

---

# Autonomous Landing of an Unmanned Helicopter based on Vision and Inertial Sensing

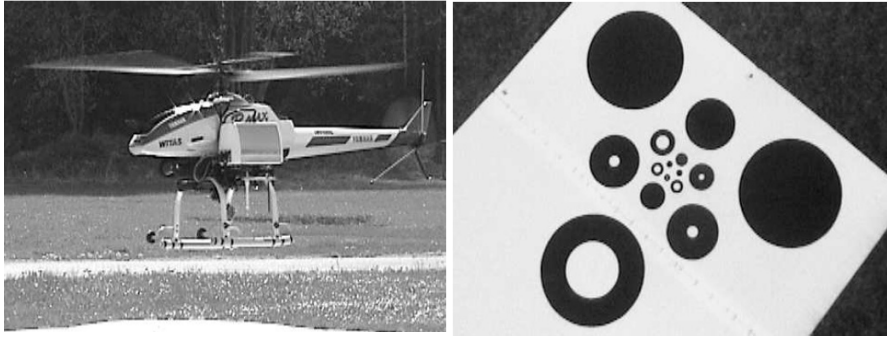
Torsten Merz, Simone Duranti, and Gianpaolo Conte

Department of Computer and Information Science  
Linköping University, SE-58183 Linköping, Sweden

**Abstract.** In this paper, we propose an autonomous precision landing method for an unmanned helicopter based on an on-board visual navigation system consisting of a single pan-tilting camera, off-the-shelf computer hardware and inertial sensors. Compared to existing methods, the system doesn't depend on additional sensors (in particular not on GPS), offers a wide envelope of starting points for the autonomous approach, and is robust to different weather conditions. Helicopter position and attitude is estimated from images of a specially designed landing pad. We provide results from both simulations and flight tests, showing the performance of the vision system and the overall quality of the landing.

## 1 Introduction

Many autonomous landing systems for Unmanned Aerial Vehicles (UAVs) are based on GPS and a dedicated close range sensor for accurate altitude measurement (radar altimeter, sonar, infrared or theodolites). However, in urban environments buildings and other obstacles disturb the GPS signal and can even cause loss of signal (multi-path effects, EM noise due to active emitters). Once the GPS signal is lost, the dead reckoning capability of affordable on-board inertial navigation systems does not allow precision navigation for more than few seconds, before diverging. Hence the need of a robust observation of the position: a vision system is self-contained, not jammable, and provides in the proposed implementation a position measurement one order of magnitude more accurate than standard GPS (cm accuracy or better). Estimating velocity from vision is difficult due to limited image frame rate and sensor resolution. In the proposed method velocity is estimated accurately and robustly by fusing vision position with the measurements of inertial sensors that usually belong to the standard instrumentation of an UAV for stabilization. The problem is to develop: (a) a vision system with a sufficient operating range to allow robust pose estimation from a reasonable distance at a sufficient rate with low latency using a landing pad of minimal size; (b) a method to fuse these data with inertial measurements; (c) a suitable flight controller. In an autonomous landing system all components have to match each other. For instance, for calculating vision estimates a



**Fig. 1.** The WITAS helicopter descend-  
ing to the landing pad. **Fig. 2.** Landing pad with reference pat-  
tern seen from the on-board camera.

proper trade-off between accuracy, range, latency, and rate has to be found optimizing the overall performance of the system.

Our method requires a special landing pad (Fig. 2). As unmanned helicopters usually operate from a designated home base this is not a real constraint. A precise and fast pan/tilt camera is used to extend the range of the vision system and decouple the helicopter attitude from the vision field. We developed a single camera solution, as multi-camera systems make the system more complex and expensive and don't offer significant advantages when using known landmarks. For the experimentation we used a helicopter platform (Fig. 1) which had been developed in the WITAS project [2,1].

Vision-based control of small size autonomous helicopters is an active area of research. A good overview of the state-of-the-art can be found in [7]. Our contribution to the landing problem consists of: (a) many demonstrated landings with a system only based on images from a single camera and inertial data using off-the-shelf computer hardware; (b) a wide envelope of starting points for the autonomous approach; (c) robustness to different weather conditions (wind, ambient light); (d) a quantitative evaluation of the vision system and the landing performance.

