The Characteristics of the Wii Remote Infrared Camera

The Wiimote is a Bluetooth-compatible controller, designed for interfacing the Nintendo Wii game console. The device provides a three-axis accelerometer, 12 digital buttons and an IR camera tracker as sensors. An expansion port can be used to connect additional input devices, like the Nunchuck controller. The price of 40 € is relatively inexpensive, considering the internal components.

The optical sensor is normally used in conjunction with a strip containing two infrared spots to determine the position and orientation of the controller to control a cursor on the screen. Recent publications benefit from the specialities of the Wiimote as it can be connected to a PC by using Bluetooth. Sreedharan et al. [17] analyses the motion by interpreting the acceleration sensor information. Shou et al. [14] integrates this controller into a game environment in a multi-wall virtual reality theatre.

As size and weight is important for miniature flying robots and the Bluetooth connection is not needed when operating onboard, we detached the camera from the controller. The sensor weight of 0.4 g at a dimension of $8 \times 8 \times 5 \text{ mm}^3$ makes it an ideal MFR onboard sensor. The multi-object tracking engine (MOT sensor), manufactured by PixArt Imaging is capable of blob tracking of up to four IR sources. By eight times subpixel analysis, the native resolution of 128×96 pixel is scaled up to 1024×768 pix-

els. The sensor supports different modes, which differ in sensitivity and information about the infrared blobs. The complete information includes the blob size, intensity and the bounding box. For our use, the basic mode, providing only the position, is sufficient. The horizontal field of view is approximately 45° , and the refresh rate of 100 Hz in Bluetooth mode is adequate for fast optical tracking. When operating in I²C bus mode, frequencies of 250 Hz and higher can be achieved.

Just a few electronic components are required to integrate the sensor in a microcontroller circuit. The camera runs at 3.3 V and needs an external synchronisation of 24 Mhz. The I²C bus address and the communication protocol is known, so the data can be obtained easily by an I²C host via polling.

4 The UAV System

The AscTec X3D-BL Hummingbird quadcopter is 53 cm in diameter, weighing only approximately 0.5 kg. A 2.1 Ah lithium-polymer accumulator powers the electronic motors and the onboard electronics. Its flight time of up to 23 minutes depends on the additional payload and the flight maneuvers.

The Hummingbird platform comes with a three-axis gyroscope, an accelerometer, a compass module, a GPS sensor and pressure sensor. The sensors, the microcontroller and the flight control algorithm running at 1 kHz are fast enough to stabilise the quadcopter sufficiently, making flying relatively easy compared to common model helicopters. The pressure sensor allows keeping a specific height, and compass and GPS are used to keep a given position in the air or to fly to waypoints, transmitted by the ground station connected via a ZigBee transmission module. So the Hummingbird provides basic outdoor autonomy without any additional hardware, but also permits high level applications to control the flight and to connect additive devices via two serial ports.

By varying the speed of the four motors, the aircraft can tilt, turn and change its height. A detailed analysis of the advantages of quadrotors and a description of their dynamic model can be found in [1].

As we did experiments with other aircrafts, we decided to program a separate board, populated with an Atmel ATmega 644P microcontroller. All processing is done on this board and the microcontroller acts as a gateway to send sensor information to the base station. The 8-bit microcontroller is clocked at 14 MHz, which enables serial communication with the quadcopter and the base station. While the first serial port is communicating with the robot, the second port is connected to the ZigBee module which sends messages to a ground station. The base station is used for monitoring the current pose estimation and control status and can be used for changing control loop parameters.

The framework for the microcontroller is able to communicate with different control protocols, so our tracking can be applied to various quadcopters. The Hummingbird has proven to provide roll and pitch estimations of sufficient quality, leading to accurate position estimations.

The camera is fixed in the center of the quadcopter frame, so the position of the aircraft can easily be calculated without additional translations to compensate the position offset.

5 Retrieving the Pose From Camera and IMU

This section describes how the current position vector $\mathbf{p} = (x, y, z, \psi)^T$ is estimated. x, y and z are the Cartesian coordinates of the landing place origin, relative to the camera and ψ is the orientation in yaw. The position vector \mathbf{p} can be estimated by using the sensor information, provided by the Wiimote camera combined with the roll and pitch angles provided by the IMU.

