periodica polytechnica

Electrical Engineering
53/3-4 (2009) 139–150
doi: 10.3311/pp.ee.2009-3-4.06
web: http://www.pp.bme.hu/ee

RESEARCH ARTICLE

Sensor fusion and actuator system of a quadrotor helicopter

László Kis / *Béla* Lantos

Received 2010-05-25

Abstract

This article focuses on the sensor and actuator system of an autonomous indoor quadrotor helicopter. The sensor system has two parts: an inertial measurement unit (IMU) and a vision system. The fusion between the two systems is solved by extended Kalman filters. The calibration of the inertial measurement unit is described for various types of errors. The variance analysis is performed for the noise sources of sensors. The actuators of the helicopter are the four rotors. The identification and low level control of the brushless DC motor based rotor system is also presented. The embedded control system integrates a lot of processors and communication lines. The verification of the system's parts were performed under real-time conditions. The main results of the paper are the new calibration algorithm for the different sensors and the real-time realization of the complex sensory system that may be part of a formation control system of UAVs.

Keywords

indoor autonomous helicopter · quadrotor · sensory system · state estimation · actuator system

Acknowledgement

This research was supported by the Hungarian National Research Program under grant No. OTKA K 71762.

László Kis

Department of Control Engineering and Information Technology, BME, H-1117 Budapest Magyar Tudósok krt. 2., Hungary
e-mail: lkis@iit.bme.hu

Béla Lantos

Department of Control Engineering and Information Technology, BME, H-1117 Budapest Magyar Tudósok krt. 2., Hungary
e-mail: lantos@iit.bme.hu

1 Introduction

Miniature unmanned ground vehicles have been in the researchers' focus for a long time. Their sizes make them capable to work in laboratory environments. The situation is different in the case of unmanned aerial vehicles - UAVs. For flight UAVs usually need large open space. This paper introduces a miniature quadrotor helicopter, which is designed to fly in indoor environment.

The idea of the quadrotor construction is not new, the first known realization is from 1907 by Jacques and Louis Breguet in France. For decades this type of helicopter has been forgotten because it was more unstable and less maneuverable than the conventional helicopters. But nowadays there are control systems and actuators, which are fast enough to control these UAVs. Moreover one can profit from its mechanical robustness, because the only moving parts are the four rotors.

Our project started as a cooperation between the Computer and Automation Research Institute of the Hungarian Academy of Sciences (MTA SZTAKI) and Department of Control Engineering and Information Technology of the Budapest University of Technology and Economics (BME IIT) in 2006. The goal is to develop an indoor quadrotor helicopter as a testbed for nonlinear control algorithms. The main concept is to design and implement each part of the helicopter separately, then integrate them and discover the potential of this quadrotor system. Because of the indoor environment vision system and 3D inertial measurement unit (IMU) are used for position and orientation sensing. This paper discusses the sensor fusion and actuator system of the quadrotor helicopter that is part of the authors' contribution to this project.

Lots of successful quadrotor researches exist around the world. The aim of these projects is diversified, which shows the inherent potential of this type of construction. Usually most of the projects concentrate on only some aspects of the problems. For example many papers can be found about the dynamic model and the control theory of these vehicles [1], [2], [9]. Others develop their helicopter for different services. The STARMAC platform [5] was built in Stanford University for a multi agent testbed. A combined aerial and ground testbed (RAVEN)

was developed in the Massachusetts Institute of Technology [6]. A semi-autonomous quadrotor helicopter was designed in the Pennsylvania State University for low-cost services [3]. A control system and a pilot augmentation system was developed at the Australian National University for a quadrotor helicopter called X4-flyer [11].

The paper is structured as follows. In Section 2 there is a short description of the helicopter. Section 3 discusses problems and solutions of the sensory system including its connection to the state estimation. The actuator system is described in Section 4. Real-time issues and operation in practical aspect are in Section 5. Test results can be found in Section 6. The paper is closed with conclusion and future plans in Section 7.

2 Description of the helicopter

The concept of functioning can be seen on Fig. 1. There are four rotating propellers in the four corners of a square. The axes of these propellers are perpendicular to the plane of the square. Actuator signals are the four rotational speeds of the propellers (Ω_i). The generated lift force and torques can be calculated as

$$F_{lift} = F_1 + F_2 + F_3 + F_4 = b\sum_{i=1}^{4}\Omega_i^2 \tag{1}$$

$$\tau = \begin{pmatrix} lb(\Omega_4^2 - \Omega_2^2) \\ lb(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{pmatrix} \tag{2}$$

where b and d are propeller constants, l is the distance between the mass centre of the body and the rotors and Ω_i is the angular velocity of one rotor.

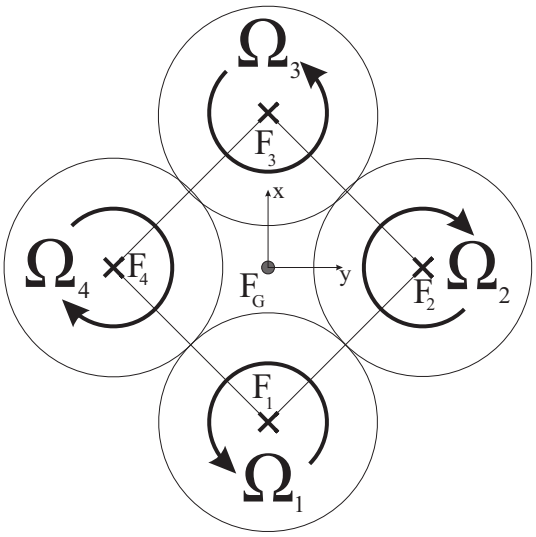

Fig. 1. Concept of the helicopter

The block diagram of the helicopter can be seen on Fig. 2. The main processing unit is a Freescale MPC555 with a Power PC architecture and a float-pointing unit. It runs the control and path planning algorithms and the state estimation. The processor communicates with other units via CAN link. The motor control unit is implemented on an Atmel AT90CAN microcontroller. This unit communicates with the four brushless DC (BLDC) motors by PWM signals. There are four optical tachometers next to the motors, which can produce revolution information for the control.

A Crossbow mNAV100CA inertial measurement unit (IMU) is used for inertial sensing. It has three dimensional (3D) accelerometers and 3D angular velocity sensors. There are other sensors on this unit as well, but the GPS, magnetometers, the pressure sensor and the air-flow meter cannot be used in indoor circumstances. There are also three thermometers built in the angular velocity sensors. The IMU is a commercial one and it was implemented on an Atmel ATMega128 microcontroller, which has no CAN interface. The RS-232 to CAN transmission is solved by an Atmel AT90CAN microcontroller.