## 2 Vision System

The vision system consists of a camera mounted on a pan/tilt unit (PTU), a computer for image processing, and a landing pad (a foldable plate) with a reference pattern on its surface. In this section, we explain the design of the reference pattern, describe the image formation, and present the image processing algorithm.

The reference pattern is designed to fulfill the following criteria: fast recognition, accurate pose estimation for close and distant range, minimum size, and minimal asymmetry. We have chosen black circles on white background as they are fast to detect and provide accurate image features (Fig. 2). From

the projection of three circles lying on the corner points of an equilateral triangle the pose of an object is uniquely determined, assuming all intrinsic camera parameters are known. Circles are projected as ellipses, described by the center point  $\mathbf{u}_e$ , the semi-major axis  $l_a$ , the semi-minor axis  $l_b$ , and the semi-major axis angle  $\theta_e$ . The pose of the landing pad with respect to the camera coordinate system is estimated by minimizing the reprojection error of the extracted center points and semi-axes of the three ellipses. We use five circle triplets of different size (radius 2 to 32 cm, distance 8 to 128 cm) with common center point to achieve a wide range of possible camera positions. Each triplet is uniquely determined by a combination of differently sized inner circles.

A point  ${}^p\tilde{\mathbf{x}}$  in the landing pad frame is projected on the image plane as follows:

$$\tilde{\mathbf{u}} = \mathbf{P} {}^p\tilde{\mathbf{x}} = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} {}^p\mathbf{R} & {}^c\mathbf{t}_p \\ \mathbf{0}_3^T & 1 \end{pmatrix} {}^p\tilde{\mathbf{x}} \quad \tilde{\mathbf{u}} \in \mathcal{P}^2 \quad {}^p\tilde{\mathbf{x}} \in \mathcal{P}^3 \quad (1)$$

The extrinsic camera parameters are given by the three Euler angles of the rotation matrix  ${}^p\mathbf{R}$  and the three components of the translation vector  ${}^c\mathbf{t}_p$ . We use a camera model with the following intrinsic parameters: "focal lengths"  $\alpha_u$  and  $\alpha_v$  in pixels, principal point  $(u_0, v_0)$ , and four lens distortion coefficients. All intrinsic parameters are calibrated using Bouguet's calibration toolbox [3]. A conic in  $\mathcal{P}^2$  is the locus of all points  $\tilde{\mathbf{u}}$  satisfying the homogeneous quadratic equation  $\tilde{\mathbf{u}}^T \mathbf{C} \tilde{\mathbf{u}} = 0$ . The transformation of a circle  $\mathbf{C}_p$  on the landing pad into an ellipse  $\mathbf{C}_i$  in the image plane is given by[4]:

$$\mathbf{C}_i = (\mathbf{H}^{-1})^T \mathbf{C}_p \mathbf{H}^{-1} \quad (2)$$

The homography matrix  $\mathbf{H}$  is the projection matrix  $\mathbf{P}$  without third column ( $z = 0$ ). We calculate the ellipse center and axes from  $\mathbf{C}_i$  and represent the parameters in a common feature vector  $\mathbf{c}$ .

Fig. 3 shows a data flow diagram of the vision system. Round-edged boxes represent image processing functions, sharp-edged boxes indicate independent processes, and dashed lines show trigger connections. Closed contours are extracted from gray-level images using a fast contour following algorithm with two parameters: edge strength and binarization threshold. The latter is calculated from the intensity distribution of the reference pattern. In the contour list we search for the three biggest ellipses belonging to a circle triplet. Ellipse parameters are estimated by minimizing the algebraic distance of undistorted contour points to the conic using SVD [4,6]. After having found three ellipses, the corresponding contours are resampled with sub-pixel accuracy. A coarse pose is estimated based on the ratio of semi-major axes