5.1 Pattern Analysis

By running the Wiimote camera in basic information mode, the position of four points are transmitted to the microcontroller. The data is forwarded to the base station, where an image of the current view can be visualised. Fig. 2(b) shows a typical image, received at the base station while the quadcopter was in 50 cm height. The four dots are linked with colored lines, to get a faster visual feedback for the operator. The

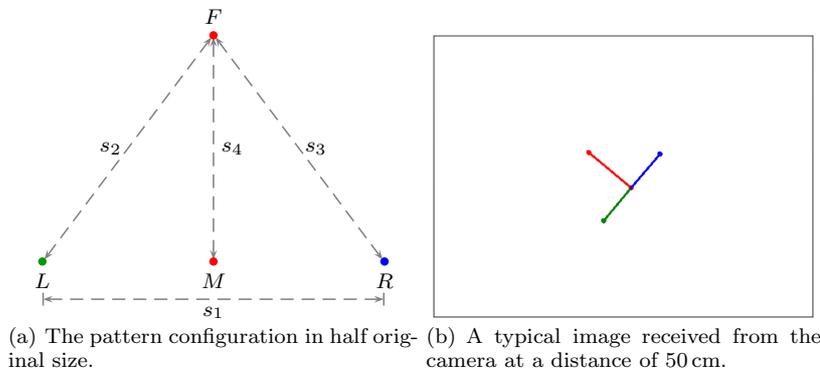


Fig. 2 The pattern, attached to the landing place.

pattern we constructed measures 90 mm from the left to the right IR spot (s_1), and 60 mm from the middle to the front IR spot (s_4), which fits easily in a standard circuit board. Each spot is represented by a single 940 nm wavelength infrared LED. This configuration has proven to be a good size for indoor landing pad tracking, where the pattern must be recognised at a relatively close distance. Larger patterns would allow for more precise tracking at a larger distance, but would no longer fit in the field of view when getting closer. Our pattern can be completely covered by the camera from a distance above 150 mm. However, when the aircraft is moving, single spots of the pattern usually get lost due to the inclination, even at larger distances.

To get unambiguous position information, the disordered points **F** (front), **L** (left), **M** (middle) and **R** (right) first have to be identified (Fig. 2(a)). Several techniques could be used to sort the points. Our approach focuses on fast processing on a microcontroller, which means that we need to avoid floating point operations as well as trigonometric functions.

Given a set of four points, we calculate the distances of all points to lines, defined by two of the remaining three points. As **M, L, R** are lying in one line, a combination

of $\mathbf{M}, \mathbf{L}, \mathbf{R}$ leads to the minimum point to line distance. The point not used in this combination can be indirectly identified as \mathbf{F} .

The maximum distance between two of the three points left identifies \mathbf{M} to lie between \mathbf{L} and \mathbf{R} . Identifying these points is sufficient for our tracking control.

For a complete identification, \mathbf{L} and \mathbf{R} should also be identified. Let \mathbf{A} and \mathbf{B} be the remaining last points to recognise. By using $\mathbf{A} = (a_x, a_y)^T$, $\mathbf{M} = (m_x, m_y)^T$ and $\mathbf{F} = (f_x, f_y)^T$, the following equation identifies \mathbf{A} unambiguously as \mathbf{L} (s is positive) or \mathbf{R} (s is negative):

$$s = \text{sgn}[(a_x - m_x)(f_y - m_y) - (a_y - m_y)(f_x - m_x)] \quad (1)$$

If all points have been identified, the position vector \mathbf{p} of the quadcopter relative to the landing pad can be determined.

5.2 Estimating Yaw and Distance

The yaw angle ψ and the z distance can be calculated directly from geometric information, while the x and y position estimation depends on accurate roll and pitch angle information, provided by the IMU. The yaw angle ψ is measured relative to the straight line, defined by \mathbf{M} and \mathbf{F} and can be easily calculated:

$$\psi = \text{atan2}(m_x - f_x, m_y - f_y) \quad (2)$$

The viewing angle per pixel of the camera in radians is given as $\rho = 7.61 \cdot 10^{-4}$. Let s_i be the physical space between two LEDs and d_i be the pixel distance of the corresponding points. The physical distance z_i from the camera to the pattern can be calculated as:

$$z_i = \frac{s_i}{\tan(d_i \rho)} \quad (3)$$

The following points were used for distance measuring in our experiments: $d_1 = |\mathbf{L} - \mathbf{R}|$, $d_2 = |\mathbf{L} - \mathbf{F}|$, $d_3 = |\mathbf{R} - \mathbf{F}|$, $d_4 = |\mathbf{M} - \mathbf{F}|$. We got satisfactory results in distance measuring with a stationary camera by only using d_1 . However, when flying, disturbances cause noise in subpixel analysis and notable errors in distance calculation. By using the mean value, the error caused by pixel noise can be decreased:

$$z = \frac{1}{4}(z_1 + z_2 + z_3 + z_4) \quad (4)$$

5.3 Estimating the x and y Position

This section describes how the x and y position of the aircraft can be estimated, and why an IMU is required with the presented pattern.