The position and orientation sensing is based on a separated vision system using one camera. The algorithms run over a host PC. This PC implements also a user interface, where one can describe the path, send start/stop commands or do logging. The whole module sends the information to the helicopter via radio frequency (RF) channel based on MaxStream XBee Pro units. Each RF transmitter has serial interface. On the board of the helicopter the translation into CAN packages is done by a similar Atmel AT90CAN as in the case of the IMU.

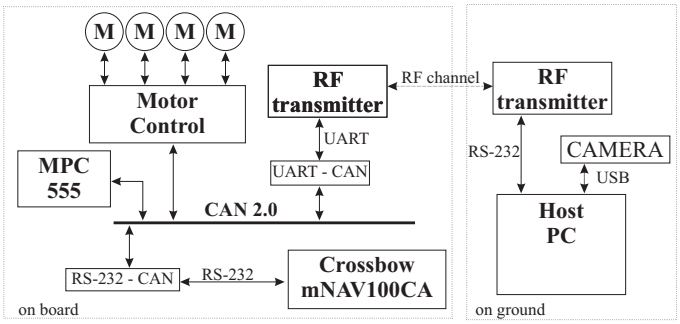

Fig. 2. Block diagram of the helicopter

The control system has three different parts. First is a backstepping controller, which is able to control the non-linear system of the helicopter as described in [7].

The second part is a path planning algorithm that produces the desired path for the control. The main concept is based on a path corner point (checkpoint) system. The path described by its checkpoints and the path planning produces a second order trajectory from one checkpoint to the next. Because of the under-actuated property of the system, the path is described by four parameters: the three position coordinates and the rotation around the vertical axis.

The last part is the state estimation that can produce the actual position, velocity, orientation and angular velocity information for the control using non-linear backstepping algorithm.

3 Sensory system

In the whole system there are two different sources of information about the actual state of the helicopter. First is the IMU. It can produce 3D angular velocity and acceleration information with 10ms sampling time. The other is the vision system. It is a slower one, its sampling time is 100ms. Therefore a module is needed which can perform a sensor fusion feature. With this object two extended Kalman filters are used.

3.1 Coordinate systems

Six frames (coordinate systems) can be defined according to the sensory system. The graph of the frames can be seen on Fig. 3. K_W is the frame of the world. Its origin and orientation is defined in Section 3.4. K_C is the frame of the camera and K_{virt} is a virtual coordinate system, described also in Section 3.4. K_H is the frame of the helicopter, this moves together with the helicopter's body. K_S is the desired frame of the IMU and K_{S_0} is the original frame of the IMU, defined by the manufacturer. Transformations $T_{C,W}$ and $T_{virt,C}$ are described in Section 3.4 and T_{S_0} is described in Section 3.3.1. T_S is originated from the mechanical fixation of the IMU on the body of the helicopter. The fact of the equality of the transformations between K_{virt}, K_H and K_C, K_W is also explained in Section 3.4.

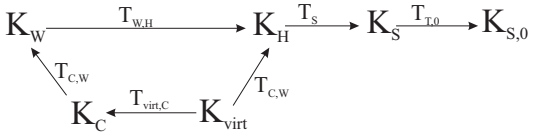

Fig. 3. Graph of the frames

3.2 State estimation

The concept of the separated state estimation is that the IMU cannot be placed in the mass centre of the helicopter. The frame of the helicopter is attached to this mass centre and the control algorithms run in this coordinate system. The frame of the IMU is different from that, therefore an acceleration transformation is needed from the sensor frame to the helicopter frame. It is complicated, because the frame of the helicopter can rotate in K_W. Therefore the estimated angular velocity and angular acceleration of the frame of the helicopter is needed. Moreover to subtract the gravity acceleration vector from the measured acceleration an estimated orientation should be known. Therefore orientation and angular velocity are estimated first by the first extended Kalman filter (EKF1), then acceleration is transformed and finally position and velocity are estimated with the second extended Kalman filter (EKF2).

The exact algorithm of these filters is described in [7]. The mean value of the initial state and its covariance matrix should be known, the system noise (w) and the measurement noise (z) are assumed to be uncorrelated and have zero mean and their covariance matrix should be known. For a single state vector x the algorithm for computing the estimated state ($\hat{x}$) is as follows. The system should be in the form of:

$$x_{k+1} = f(x_k, u_k, w_k) \tag{3}$$

$$y_k = g(x_k, z_k) \tag{4}$$

where u is the input signal of the system and y is the system output. Defining the following matrices,

$$A_{k-1} = \frac{\partial f(x_{k-1}, u_{k-1}, w_{k-1})}{\partial x}|_{x_{k-1}=\hat{x}_{k-1}, w_{k-1}=0}$$

$$B_{w,k-1} = \frac{\partial f(x_{k-1}, u_{k-1}, w_{k-1})}{\partial w}|_{x_{k-1}=\hat{x}_{k-1}, w_{k-1}=0}$$

$$C_k = \frac{\partial g(x_k, z_k)}{\partial x}|_{x_k=\overline{x}_k, z_k=0}$$

$$C_{z,k} = \frac{\partial g(x_k, z_k)}{\partial z}|_{x=\overline{x}_k, z_k=0}$$

the algorithm has two steps:
Prediction:

$$\overline{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0) \tag{5}$$

$$M_k = A_{k-1}\Sigma_{k-1}A_{k-1}^T + B_{w,k-1}R_{w,k-1}B_{w,k-1}^T \tag{6}$$

Time update:

$$S_k = C_k M_k C_k^T + C_{z,k}R_{z,k}C_{z,k}^T \tag{7}$$

$$G_k = M_k C_k^T S_k^{-1} \tag{8}$$

$$\hat{x}_k = \overline{x}_k + G_k(y_k - g(\overline{x}_k, 0)) \tag{9}$$

$$\Sigma_k = M_k - G_k S_k G_k^T \tag{10}$$

According to [7], the separated orientation and position subsystems for the estimation are:
EKF1:

$$\begin{pmatrix} \eta_{k+1} \\ \rho_{S,b_{k+1}} \end{pmatrix} = f_{ang}\left(\begin{pmatrix} \eta_k \\ \rho_{S,b_k} \end{pmatrix}, \rho_{S_k}, w_{ang,k}\right) \tag{11}$$

$$\eta_{m,k} = g_{ang}(\eta_k, z_{ori,k}) \tag{12}$$

EKF2:

$$\begin{pmatrix} v_{k+1} \\ a_{S,b_{k+1}} \\ p_{k+1} \end{pmatrix} = f_{acc}\left(\begin{pmatrix} v_k \\ a_{S,b_k} \\ p_k \end{pmatrix}, a_k, w_{acc,k}\right) \tag{13}$$

$$p_{m,k} = g_{acc}(p_k, z_{pos,k}) \tag{14}$$

where $\eta = (\Phi, \Theta, \Psi)^T$ is the actual orientation expressed by Euler (roll, pitch, yaw) angles. v and p are the actual velocity and position, $\rho_{S,b}$ and $a_{S,b}$ are the bias error of the angular velocity and the acceleration sensor, ρ_S and a are the angular velocity and acceleration from the IMU, η_m and p_m are the measured orientation and position produced by the vision system. The w_{ang} is the state noise of ρ_S and $\rho_{S,b}$, then w_{acc} is the state noise of v, $a_{S,b}$ and a, while z_{ori} is the noise of the orientation measurement and z_{pos} is the measurement noise of the position. All signals are three dimensional ones. To initiate the extended Kalman filters the correlation matrices of the noises will be needed.