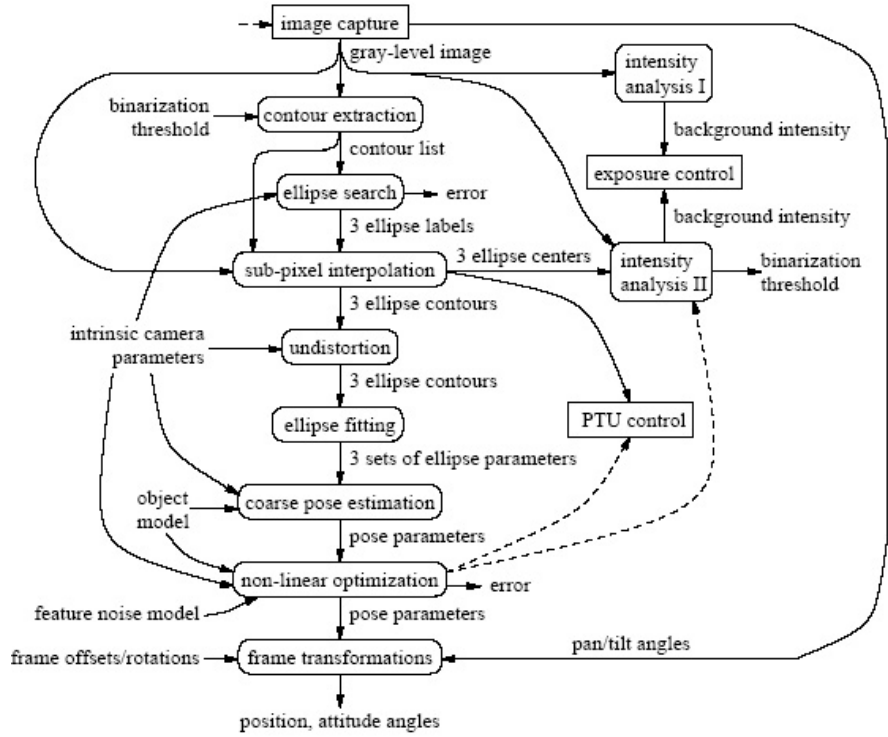


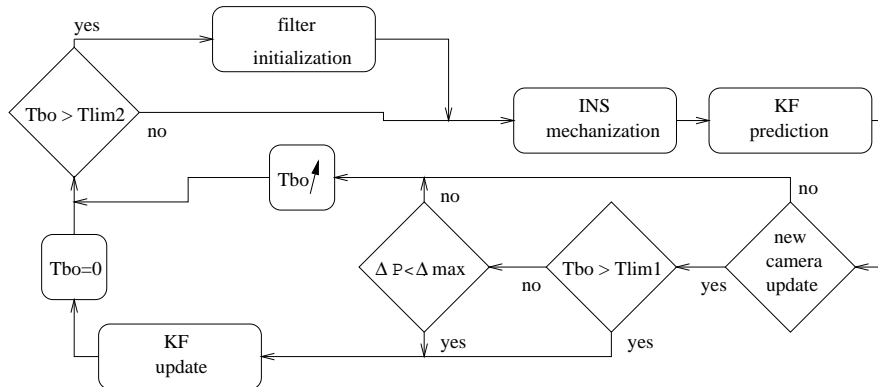
Fig. 3. Dataflow in the vision system.

and circle radii. The estimation is optimized by minimizing the reprojection error:

$$\min_{\mathbf{a}} \sum_{i=1}^{12} \left( \frac{d_i - c_{i \bmod 4}}{\sigma_{i \bmod 4}} \right)^2 \quad \begin{aligned} \mathbf{d} &= (u_{e1}, v_{e1}, l_{a1}, l_{b1}, \dots, u_{e3}, v_{e3}, l_{a3}, l_{b3}) \\ \boldsymbol{\sigma} &= (\sigma_c, \sigma_c, \sigma_l, \sigma_l) \\ \mathbf{c} &= (\hat{u}_e(\mathbf{a}), \hat{v}_e(\mathbf{a}), \hat{l}_a(\mathbf{a}), \hat{l}_b(\mathbf{a})) \end{aligned} \quad (3)$$

This function is non-linear and minimized iteratively using the fast-converging Levenberg-Marquardt method [6]. It's initialized with the pose parameters from the first estimate. The uncertainties of the ellipse centers  $\sigma_c$  and axes  $\sigma_l$  are known from separate noise measurements. Finally, the pose parameters are converted to helicopter position and attitude using angles from the PTU and known frame offsets and rotations. The PTU control runs in parallel to the image processing using pixel coordinates of the pattern center as input, aiming at centering the landing pad in the frame as soon as it's localized.