Since the aircrafts desired position is directly above the pattern, the view angle to the pattern is limited by the small roll and pitch angles required for position corrections. These angles are usually between $\pm 3^\circ$ during controlled flight. In such an angular attitude, as shown in Fig. 3(b), the pattern appears in a displaced position in the camera image. Let this angle be denoted by *geometric angle*.

If the camera orientation is known, the x and y position relative to the center of the pattern can be estimated. Any camera displacement also leads to distortion of the

pattern, which permits to estimate the attitude relative to the pattern. Experiments have proven that an estimation with a resolution of only 5° at a distance of 80 cm is possible. However, the notable distortion of the used pattern only counts a few pixel (depends on the distance) and is ambiguous for the pitch angle. A positive pitch angle leads to the same shortened d_4 as a negative angle would do. These limitations make the use of the IMU necessary. A modified pattern, optimized for distortion analysis could allow for controlling without an IMU.

A displacement of the aircraft means that the physical angles provided by the IMU differ from the geometric angles. By combining the geometric and the physical roll and pitch angles, the x and y position relative to the pattern can be calculated in respect to the current z position. Let \mathbf{O} be the origin position of the landing pad, defined by

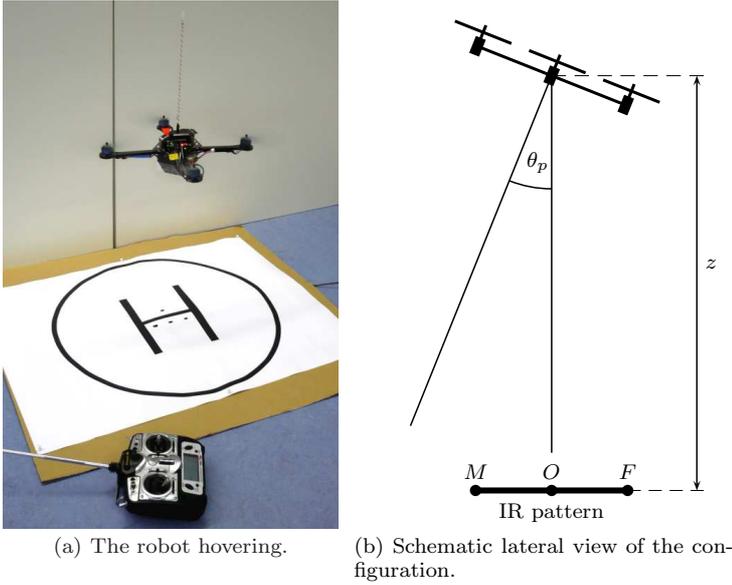


Fig. 3 The robot autonomously hovering above the pattern.

$\mathbf{O} = 1/2(\mathbf{M} + \mathbf{F})$. and \mathbf{C} be the middle of the image, defined by the camera resolution of 1024×768 pixels: $\mathbf{C} = (c_x, c_y)^T = (512, 384)^T$. The physical roll φ_p and pitch θ_p angles are provided by the onboard IMU and requested before each position estimation. The geometric roll φ_g and pitch θ_g angles, representing the angles of vision, are calculated by:

$$\theta_g = (o_x - c_x)\rho \quad (5)$$

$$\varphi_g = (o_y - c_y)\rho \quad (6)$$

Potential inaccuracies in camera positioning are compensated with the calibration angles φ_c and θ_c . The calibration angles only have to be measured once after attaching the camera to the aircraft. By correcting the physical angles of the aircraft with the calibration angles and combining them with the visual angles, the real displacement of

the landing pad can be estimated:

$$\varphi = \varphi_g + \varphi_p + \varphi_c \quad (7)$$

$$\theta = \theta_g + \theta_p + \theta_c \quad (8)$$

$$x = \tan(\varphi)z \quad (9)$$

$$y = \tan(\theta)z \quad (10)$$

6 Flight Control Algorithm

After successfully estimating the current position $\mathbf{p} = (x, y, z, \psi)^T$, the aircraft can be controlled to hover at a defined height above \mathbf{O} . The algorithm is inspired by Gurdan et al. [5], where four independent controllers were operating.