Then the algorithms are similar both for the orientation and the position estimation, with the following chosen vectors:

$$\begin{aligned} x_{ang} &= (\eta^T, \rho_{S,b}^T)^T,\ u_{ang} = \rho_S,\ y_{ang} = \eta_m \\ x_{acc} &= (v^T, a_{S,b}^T, p^T)^T,\ u_{acc} = a,\ y_{acc} = p_m \\ R_{w,k} &= E[w_k, w_k^T],\ R_{z,k} = E[z_k, z_k^T] \end{aligned} \tag{15}$$

3.3 Inertial Measurement Unit

There are accelerometers and angular velocity sensors on the board of the helicopter. To get acceptable measures from the IMU few types of errors should be noted. Primary information is listed on the datasheets of the sensors. To estimate the values of these errors, calibrations can be done. There are three different phases when error values can be calculated. First is before assembling the sensor to the board of the helicopter, called off-line calibration. Here there are a lot of opportunities, because time efficiency is not so important. In this case time independent errors can be measured. These errors are the non-perpendicularity of the axis of the sensors, the gain errors and the gravity effect on the angular velocity sensor. The gain error of the acceleration is also temperature dependent and the temperature can change from flight to flight, but this dependency is linear and can be calculated off-line.

The second phase is just before the flight. In this case there are only few seconds for calibration and usually measurements can be done only in one orientation . During this method, errors can be calculated which are supposed to be constant during the whole flight, and measurement values can be used to initiate the third phase. The values to be measured are the temperature and the initial biases.

The third phase is even more an estimation than a calibration. Based on the before start-up measurements, the extended Kalman filters can be initiated to estimate the biases of the acceleration or the angular velocity.

For each calibration step 3D acceleration and angular velocity are measured. These measurements are always mean values of a few second long data acquisition. By this way noise effect is reduced considerably.

3.3.1 Acceleration calibration

Off-line calibration The concept of this calibration is that it should be general and should not need any special equipment. The theory is the following. If there isn't any error in the acceleration in stationary position in different orientations, then the measured gravity vector should be on the surface of a sphere with radius of one (the sensor was made in the USA and measures the acceleration in G units) and with a centre in the origin. Because of the composite error sources, the gain, bias and non-perpendicularity errors, this surface in reality is an ellipsoid with unknown centre. This is still a quadratic surface:

$$[a_i^T\ 1]Q\begin{bmatrix} a_i \\ 1 \end{bmatrix} = 0 \tag{16}$$

where a_i is the measured 3D acceleration vector in the frame of K_{S_0}, and Q is a 4×4 symmetric, positive definite matrix with 10 free parameters. Let it be transformed to the following form:

$$a_{pi}^T p = 0 \tag{17}$$

where $a_{pi}^T = [a_x^2\ a_y^2\ a_z^2\ a_x a_y\ a_x a_z\ a_y a_z\ a_x\ a_y\ a_z\ 1]$ and p is the vector of the 10 free parameters of Q. Let F be the matrix containing the measured a_{pi}^T in its rows, then the $Fp = 0$ homogeneous linear equation should be solved. To avoid the trivial solution the $p^T p = 1$ constraint is used, which reduces the numbers of free parameters to 9. The problem is solved by the total least squares (LS) method. Decompose F with the singular value decomposition (SVD) in the form $F = USV^T$. Then p will be the column of V^T connected to the least singular value.

Now separation of $Q = T^T T$ should be done, where T is the transformation between the real ellipsoid and the ideal sphere. Let Q be in the following form:

$$Q = \begin{bmatrix} M & v \\ v^T & c \end{bmatrix} \tag{18}$$

where M is a symmetric 3×3 matrix and v is a 3×1 vector. Matrix T is assumed in the following form:

$$T = \begin{bmatrix} N & -Np_e \\ 0^T & 1 \end{bmatrix} \tag{19}$$

where N is an orthogonal rotation matrix and p_e is the position of the centre of the ellipsoid, then p_e can be calculated as:

$$p_e = -M^{-1}v \tag{20}$$

and N is the separation of M as $M = N^T N$. This problem can be solved by the eigenvalue, eigenvector separation.

The T matrix transforms the ellipsoid to the sphere but the orientation is unknown. At this point the desired frame of the IMU (K_S) can be defined. The sensor must be rotated to the orientation where the desired Z axis is parallel to the gravity vector and a measurement has to be taken. Then the step should be repeated with the desired X axis. Thus the unity vectors of the axes of K_S are measured in K_{S_0}. Then the measured X vector is projected to the plane which normal is the measured Z vector thus two perpendicular vectors are calculated which describe now the orientation of K_S with its Y and Z axis in the frame of K_{S_0}. From this information one can find the rotation between the desired and original frame. Let matrix E be a 4×4 homogeneous matrix with a 0 position vector and with the orientation transformation between K_{S_0} and K_S.

In the end of the off-line acceleration calibration the calibrated values in K_S are:

$$\begin{bmatrix} a_{cal} \\ 1 \end{bmatrix} = ET\begin{bmatrix} a_{measured} \\ 1 \end{bmatrix} \tag{21}$$

We note that this algorithm calculates also the biases that are valid during the calibration. The calibrated measurements will be used later at the off-line calibration of the angular velocity sensor.

The gain error of the accelerometer is temperature dependent. This is a linear dependency so at least two measurements in different temperature conditions are needed. First, gain values should be separated from T. These values are the length of the columns of the N matrix. So N can be separated as $N = \tilde{N}\ \mathrm{diag}(s_x, s_y, s_z)$, where $\tilde{N}$ is an orthonormal ($\tilde{N}^{-1} = \tilde{N}^T$) matrix and s_x, s_y, s_z are the gain errors.

Now the following equation has to be solved:

$$[t_i\ 1]\begin{bmatrix} m_x & m_y & m_z \\ b_x & b_y & b_z \end{bmatrix} = \begin{bmatrix} s_{xi}\ s_{yi}\ s_{zi} \end{bmatrix} \tag{22}$$

where t_i is a vector with measured temperatures and $[s_{xi}\ s_{yi}\ s_{zi}]$ vectors are the measured gain errors. This problem can be solved by three least squares (LS) algorithms.