Two methods for analyzing image intensities are implemented. The first estimates the background intensity of the reference pattern based on the assumption being the brightest surface in the image. When the landing pad is detected, the second method is applied. It computes the background inten-



**Fig. 4.** The filter architecture.

sity and the binarization threshold based on the intensity distribution of the pattern. The exposure controller controls the camera shutter time and iris aiming at keeping the background intensity in a certain range.

### 3 Sensor Fusion

The position and attitude estimates provided by the vision system can not be fed directly into the controller due to their intrinsic lack of robustness: the field of view can be temporarily occluded (for example by the landing gear), the illumination conditions can change dramatically just by moving few meters (sun reflections, shades, etc.). On the other hand, vision readings are very accurate, when available.

Hence, a navigation filter based on a Kalman filter (KF) has been developed, fusing highly accurate 3D position estimates from the vision system with inertial data provided by the on-board accelerometers and angular rate gyros. Besides filtering out a large part of the noise and outliers, the filter provides a satisfying dead reckoning capability, sufficient to complete the landing even when the vision system is "blind"<sup>1</sup>, see Fig. 8.

The implementation of the KF is done using the *error state space* or *indirect* formulation with *feedback mechanization* (Fig. 4). The states of the filter are the estimated inertial navigation system (INS) errors. The three observations are given by the difference between the INS position and the position from the vision system (lateral, longitudinal and vertical position relative to the pattern). The advantage of the indirect formulation versus the direct formulation (position, velocity and attitude are among the state

<sup>1</sup> During the last 50 cm before touch down the vision system is often "blind" due to two factors: (a) the shade of the helicopter covers part of the pattern at touch down, and (b) when the distance of the camera to the pattern is very small it is very hard for the controller of the pan/tilt unit to keep the pattern in the picture.

variables in the filter) lies in the fact that the INS errors have much slower dynamics than the navigation variables and are very well described by linear equations. In this particular application we have to deal with black-out camera periods in the order of seconds. The indirect KF is quite robust in this respect, in fact an effective indirect filter can be developed with a sample periods of the order of half a minute [5]. The estimated errors are fed back into the mechanization algorithm to avoid unbounded growth of the INS errors.

The inertial measuring unit (IMU) used in this application is integrated in the Yamaha Attitude Sensor (YAS) and it is composed of three accelerometers and three rate gyros. The output rate is 200 Hz for the gyros and 66 Hz for the accelerometers. The filter runs at 50 Hz and the inertial sensors are sampled at the same frequency. Both gyro and accelerometer outputs are prefiltered at 25 Hz to take into account the information available between the filter samples.

The filtered IMU outputs are used in the INS mechanization step to calculate the position, velocity and attitude by solving the inertial navigation equations (4) where  $\mathbf{r}^n$  and  $\mathbf{v}^n$  are the position and velocity vectors,  $\mathbf{C}_b^n$  is the direction cosine matrix of the attitude angles.  $\mathbf{f}^b$  and  $\boldsymbol{\Omega}_{ib}^b$  are the accelerometers and gyros outputs,  $\mathbf{g}^n$  is the gravity vector and  $\boldsymbol{\omega}_{ie}^n$  the Earth rotation rate.

$$\begin{aligned}
\dot{\mathbf{r}}^n &= \mathbf{v}^n & \delta \dot{\mathbf{r}} &= -\boldsymbol{\omega}_{en} \times \delta \mathbf{r} + \delta \mathbf{v} \\
\dot{\mathbf{v}}^n &= \mathbf{C}_b^n \mathbf{f}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}^n + \mathbf{g}^n & \delta \dot{\mathbf{v}} &= -(\boldsymbol{\omega}_{ie} + \boldsymbol{\omega}_{in}) \times \delta \mathbf{v} - \boldsymbol{\psi} \times \mathbf{f} + \delta \mathbf{a} \\
\dot{\mathbf{C}}_b^n &= \mathbf{C}_b^n (\boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{in}^b) & \dot{\boldsymbol{\psi}} &= -\boldsymbol{\omega}_{in} \times \boldsymbol{\psi} \\
& & \delta \dot{\mathbf{a}} &= -\beta \delta \mathbf{a}
\end{aligned} \tag{5}$$