The only variable in our controller is the desired height (z_{des}) above the target z . All other elements of \mathbf{p} should remain zero.

6.1 Tracking and Flight Control Method

An overview of our system including pattern recognition, position estimation and flight controller is shown in Fig. 4. The aircraft provides an interface to flight control, which uses the output of our four independent PID controllers. The four motors are controlled on a lower level by the internal controllers of the aircraft.

The control loop is currently performed with a frequency of 20 Hz. The controller

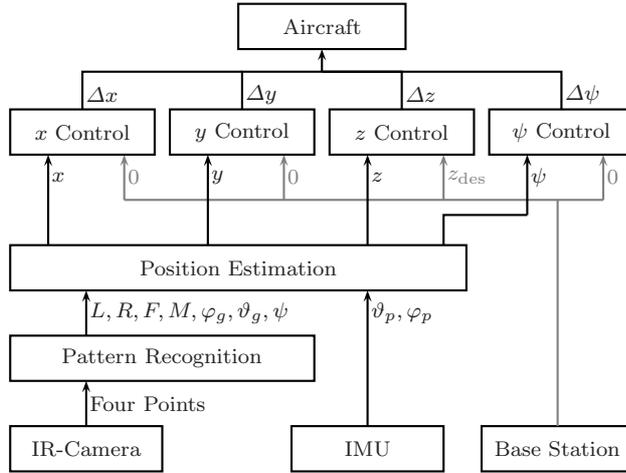


Fig. 4 Global picture of the control system. The desired height (z_{des}) can be varied by the base station.

requires an actual pose estimate by the IMU, whose request lasts approximately 30 ms in total. Additional 10 ms are required to send sensor information to the base station, where the current status is monitored and displayed. The remaining 10 ms are available

for retrieving sensor information, running the control algorithm and receiving data from the base station. A considerably higher control frequency would be possible with a faster IMU request.

6.2 Height Controller

The thrust control value, needed to hover at a desired height, changes, depending on the actual payload and battery charge. This is why the thrust controller is divided into two parts, an accumulator for the base value and a fast controller for position control. We obtained good results by counting up whenever the robot is below the desired height for some cycles, and down otherwise. The fast response controller is implemented as a PID loop with K_D being the biggest component. Figure 5 shows the structure of the height controller.

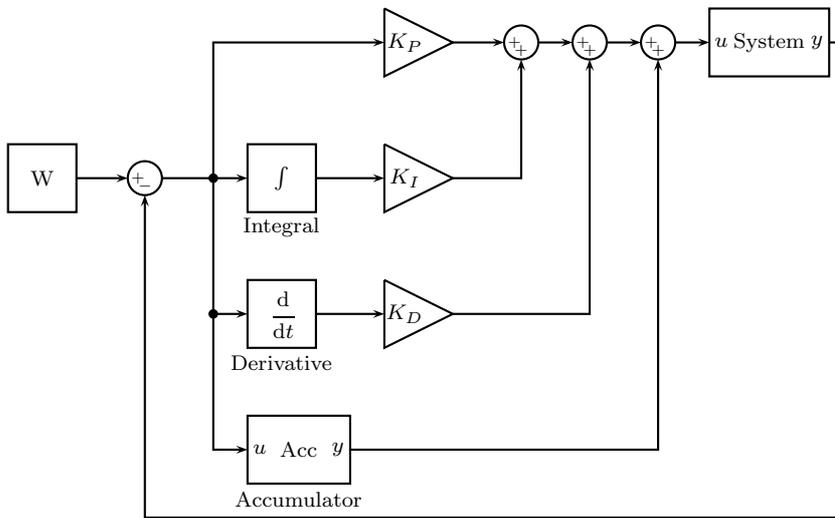


Fig. 5 The height controller.

6.3 Roll/Pitch and Yaw Controller

The x and y controllers are identical and were harder to derive, since the behaviour response is not proportional to horizontal speed but to rotational velocities. A predictive control is required to achieve a stable position hold. By designing a cascaded control loop where not only the speed but also the acceleration is highly weighted, the behaviour obtains the desired prognostic ability. Figure 6 shows the structure of the x and y controller.

The yaw controller is implemented as a PID loop with a large K_P and a small K_I .