Then the actual gain error is

$$[s_x\ s_y\ s_z]^T = t_{actual}[m_x\ m_y\ m_z]^T + [b_x\ b_y\ b_z]^T \tag{23}$$

If $\tilde{T}$ is the T matrix where N is changed to $\tilde{N}$, then the calibrated acceleration in K_S is

$$\begin{bmatrix} a_{cal_S} \\ 1 \end{bmatrix} = E\tilde{T}\ \mathrm{diag}(s_x, s_y, s_z, 1)\begin{bmatrix} a_{measured} \\ 1 \end{bmatrix} \tag{24}$$

and the transformation between K_S and K_{S_0} is

$$T_{S_0} = E \cdot \tilde{T} \tag{25}$$

Calibration before start-up To calculate the bias before start-up, it is assumed that in the starting position the orientation of the helicopter is known. For example the helicopter should start from an orientation where the x-y plane of the sensor frame is horizontal. In this case the measured gravity vector should be $[0\ 0\ -1]$. Acceleration information should be collected and with the measured temperature the calibrated value can be calculated like in (24). Then the bias will be the deviation from the expected value.

Estimation during flight The reason beside the bias estimation can be seen on Fig. 4. It is a 10 minutes long data acquisition and one of the biases changes $0.2\frac{m}{s^2}$.

The acceleration bias estimation is done by the second stage of the extended Kalman filter (EKF2). To initiate the $a_{S,b}$ state values from the start-up calibration are used. The v state of the Kalman filter can be initiated by 0 and the state p is the starting position measured by the vision system.

For the start of the filter the correlation matrix of the noises are needed. It is always difficult to give a starting estimation for the correlation of the state noises, because these are related to the internal properties of the system. As described in [7],

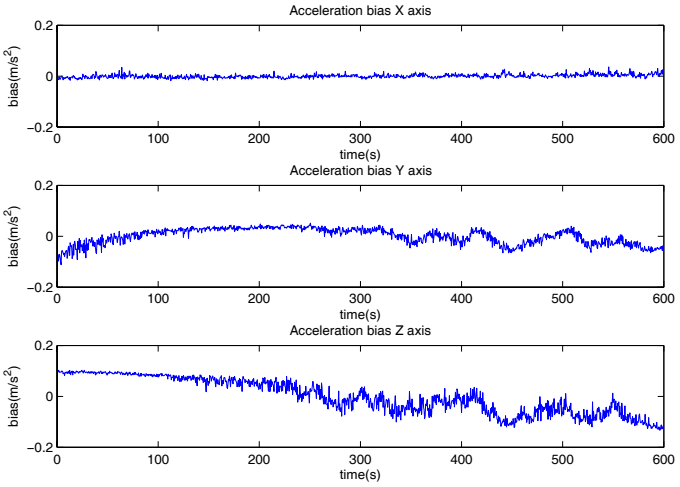

Fig. 4. Variation of the acceleration bias

in the case of position and velocity estimation, the acceleration noise, acceleration bias noise and velocity noise should be investigated. To calculate an initial value, a few seconds long acquisition in stationary position is taken and the mean of the data is subtracted. The first two noises are measured by their sum, so it is difficult to separate them, the third noise cannot be measured. One solution is to separate the measured acceleration noise in frequency, the low frequency part is the bias noise and the high frequency part is the other.

The results of one measurement are the following. The covariance matrices can be approximated with diagonal ones, because the order of the off-diagonal elements are smaller than the diagonal elements. The covariance matrix of the acceleration noise is:

$$R_{w,acc_{noise}0} = \mathrm{diag}(9 \cdot 10^{-4}\ , 3 \cdot 10^{-3}\ , 4.2 \cdot 10^{-3})\left(\frac{m}{s^2}\right)^2$$

The covariance matrix of the acceleration bias noise is:

$$R_{w,acc_{bias}0} = \mathrm{diag}(10^{-4}\ , 10^{-3}\ , 4.7 \cdot 10^{-3})\left(\frac{m}{s^2}\right)^2$$

The third state noise is a velocity noise. It cannot be directly measured, therefore some kind of approximation is given for its variance, based on the acceleration noise. It can be said that the velocity is the integrated acceleration. If the integration is approximated with Δt sampling time in the following form:

$$v_i = \int_{i-1}^{i} a_i \approx \frac{a_{i-1} + a_i}{2}\Delta t \tag{26}$$

one can calculate the variance of the v_i. If the variance of a_i is $\sigma^2(a_i) = \sigma^2(a_{i-1}) = \sigma^2$ constant and it is supposed that the different a_i values are not correlated, than $\sigma^2(v_i) = \frac{\sigma^2}{2}\Delta t^2$. In our case the sampling time of the IMU is 0.01s, therefore the covariance matrix of the velocity noise can be initiated with $R_{w,vel_0} = R_{w,acc_0} \cdot 5 \cdot 10^{-5}$.

It is supposed that there is no correlation between different state noises, so the covariance matrix of the full state noise vector can be approximated with a diagonal one:

$$R_{w,0} = \mathrm{diag}(R_{w,acc_{noise}0}, R_{w,acc_{bias}0}, R_{w,vel}{}_0) \quad (27)$$

This approximation will be significant in Section 5.2.

3.3.2 Angular velocity calibration

Off-line calibration The main concept is the following. Measurements are taken mainly in stationary position, where the expected angular velocity is 0 °/s. First acceleration effect and the biases are reduced, then the gain error will be calculated. The remaining error is the non-perpendicularity of the axes. A similar process as at the accelerometer could be used if a known, constant angular velocity can be produced. For this a special equipment is needed which can produce constant angular velocity. Unfortunately such an equipment was not available therefore this type of error was not considered.

The type of the angular velocity sensor is an Analog Devices ADXRS150. According to the datasheet, the acceleration effect is in a linear connection with the acceleration vector. So in stationary position, where the gain error is negligible, the measured angular velocity in K_{S_0} is the following:

$$\omega_{measured} = \omega_{real} + \omega_{bias} + K^T a_{cal} \quad (28)$$

where K is the acceleration compensation matrix and a_{cal} is the three dimensional calibrated acceleration vector, measured at the same time as the angular velocity. Note that a_{cal} is in the desired sensor frame (K_S) and $\omega_{measured}$ is in the original one (K_{S_0}) of the IMU, so in K matrix there is a rotation part as well. Rewrite (2) to the following form:

$$\omega_{bias} + K^T a_{cal} = R_{ang}^T a_{calh} = [K^T \omega_{bias}] \begin{bmatrix} a_{cal} \\ 1 \end{bmatrix} \quad (29)$$

if ω_{real} is expected to 0° /s, then the following equation should be solved:

$$\omega_{measured}^T = (a_{calh})^T R_{ang} \quad (30)$$

If the measured angular velocities are put into the rows of an Ω matrix and the calibrated acceleration vectors into the A matrix's rows in homogeneous coordinate form, then the problem can be solved with the LS algorithm:

$$R_{ang} = (A^T A)^{-1} A^T \Omega \quad (31)$$

Hence the calibrated angular velocity without gain compensation in K_{S_0} is:

$$\widetilde{\omega} = \omega_{measured} - R_{ang}^T a_{calh} \quad (32)$$

Now $\tilde{\omega}$ is in the original frame (K_{S_0}). In the following step it will be transformed to the desired sensor frame (K_S). Let Γ be the upper 3×3 rotational block of the T_{S_0} matrix in (25). Then Γ describes the rotation from K_{S_0} to K_S.