Several filter configurations have been tested, the final implementation is a 12-state KF with 9 navigation error states and 3 accelerometer biases. The dynamic model of the KF is based on the error dynamics equations (5) where  $\delta \mathbf{r}$ ,  $\delta \mathbf{v}$  and  $\boldsymbol{\psi}$  are the position, velocity and attitude error vectors and  $\delta \mathbf{a}$  are the accelerometer biases.

In Fig. 4 the filter architecture is shown. The black-out time ( $T_{bo}$ ) is the time elapsed since the last valid update. When a new update from the vision system is available, it is first compared with the predicted position and if the difference ( $\Delta P$ ) is smaller than the maximum allowed tolerance ( $\Delta_{max}$ ) it is passed to the KF as a new observation. This consistency check is active only if  $T_{bo} < T_{lim1}$  because the uncertainty of the system increases each time the update is not made. The KF prediction equations are applied at each time step. When there is a black-out from the vision system the uncertainty of the INS error system represented by its covariance grows unbounded until it's not possible to believe in it anymore.  $T_{lim1}$  is the maximum black-out time after that the new vision update is passed directly to the filter. The covariance update equation reduces the covariance of the error system and the decrease is large when the covariance of the measurement is small. The covariance of the vision system measurement is quite small, this means that


immediately after the update the uncertainty of the INS error system is low and the consistency check can be activated again. Of course, it can happen that the first update after the black-out is an outlier and in this case it can't be detected. For stability reasons the landing is aborted after a maximum black-out time  $T_{lim2}$  and the filter is reinitialized.

## 4 Flight Controls

The requirements set on the flight control system during landing are the following:

1. The landing mode should be engaged from any point where the landing pad is visible, that means approximately within a 20 m radius hemisphere, centered on the pattern.
2. Once the landing mode is engaged, the helicopter state should be compatible with the proper functionality of the vision system, until touchdown, this means that during the approach phase the following should be considered : (a) the helicopter's position and attitude should not be such as to cause physical occlusion of the visual field; this may happen due to the landing gear skids or the mechanical limitations of the pan/tilt unit; (b) the regions where the accuracy of the vision system is worst should be avoided, if possible; (c) the helicopter velocity and angular rates should not saturate the pan/tilt unit capability for compensation: too high angular rates of the visual beam may result in blurred images; (d) the position of the dominant light source (sun) should be considered, to avoid full reflections.
3. The wind direction has to be taken into account: tailwind landings should be avoided.
4. The control system should be dimensioned for wind levels up to 10 m/s.
5. The engine should be shut down autonomously, once touch-down is detected. The detection should be timely, since early detections cause high touch down loads and late detections can cause ground resonance.
6. The vertical velocity at touch down should be of the same order of magnitude as a proper manual landing.

In the following, the landing procedure is described. Fig. 5 shows the sequence of control modes and the triggering conditions. As soon as the navigation filter provides state estimates, the helicopter turns towards the pattern to guarantee occlusion-free view of the pattern and flies to a point located 5 meters on the vertical of the desired touch down point ( $P_{TD}$ ). Once the helicopter is on top of  $P_{TD}$ , the heading is changed for landing, taking into consideration the illumination conditions (sun from the side is optimal) and the wind conditions (optimum with head-wind). The final descent is conducted at a constant sink rate of 20 cm/s. At 8 cm from the ground, the