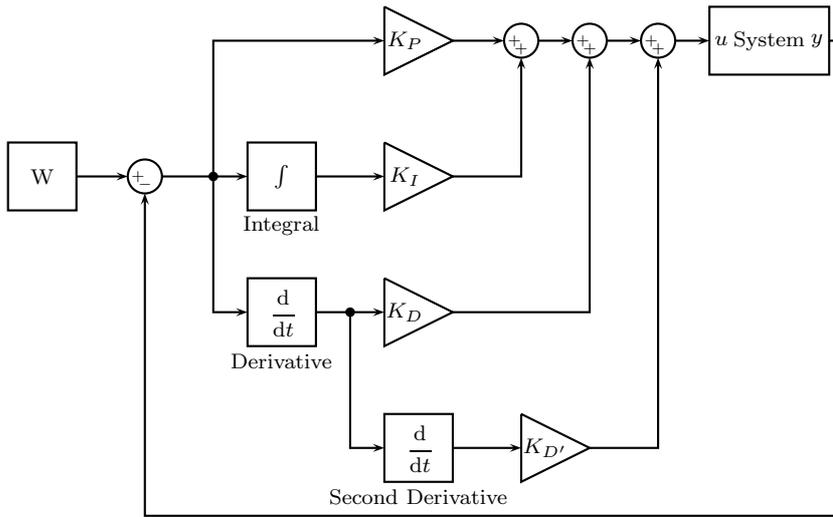


Fig. 6 The x and y controller.

7 Experimental Results

Experiments with different quadcopters have shown that an accurate attitude estimate is essential for reliable position estimation with our approach. The following example data represents typical experimental results and should demonstrate the performance of the system.

7.1 Attitude Estimation

To validate the quality of the attitude estimation provided by the IMU, a slightly different configuration of the presented system was used. A Wiimote camera, heading to the front, allows for easily designating the roll angle of the aircraft from image data. By using a bigger pattern, the angle estimation is very accurate and can be used as reference. Comparing the angles measured with camera and IMU proves good quality for the autopilot attitude estimation. We measured a standard deviation of approximately 3° , while the error depends on the angular velocity. These results are based on the micro-electro-mechanical systems (MEMS) gyrotor sensor of the autopilot IMU. By contrast, IMUs using piezo gyrotor sensors usually have to struggle with worse sensor data. The quadrotors with piezo gyrotor sensors we tested were not capable of following the angle measured by the Wiimote camera for a longer time.

7.2 Flight Control Accuracy

The Hummingbird quadcopter comes with a GPS position controller and an air-pressure sensor for height control. These functions were been disabled during our indoor experiments. The yaw angle is controlled by the internal magnetic compass of the

Table 1 Controller characteristics of five minutes flight at 50 cm height

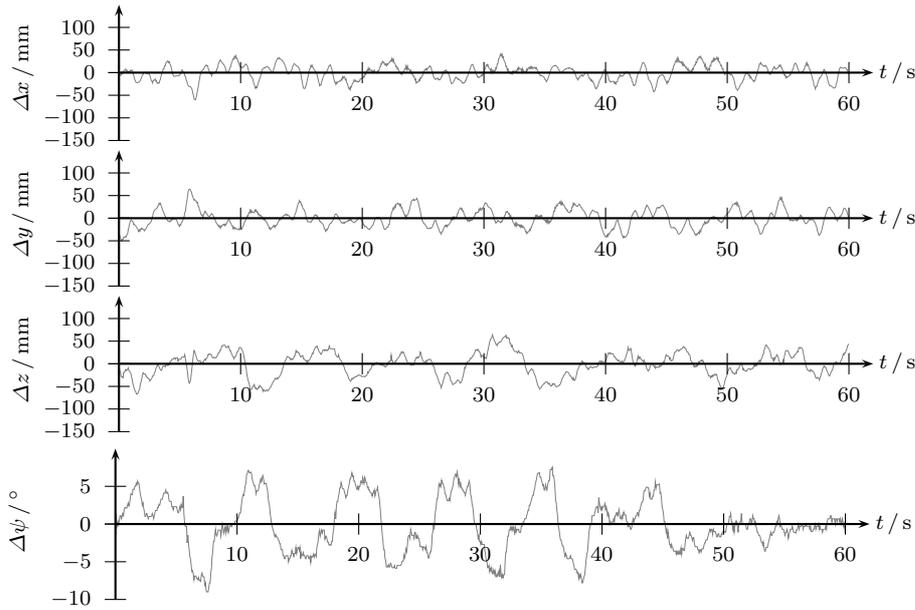
	$\Delta x/\text{mm}$	$\Delta y/\text{mm}$	$\Delta z/\text{mm}$	$\Delta\psi/^\circ$
Minimum	-60.00	-54.00	-80.00	-9.10
Maximum	54.00	64.00	87.00	7.90
Peak to peak	114.00	118.00	167.00	17.00
Mean	-0.02	-0.09	-1.16	-0.01
Standard deviation	17.54	18.55	27.78	3.03