K_{S_0} and K_S are attached to the rigid body of the sensor and their origin are the same, therefore the compensated angular velocity in the desired frame (K_S) is:

$$\omega_S = \Gamma \left(\omega_{measured} - R_{ang}^T a_{calh} \right) \quad (33)$$

The method of gain calibration is as follows. Starting from a stationary position the sensor should be rotated around the Z axis and put again in a stationary position. Repeat it with the remaining two axes. The rotation can be done manually, approximately around the axes. The separation of the axes can guarantee that the rotation angle will be large enough for an accurate calibration. About 90° is enough. All of the measured acceleration and angular velocity should be stored from the first stationary position until the second one.

The angle of one rotation can be calculated as the angle between the measured, calibrated acceleration vector in the two stationary positions. This information can be calculated also from the numerical integration of the stored angular velocity, for example in the case of Z axis:

$$\psi_{ang} = \left(\sum_i \omega_{zi} \right) \Delta t \quad (34)$$

where Δt is the sampling time of the sensor. If the known rotation angle from the acceleration measurement is ψ_{acc}, then the gain error is:

$$s_\psi = \frac{\psi_{acc}}{\psi_{ang}} \quad (35)$$

This can be calculated for the other two axes as well. Finally the calibrated angular velocity can be calculated as:

$$\omega_{cal_S} = \mathrm{diag}(s_\Phi,\ s_\Theta,\ s_\Psi)\Gamma \left(\omega_{measured} - R_{ang}^T a_{calh} \right) \quad (36)$$

Calibration before start-up According to the sensor's datasheet only the bias error is temperature dependent. This can be compensated before start-up. If the starting position is stationary the expected value is 0°/s, so the measured, calibrated angular velocity will be the bias error. It can be used as the initial value of the $\rho_{S,b}$ state of the extended Kalman filter.

Estimation during flight For the extended Kalman filter (EKF1) the initial state of the $R_{w,ang}$ covariance matrix is needed. This can be measured from a few second long angular velocity data acquisition. The results of these data show that the covariance matrix can be estimated by a diagonal one, as in the case of acceleration. One result of the covariance is the following:

$$R_{w,ang_{noise}0} = \mathrm{diag}(1.3 \cdot 10^{-4},\ 2.88 \cdot 10^{-4}\ 2.8 \cdot 10^{-4})\ (rad/s)^2 \quad (37)$$

$$R_{w,ang_{bias}0} = \mathrm{diag}(2.82 \cdot 10^{-6},\ 8.52 \cdot 10^{-6}\ 6.4 \cdot 10^{-6})\ (rad/s)^2 \quad (38)$$

The separation of noise is similar as in the case of acceleration.

3.3.3 Calibration algorithm

Merging the acceleration and orientation together then the steps of the algorithm are as follows:
Off-line method:

1 measure temperature, acceleration and angular velocity at least in 10 different orientations
2 calculate p with SVD and compose Q
3 separate Q to $T^T T$ (7),(8)
4 repeat steps 1-3 at least for two different temperatures
5 measure and compose E for one temperature (it is same for each one)
6 specify $\widetilde{T}$ for one temperature (it is same for each one)
7 calculate s_x,s_y,s_z for each temperature and specify m_x,m_y,m_z,b_x,b_y,b_z (9)
8 determine a_{cal} (23),(24)
9 produce R_{ang} (31)
10 specify Γ
11 determine s_Φ,s_Θ,s_Ψ (34),(35)

Before start-up method:

1 measure temperature (t)
2 determine $S_{acc} = \mathrm{diag}(t \cdot m_x + b_x, t \cdot m_y + b_y, t \cdot m_z + b_z)$
3 measure the biases (a_{bias},ω_{bias})

Real-time method:

1 $a_{cal} = E \cdot \widetilde{T} \cdot S_{acc} \cdot a_{measured} - a_{bias}$
2 $\omega_{cal} = \Gamma(\omega_{measured} - R^T_{ang} a_{cal_h}) - \omega_{bias}$

3.4 Vision system

The basic concept of the vision system is originated from motion-stereo approach, but only one camera is used. This camera is attached to the ceiling and the working space of the helicopter is in front of the camera. The helicopter has at least 7 active markers. Then the algorithm is the following:

The camera is attached to the ceiling of the room and let its frame be K_C. In the start-up situation the world frame (K_W) is defined by the frame of the helicopter (K_H). In this situation K_W and K_H are equal. An image should be taken with the camera. Let this image be I_1 and it should be stored. Denote $T_{C,W}$ the unknown transformation between K_C and K_W. Once the spatial positions of the markers are known, from I_1 the transformation $T_{C,W}$ can be specified.

During flight the frame of the helicopter will be in a different position. This situation is shown on Fig. 5. Let a virtual frame (K_{virt}) be defined as a frame linked to K_H with the transformation $T_{virt,H} = T_{C,W}$. A virtual camera is represented by the attached frame K_{virt}. A virtual image (I_{virt}) should be taken with this virtual camera. Because of the fact that the transformation between K_{virt}, K_H and between K_C, K_W is defined to be same, in point of the markers I_1 and I_{virt} are same.

Let the live image of the real camera during flight be denoted by I_{live}. Using the 7-point algorithm (described in [4]) the transformation ($T_{virt,C}$) between K_{virt} and K_C can be calculated. The positions of the markers in the first, stationary situation can also be produced in K_C by this algorithm. Thus $T_{C,W}$ and $T_{virt,C}$ can be calculated.