Mode	Horizontal control	Yaw control	Altitude control	Logical condition for Mode Transition
<b>READY</b>				
<b>AIM</b>	hold	$\psi \rightarrow \psi_{AIM}$	hold	$ \psi - \psi_{AIM}  < 5^\circ$
<b>APPROACH</b>	linearly moving to $P_D$ @ $V_{HOR} < 0.5$ m/s	hold	Descending to PD $h \rightarrow h_{D1}$ (5 m) @ $VZ = -50$ cm/s	$ P - P_{D1}  < 2$ m $ h - h_{D1}  < 0.4$ m $ V_Z  < 10$ cm/s $ V_{HOR}  < 0.3$ m $ \psi - \psi_{AIM}  < 3^\circ$
<b>ALIGN</b>	hold	$\psi \rightarrow \psi_{TOUCHDOWN}$	Hold	$(\psi - \psi_{TOUCHDOWN}) < 3^\circ$
<b>DESCEND</b>	hold	hold	$h \rightarrow h_{D2}$ (1 m) @ $VZ = -20$ cm/s	$ P - P_{D2}  < 0.25$ m $ h - h_{D2}  < 0.1$ m $ V_{HOR}  < 0.3$ m $ V_Z  < 0.1$ m/s vision data VALID
<b>TOUCH DOWN</b>	hold	hold	$VZ = -20$ cm/s	$h < 0.1$ m
<b>SHUT OFF</b>	hold	hold	Throttle back descending ramp	

**Fig. 5.** Mode sequence leading to touch down. Note that all transitions are one-directional: once a mode is exited it can not be re-entered.

throttle command is ramped down, inducing loss of lift and touch down.

The control laws of the helicopter consist of an inner loop (pitch, roll and yaw angle control, and vertical velocity control) and an outer loop (position and velocity control). The inner loop consists of the Yamaha Attitude Control System (YACS), the properties of which have been identified with a dedicated system identification session. The control equations of the outer loop can be summarized as following:

$$\begin{aligned}
 \theta_C &= K_{px} \delta X + K_{pvx} \delta V_X + K_{ivx} \delta V_{Xsum} \\
 \Delta \phi_C &= K_{py} \delta Y + K_{pvy} \delta V_Y + K_{ivy} \delta V_{Ysum} \\
 V_{ZC} &= K_{ivz} \delta V_{Zsum} + K_{pvz} (V_{Ztarget} - V_Z) \\
 V_{Ztarget} &= \text{limit}(0.75 \delta Z, V_{Zmin}, V_{Zmax}) \\
 \omega_C &= \text{limit}(K_{pw} \delta \psi, -26 \text{ deg/s}, 26 \text{ deg/s})
 \end{aligned} \tag{6}$$

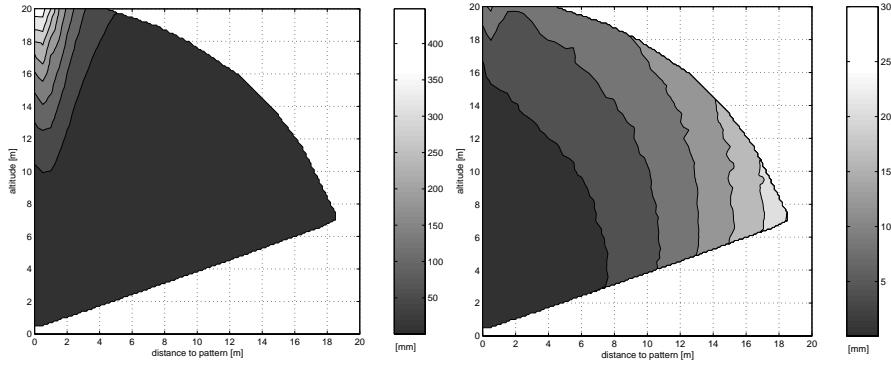
where the subscripted  $K$  are control gains, the  $\delta$  are control errors, the subscript *sum* indicates the integral terms,  $\theta_C$  is the commanded pitch angle,  $\Delta \phi_C$  is the commanded roll angle variation,  $\omega_C$  is the commanded yaw rate and  $V_{ZC}$  is the commanded vertical velocity<sup>2</sup>.

## 5 Experimental Results

The helicopter used for experimentation is a slightly modified Yamaha RMAX (Fig. 1). It has a total length of 3.6 m (incl. main rotor) and a take-off weight

<sup>2</sup> During the descent and touch down phases, the gains of the velocity terms ( $K_{pvx}$  and  $K_{pvy}$ ) are increased by one fifth and the integral terms in the horizontal control are activated, for faster and more precise position control.