Table 2 Controller characteristics of five minutes flight at 1 m height

	$\Delta x/\text{mm}$	$\Delta y/\text{mm}$	$\Delta z/\text{mm}$	$\Delta\psi/^\circ$
Minimum	-156.00	-128.00	-109.00	-10.00
Maximum	130.00	117.00	155.00	13.20
Peak to peak	286.00	245.00	264.00	23.20
Mean	-0.05	-0.12	-0.96	0.10
Standard deviation	39.35	38.50	33.99	2.80

quadrocopter and can not be disabled. However, experiments with another quadrocopter have proven that yaw control is uncritical. The results shown here are with assistance of the internal yaw controller of the quadrocopter.

Following a large number of flights and retrieval of working parameters, the system

**Fig. 7** Errors of one minute flight at 50 cm height.

has proven its applicability for short distance indoor landing place tracking and flight control. Tables 1 and 2 show the controller characteristics of flights in 50 cm and 1 m

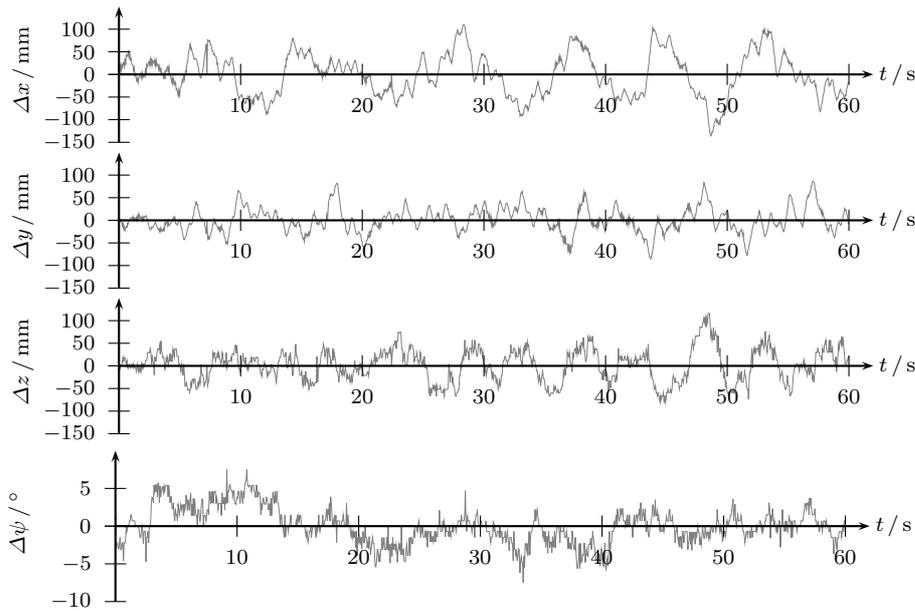


Fig. 8 Errors of one minute flight at 1 m height.

height above the landing place, 5 minutes each.

As mentioned in Section 5.3, the estimation of roll and pitch is essential for accurate x and y approximation. While the standard deviation of Δz in 50 cm and 1 m is rather comparable, one can notice a significant change in x and y position control. The yaw angle ψ is affected by balancing manoeuvres in flight and thus depends on the position changes in x -, y - and z -directions. In approximately 60% of the flight time at 50 cm height, an accuracy of ± 1.5 cm in x and y position could be achieved. Details of the positioning probability are shown in Fig. 9. A higher control frequency would even improve the stabilisation, but is prevented by the slow response time of 30 ms at which the quadcopter provides the pose estimation in the current configuration. A detailed record of one minute of the flights is shown in Figs. 7 and 8. The plots show the position estimate sent to the base station while flying. The sensor data is smooth and nearly noise-free in a short distance. Outliers, caused by external influences as infrequent reflective lights, can easily be filtered. When operating in a height of 1 m, errors in distance approximation and especially impreciseness in roll and pitch lead to a rougher curve.