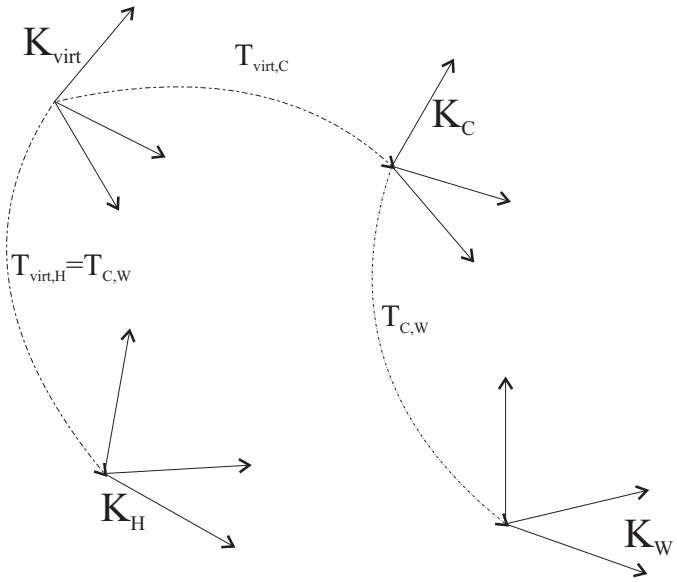

Fig. 5. Graphs of the vision system

Then the position and orientation of the helicopter in K_W is described by

$$T_{W,H} = (T_{C,W})^{-1} \cdot (T_{virt,C})^{-1} \cdot T_{C,W} \tag{39}$$

The operation of the 7 point algorithm is the following. Reference [4] describes the method how to calculate the essential matrix (Λ) which gives the relationship between the pairs of markers on image I_{live} and I_1. Denote $p_{i,1}$ the two dimensional (2D) coordinate of a marker on I_1 and $p_{i,live}$ the 2D coordinate of the same marker on I_{live}. Then the essential matrix $\Lambda_{1,live}$ satisfies

$$p^T_{i,1} \Lambda_{1,live} p_{i,live} = 0 \tag{40}$$

for each marker.

In ideal case

$$\Lambda_{1,live} = R \cdot \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \tag{41}$$

where t is the vector from the origin of K_{virt} to the origin of K_C in K_C and t_x,t_y,t_z are its components. R describes the orientation between the frames.

[4] gives a solution to produce an R matrix and a t vector in

the following form:

$$\Lambda_{1,live} = USV^T \tag{42}$$

$$R = U \begin{bmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T \tag{43}$$

$$t = V \cdot [0\ 0\ 1]^T \tag{44}$$

where (42) is the SVD decomposition of $\Lambda_{1,live}$. The right R matrix can be determined from the fact that the markers should be in front of the camera. Hence $T_{virt,C}$ can be assembled.

Light emitting spheres are used as markers, because it is necessary for image pre-processing that the size of the markers on the images be at least a few pixel large in all situations. Due to the spheres, the colour intensity of the markers will not be uniform on the images.

Because of the non-uniform intensity of the markers, their central points cannot be measured accurately. Therefore the essential matrix differs from the real one. Hence R and t may have errors. The effect of these errors can be seen on a 3D plot on Fig. 6. Let l_i be the projection line from the principal point of the real camera through the central of the i^{th} marker on I_{live}. Let r_i be a similar projection line from the principal point of the virtual camera through the corresponding marker's central on $I_{virt} = I_1$. Then the crosses and dots on Fig. 6 are the nearest points of l_i and r_i for each marker. Finally the line on Fig. 6 is the epipolar axis (also referred as baseline).

According to lots of measurements it can be said that the lines through the corresponding dots and crosses are almost perpendicular to the epipolar axis. Hence this misalignment error would be smaller if R would be corrected with a small rotation around the epipolar axis. Starting from the Rodrigues formula an optimal rotation angle can be calculated. One solution is described in [8].

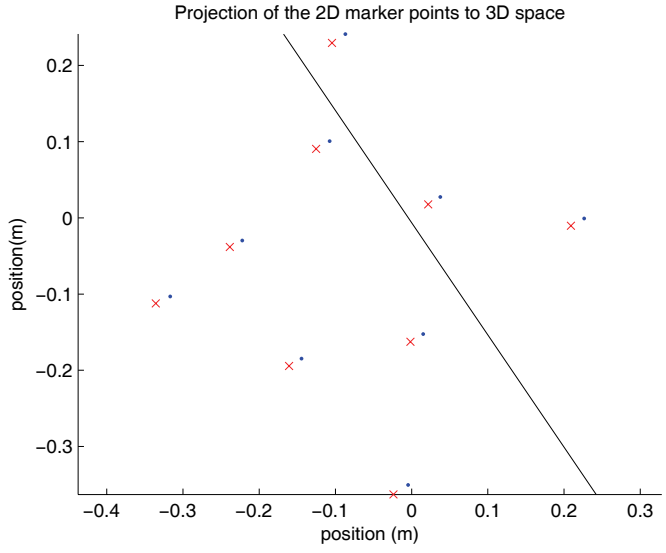

Fig. 6. Position error of the reprojected markers

Connection between the vision and state estimation system The vision system produces the y input of the state estimation, in the case of the position and orientation as well. The extended Kalman filter deals with measurement noises (z). Their effect appear in the covariance matrices, $R_{z,k}$.

As in the case of the IMU some kind of approximations should be given as initial values. Because of the fact that the helicopter starts from a stationary position, a few second long measurements can be taken. The initial value can be the mean value of the position and orientation components of the state vectors. The remaining parts are the noises with 0 expected values and correlation matrices can be calculated.

The source of the noises can be deduced from the concept of the vision system. The preprocessing unit should find the markers on the image and calculate the position of the centre of the markers. Because of the fact that the light intensity values of the markers are not constant on the whole circle, a noise will appear in each centre calculation. Since all of the markers are used for both the position and orientation calculation therefore it cannot be said that the components of the position or the orientation are uncorrelated. Hence the correlation matrix cannot be approximated with a diagonal one. Fortunately, the state estimation concept necessitates only the independence of the state and measure noises.

The resulting correlation matrices are:

$$R_{z,ori_0} = \begin{bmatrix} 0.0004 & 0.0009 & 0.0008 \\ 0.0009 & 0.0019 & 0.0017 \\ 0.0008 & 0.0017 & 0.0015 \end{bmatrix} m^2 \tag{45}$$

$$R_{z,pos_0} = \begin{bmatrix} 1.38 \cdot 10^{-4} & -8.34 \cdot 10^{-5} & -1.2 \cdot 10^{-4} \\ -8.34 \cdot 10^{-5} & 5.39 \cdot 10^{-5} & 7.84 \cdot 10^{-5} \\ -1.2 \cdot 10^{-4} & 7.84 \cdot 10^{-5} & 1.19 \cdot 10^{-4} \end{bmatrix} (rad)^2 \tag{46}$$

Finalizing the state estimation The remaining matrix, which should be initiated, is the covariance Σ_k. For $k = 0$:

$$\Sigma_0 = E\{(x(0) - E\{x(0)\})(x(0) - E\{x(0)\})^T\} \tag{47}$$

These Σ matrices can be formed from the existing information:

$$\Sigma_{ang_0} = \mathrm{diag}(R_{z,ori_0}, R_{w,ang_{bias}0}) \tag{48}$$

$$\Sigma_{acc_0} = \mathrm{diag}(R_{w,vel_0}, R_{w,acc_{bias}0}, R_{z,pos_0}) \tag{49}$$

Finally the time aspect of the sensor fusion problem should be solved. There are two different sampling times ($0.01s$ for IMU and $0.1s$ for vision). The problem can be solved in the way that if there is new information from the vision system then state update is done as in (**??**). Otherwise $\hat{x}_k = \overline{x}_k$ should simply be done.