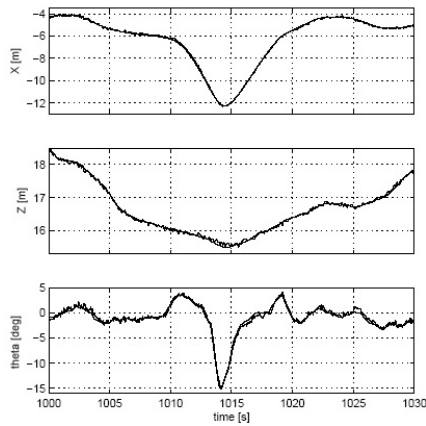


**Fig. 6.** RMS error in horizontal (left) and vertical position (right) from simulation.

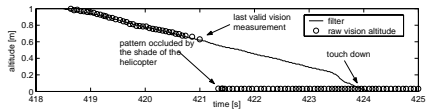
of 95 kg, including 30 kg available for payload. The vision navigation system consists of two PC104 stacks with PIII 700 MHz processors, the inertial sensors of the YAS, and a single standard CCD camera with approx. 45 degrees horizontal angle of view which is mounted on an off-the-shelf pan/tilt unit (PTU). One of the two computers is dedicated to sensor management and low level control of the helicopter, the other one for image processing and control of the camera and the PTU. The two computers communicate over a RS232C serial link. They are built inside a shock and vibration isolated box, which also includes a precision GPS, a barometric altitude sensor, a compass, a video recorder, a video transmitter, and a wireless Ethernet bridge. The PTU is mechanically limited to 111 degrees tilt and  $\pm 180$  degrees pan, the max. angular rate is 300 degrees/s and the resolution 0.051 degrees. It is mounted on a vibration isolating platform on the underside of the helicopter body.

We estimated the RMS error of the vision system in position and attitude depending on the relative position to the pattern in leveled flight. For each position 1000 samples were generated using a feature noise model that included noise from image formation, digitization, and segmentation. We developed a method to analyze noise in ellipse center position and semi-axis length. Errors introduced by the transformation from the camera frame into the body frame were not considered in simulation. Fig. 6 shows the RMS errors ( $1\sigma$ ) in horizontal and vertical position. The error doesn't change much in the simulated envelope (a 20 m radius hemisphere, with a "blind" sector from azimuth 0 to 15 degrees) due to different triplet sizes and sub-pixel feature extraction. For pitch and roll the error behaves similar to the horizontal error, with a maximum RMS value of  $\approx 1$  degree, the error in heading is negligible.

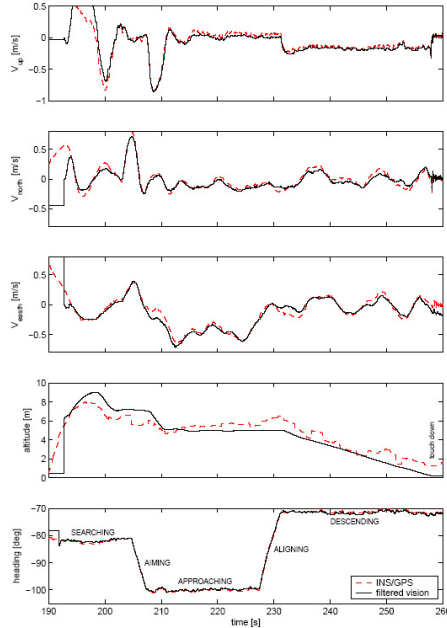
The actual accuracy of the vision system was evaluated through an in-flight comparison with a navigation solution based on the YAS and a precision RTK GPS which supplied horizontal/vertical position with 10 mm/15 mm



**Fig. 7.** Position and heading estimates from the vision system vs. estimates based on YAS and precision RTK GPS observations.