Some oscillations at different frequencies still remain in the control sequence. The ideal PID parameters depend on the working distance to the pattern. Hence, finding optimal parameters is a problem. A deviance in one axis leads to displacements in other directions. Accurate position hold can only be achieved by a combination of stable controllers for each axis. An automated parameter optimization could help to find even better parameters. Anyway, increasing the control frequency shows greater promise for improved position hold capability. One problem which we encountered are reflections of sunlight, leading to wrong pattern interpretation, as sunlight contains a notable fraction of infrared light.

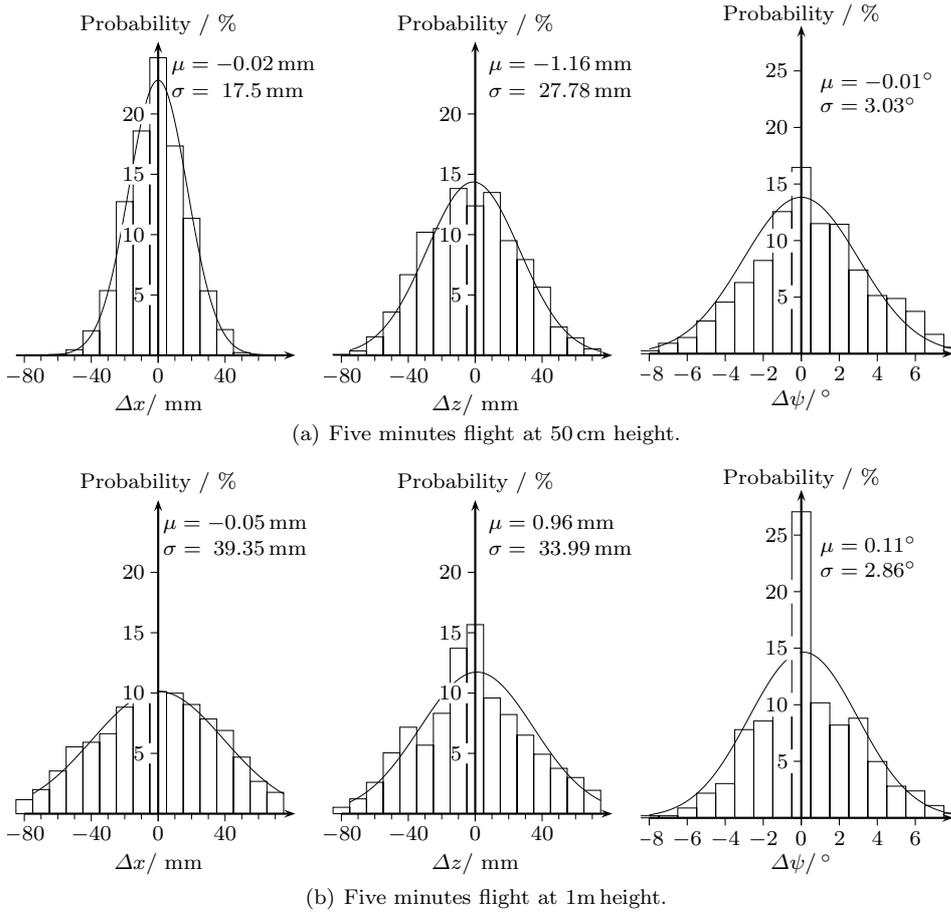


Fig. 9 Probability of positioning while tracking.

8 Conclusion and Future Work

The Wiimote camera has proven to provide accurate tracking of infrared blobs and the infrared filter avoids most disturbances caused by external light sources, except sunlight. The pixel position information can easily be interpreted as an unambiguous pattern on a target. A microcontroller is capable of estimating the position to the target and controlling a miniature flying robot in hovering flight with our approach. The field-of-view of the Wiimote camera is big enough for adjustment moves and tracking even in small distances.

The deviation from the desired position, which we obtain with our control algorithm, is small enough for quasi stationary flight for a longer time in an indoor environment. High level autonomy tasks could start from this position.

The distance to the target is limited, but by enlarging the dimension of the pattern and increasing the IR light emission by using multiple IR LEDs per point or stronger LEDs, the operating distance can be increased. However, an accurate roll and pitch

estimate, provided by the IMU of the aircraft is essential. A small angular error leads to inadmissible position approximation. A second Wiimote camera, or fusion with other sensors would lead to better positioning at a larger distance.