4 Actuator system

The actuators of the indoor helicopter are four propellers. The system which rotates these propellers has five separate parts. The first is the propeller, the second is the three-phase brushless DC motor. The third part is a commercial driver which can produce the three-phase signal for the motor, based on a pulse width modulated (PWM) input signal. This unit is responsible for timing and commutation. The fourth part of the system is the motor control unit. This is implemented in an Atmel AT90CAN and it does the basic control of the revolution. Finally the fifth part is an optical detector, which can measure the actual revolution of the motors.

A propeller with 0.254m (10 inch) in diameter and with 0.12m/rev (4.7 inch/rev) in pitch is used. According to measurements one propeller can produce $2.5N$ lift force with 380 rad/s angular velocity. This speed will be the usual operating point of the motors because the weight of the helicopter will be approximately 1kg.

In (6) it was assumed that the lift force of a propeller is in linear connection with the square of the revolution. According to the measurements the approximation is adequate in the angular velocity interval from 300 rad/s to 450 rad/s, if $b = 1.76 \cdot 10^{-5}$ constant is chosen.

4.1 Motor identification

The goal is controlling the revolution of the propeller. In this purpose an actuator signal should be produced for the motor driver. If the revolution control is successful it can be a good approximation of the rotor dynamic for the higher level dynamic model of the helicopter. First step of the controller design is the identification of the driver-motor-propeller system.

The main goal is to tell something about the average time constants of the system and try to approximate the scale of nonlinearity. The Identification Toolbox of the MATLAB is used. Measurements were taken around the operating point.

According to [10] the following model can describe the behaviour of the driver-motor-propeller system:

$$\frac{d}{dt}\begin{bmatrix} i \\ \Omega \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_b}{L} \\ \frac{K_m}{\Theta} & -\frac{K_f}{\Theta} \end{bmatrix}\begin{bmatrix} i \\ \Omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{1}{\Theta} \end{bmatrix} e_\tau \qquad (50)$$

where L and R are the inductivity and the resistance of the coils, Θ is the inertia of the motor and the propeller, K_b, K_m are electrical constants of the motor and K_f is the factor of the viscous friction. States i and Ω are the current and the angular velocity, u is the width of the PWM signal on the input of the driver and e_τ is the load disturbance.

The system was identificated as a second order system by a subspace-based state space identification method. The result can be seen on Fig. 7. The time constants of the system are $0.163s$ and $0.017s$. The second time constant is ten times faster than the first one, therefore it can be said that the main time constant of the actuator system is about $0.17s$. Comparing to the other time constants of the quadrotor helicopter, it can be noted that the dynamics of the rotor cannot be negligible.

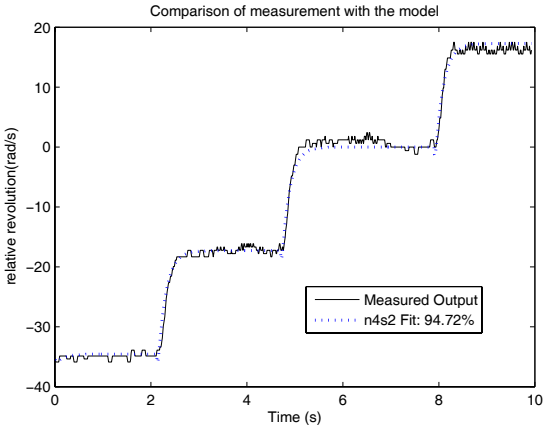

Fig. 7. Result of the identification

Examining the results in Fig. 7, it can be said that the estimated model do not suit very well to the measured data. The situation does not change so much by increasing the order of the estimated system. This is because of the non-linearity. The static characteristics of the propeller shows that this part of the system is the main reason of the non-linearity. Moreover the propeller is a light, plastic one, therefore it can deform.

Different solutions are used to control the actuator system, including PI control, two degrees of freedom control and control with state feedback and load estimation. The goal was not to decrease the time constant, but to stabilise the revolution. Results are in Section 6.3.

5 Real-time operation

The software of the helicopter should run real-time. In the actual state of the project the base sample time is generated by the actuator system. Its PWM signal has 20ms period, therefore the goal is to run the control, state estimation and measurements with this sampling time. The vision system cannot work so fast, but this problem is solved in Section 3.4.

To reach a globally real-time behaviour it is necessary to have a real-time communication. According to Fig. 2, the main communication runs on CAN bus. Beside this there is also a serial point-to-point communication. Both types can be the base of a hard real-time system. The system contains also an RF channel, where the propagation time is varying. Therefore this way of communication causes a soft real-time property. Fortunately, only the vision system communicates with the control on this channel.

5.1 Real-time communication

There are six critical signals in the whole control system, which should propagate from one part of the helicopter to an other. These are the actuator signals, the measured angular velocities of the rotors, 3D measured acceleration and angular ve-

locity from the IMU and 3D measured position and orientation from the vision system.

The CAN protocol is non-interruptible and has priority management. Therefore it can be ensured, that the critical signals propagate with a known worst case propagation time (WCPT). The only situation when a lower priority package can hold on a higher priority one is when the lower priority package has already been sent and actually under transmission. The length of the cables on the board of the helicopter are short, therefore the physical propagation time of a bit is negligible. Hence only the time of the holding by a higher/lower priority package problem is calculated. The communication speed in our case is 500 kbit/s and the package size is 108 bit, hence one package propagation time is 216 μs. The period of critical signals is e at least 20 ms. Priorities and worst-case propagation times of the signals can be seen on Table 1.

Tab. 1. Worst-case propagation times

Priority	Signal	WCPT
1	actuator signal	$432\mu s$
2	measured revolution	$648\mu s$
3	measured angular velocity (IMU)	$864\mu s$
4	measured acceleration (IMU)	$1080\mu s$
5	measured orientation (vision)	delayed
6	measured position (vision)	delayed

There are units on the board of the helicopter which have no CAN interfaces. In this case Atmel AT90CAN microcontrollers are used to transform CAN messages to serial messages. Their operation frequency is 16Mhz, therefore the transformation time is less than 50 μs.