**Fig. 8.** Altitude estimates from the navigation filter when losing vision.



**Fig. 9.** Time histories and control modes during autonomous landing. The dashed lines show readings from a standard INS/DGPS unit.

uncertainty ( $1\sigma$ ). We found good agreement between measured and simulated errors. Fig. 7 shows timeplots for distance, altitude, and pitch angle at typical starting points for autonomous landing. Position and attitude estimates were provided with an average rate of 20 Hz (using  $384 \times 288$  pixels images) and an average latency of 110 ms (including delays from capturing PAL signals).

Some tens of autonomous landings were conducted from different relative positions to the landing pad within the specified envelope, on grass and snow fields, with different wind and illumination conditions. A sample of the results is available in Fig. 10. The vertical velocity at touch down ranged between 18 and 35 cm/s, this corresponds to load factors of about 1.4 g on grass fields. The horizontal velocity at touch down was in the order of magnitude of 15 cm/s. The average touch down point precision was about 42 cm (13 % of rotor diameter). Thanks to the pan/tilt camera and a robust controller, considerable wind levels can be handled. Successful landings have been performed with wind levels on ground up to 30 km/h (2 min average), with gusts of 45 km/h.

Test ID	Wind Speed [km/h]	V <sub>Z</sub> [cm/s]	V <sub>NORTH</sub> [cm/s]	V <sub>EAST</sub> [cm/s]	V <sub>HOR</sub> [cm/s]	P <sub>TD</sub> MEASURED (X <sub>TD</sub> , Y <sub>TD</sub> ) [cm]	P <sub>TD</sub> TARGET (X <sub>TD</sub> , Y <sub>TD</sub> ) [cm]	P <sub>TD</sub> ERROR [cm]	ψ <sub>TD</sub> [°]	ψ <sub>TD-TARGET</sub> [°]
128/2/1	15	<b>-31</b>	-5	-24	<b>25</b>	(51,-17)	(94,-34)	<b>46</b>	161	160
128/3/1	14	<b>-37</b>	-20	17	<b>26</b>	(49,-37)	(94,-34)	<b>45</b>	162	160
128/4/1	13	<b>-35</b>	0	-1	<b>1</b>	(43,-31)	(87,-50)	<b>48</b>	151	150
128/4/2	13	<b>-24</b>	-23	-13	<b>26</b>	(91,-28)	(87,-50)	<b>22</b>	152	150
506/5/1	30	<b>-28</b>	-7	0	<b>7</b>	(41,-33)	(50,-87)	<b>54</b>	122	120
506/6/1	22	<b>-25</b>	-5	-11	<b>12</b>	(47,-53)	(50,-87)	<b>34</b>	123	120
506/6/2	26	<b>-27</b>	1	0	<b>1</b>	(28,-41)	(50,-87)	<b>50</b>	121	120
506/8/1	30	<b>-18</b>	-7	-17	<b>18</b>	(63,-47)	(50,-87)	<b>42</b>	121	120

Fig. 10. Flight test results from several autonomous landings.

## 6 Acknowledgements

We thank Piotr Rudol and Mariusz Wzorek for providing the pan-tilt controller and Anke Knöppler for work on camera calibration. This work was supported by the Wallenberg Foundation, Sweden.

## References

1. P. Doherty. Advanced research with autonomous unmanned aerial vehicles. In *Proc. of the 9th International Conference on the Principles of Knowledge Representation and Reasoning*, pages 731–732, June 2004.
2. P. Doherty, P. Haslum, F. Heintz, T. Merz, T. Persson, and B. Wingman. A distributed architecture for autonomous unmanned aerial vehicle experimentation. In *Proc. of the 7th International Symposium on Distributed Autonomous Robotic Systems*, pages 221–230, June 2004.
3. Jean-Yves Bouguet. Camera Calibration Toolbox for Matlab . [http://www.vision.caltech.edu/bouguetj/calib\\_doc](http://www.vision.caltech.edu/bouguetj/calib_doc).
4. K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1995.
5. P.S. Maybeck. Stochastic models, estimation, and control. In *Mathematics in Science and Engineering*, volume 141-1, pages 289–367. Academic Press, 1979.
6. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.
7. S. Saripalli, J.F. Montgomery, and G. Sukhatme. Visually-guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, 19(3):371–380, June 2003.