By accelerating the IMU request, or integrating the tracking control loop in the internal program of the UAV, the control frequency could be increased considerably. A higher control frequency would advance the control accuracy and a customised control design would eliminate most of the remaining oscillations. However, the current frequency of 20 Hz is sufficient for robust hovering control.

A pattern where not all LEDs are lying in one layer, but \mathbf{M} is raised in positive z direction, allows for better analysis. Assuming the aircraft operates in limited space above the landing place, such a 3D-pattern allows for estimating the x and y position in addition. This could lead to a flight control, independent from the IMU, if the achieved accuracy is sufficient.

References

1. Bouabdallah, S., Murrieri, P., Siegwart, R.: Design and Control of an Indoor Micro Quadrotor. In: IEEE International Conference on Robotics and Automation (ICRA) (2004)
2. Bouabdallah, S., Noth, A., Siegwart, R.: PID vs LQ control techniques applied to an indoor micro quadrotor. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 2451–2456 (2004)
3. Frew, E., Mcgee, T., Kim, Z., Xiao, X., Jackson, S., Morimoto, M., Rathinam, S., Padiyal, J., Sengupta, R.: Vision-based road-following using a small autonomous aircraft. In: IEEE Aerospace Conference, vol. 5, pp. 3006–3015 (2004)
4. Guenard, N., Hamel, T.: A practical visual servo control for a unmanned aerial vehicle. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1342–1348 (2007)
5. Gurdan, D., Stumpf, J., Achtelik, M., Doth, K.M., Hirzinger, G., Rus, D.: Energy-efficient autonomous four-rotor flying robot controlled at 1 kHz. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 361–366. Roma, Italy (2007)
6. Hay, S., Newman, J., Harle, R.: Optical tracking using commodity hardware. In: 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR), pp. 159–160 (2008)
7. Herisse, B., Russotto, F.X., Hamel, T., Mahony, R.E.: Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 801–806 (2008)
8. Hrabar, S.: 3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs. In: IEEE International Conference on Intelligent Robots and Systems, pp. 807–814 (2008)
9. Kemp, C.: Visual control of a miniature quad-rotor helicopter. Ph.D. thesis, Churchill College, University of Cambridge (2006)
10. Mak, L.C., Furukawa, T.: A 6 DoF visual tracking system for a miniature helicopter. In: 2nd International Conference on Sensing Technology (ICST), pp. 32–37 (2007)

-
11. Nordberg, K., Doherty, P., Farnebäck, G., Forssén, P.E., Granlund, G., Moe, A., Wiklund, J.: Vision for a UAV helicopter. In: International Conference on Intelligent Robots and Systems (IROS), workshop on aerial robotics. Lausanne, Switzerland (2002)
 12. Roberts, J., Corke, P., Buskey, G.: Low-cost flight control system for a small autonomous helicopter. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 546–551. Taipei, Taiwan (2003)
 13. Saripalli, S., Montgomery, J.F., Sukhatme, G.S.: Vision-based autonomous landing of an unmanned aerial vehicle. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2799–2804. Washington, DC, USA (2002)
 14. Schou, T., Gardner, H.J.: A Wii remote, a game engine, five sensor bars and a virtual reality theatre. In: OZCHI '07: Proceedings of the 2007 conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design, activities, artifacts and environments, pp. 231–234. ACM, New York, NY, USA (2007)
 15. Shakernia, O., Ma, Y., John, T., Sastry, K.S.: Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control. *Asian Journal of Control* **1**, 128–145 (1999)
 16. Sharp, C.S., Shakernia, O., Sastry, S.S.: A vision system for landing an unmanned aerial vehicle. In: IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea, pp. 1720–1727 (2001)
 17. Sreedharan, S., Zurita, E.S., Plimmer, B.: 3d input for 3d worlds. In: OZCHI '07: Proceedings of the 2007 conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design, activities, artifacts and environments, pp. 227–230. ACM, New York, NY, USA (2007)
 18. Valavanis, K.P. (ed.): *Advances in Unmanned Aerial Vehicles. State of the Art and the Road to Autonomy*. Springer (2007)
 19. Watanabe, K., Iwatani, Y., Nonaka, K., Hashimoto, K.: A visual-servo-based assistant system for unmanned helicopter control. In: IEEE International Conference on Intelligent Robots and Systems, pp. 822–827 (2008)