The serial communication is a full duplex point-to-point communication, therefore the propagation time is only the propagation of one message. Package size is 104 bit and transmission speed is 38400bit/s. Hence the propagation time is approximately 3ms.

As the result of these calculations, it can be noted that the propagation times of critical signals are much less than the base sample time of the whole system. Therefore the delay of the communication does not influence notably the behaviour of the control.

5.2 Performance

During the main operation most of the calculation capacity is reserved by the extended Kalman filters. These programs have a non-branch structure, so one can calculate the necessary number of multiplications and additions.

First these numbers were calculated for the general case. The results are 4577 multiplications and 4135 additions. The most of the calculations should be done in (**??**), where in the case of orientation estimation the sizes of the matrices are 6 × 6 and in the case of position estimation 9 × 9 and there are 4 matrix multiplications in (**??**). Which means respectively altogether 864 and 2916 multiplications and 720 and 2592 additions for the two level EKFs.

In Section 3, it was shown that the covariance matrices can be approximated as diagonal ones in the case of the IMU. These matrices are the R_w-s in (**??**). By using this approximation the number of steps for two 9 × 9 matrices' multiplication can be reduced to 81 multiplications and 0 additions. It can also produce reduction in the case of 6 × 6 matrices. Finally the number of steps for the full algorithm can be reduced to 3749 multiplications and 3307 additions, which means 19% of reduction.

More reduction can be done if one investigates the forms of the A_k and B_k matrices. They mainly depend on the model of the helicopter, but some parts of A_k or B_k are unity or zero matrices .

6 Test results

6.1 Calibration results

Similarly to calibration, tests were performed without using any special equipment. Measurements were taken in stationary position and in different orientations.

Before applying the developed calibration algorithm the gravity vectors were measured between $0.87g$ and $1.12g$. After the acceleration calibration the gravity vector reduced to the $0.98g$ to $1.01g$ interval.

Without calibration the measured angular velocity were between $\pm 2.5^{\circ}$/s because of the effect of the gravity and the bias. Performing the calibration this error reduced to $\pm 0.5^{\circ}$/s. This signal can be considered as a zero mean noise, because the mean values of a 1 second long capture were always between $\pm 0.1^{\circ}$/s during the tests.

6.2 State estimation results

The state estimation part of the helicopter was tested separately from the control. The vision system produces information about the actual position and orientation of the helicopter. This subsystem can also be tested, so the vision system produces accurate information.

The reference position and orientation measurements have 0.1s sampling period (10 frame/s). The state estimation is 5 times faster. The base of the validation is to check the prediction property namely whether the tendency of the output of the state estimation after a measured reference position (or orientation) keeps to the future (yet unmeasured) reference position (or orientation) or not. Some results of the tests can be seen in Fig. 8. They show that after a fast change in the position the state estimation will follow this change only with a short delay. It can also be observed that the speed of change in the output is not so fast as the reference signal. This feature is originated from the noise reduction property of the extended Kalman filter.

In connection with the state estimation, the situation when the camera does not detect the helicopter was also investigated. If the refresh equation (**??**) of the Kalman filter will not be performed for a long time because missed information from the

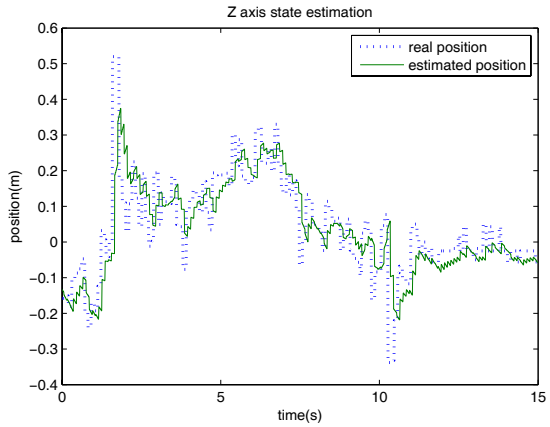

Fig. 8. State estimation test

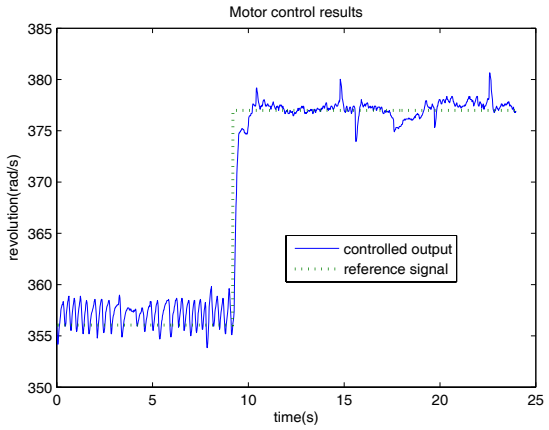

Fig. 10. Controlled actuator signal with a PI controller

vision system, then the state will not be refreshed. One result of the situation is drawn in Fig. 9 and illustrates long time drift in position.

This capture was recorded in a stationary position, in the origin of the world frame, so the estimated value would be 0 m. It can be seen that after a few seconds the errors of the X and Y axes reach the 0.5 m error and increases further. This phenomenon appears because there is no absolute information. The state estimation should numerically integrate the acceleration two times. During this integration even if the acceleration has a small bias error, it accumulates very fast and increases the error. Without a refresh from an absolute data, the estimation cannot reduce this accumulated error.

The conclusion of this test is very important for system behaviour. The control system should use a timer to measure the time intervals when the state of the helicopter cannot be refreshed because of missed vision information. If this timer reaches a few seconds an emergency routine should be started in the control algorithm to avoid the crash of the vehicle.

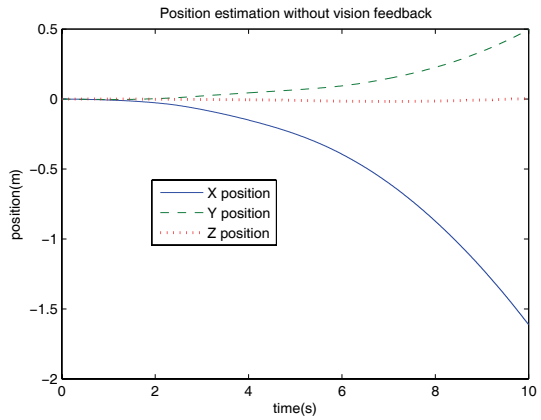

Fig. 9. State estimation without camera

6.3 Motor control results

In Fig. 10 the output signals of a controlled actuator system can be seen. Three types of motor controls were tried, but each control algorithm produced very similar results. In Fig. 10 the controller is a simple PI one. Around the operating point the controlled output follows the desired output, but there are narrow spikes. These can be originated from the deformability of the propeller and the unknown system behaviour of the commercial motor driver. The other typical characteristic is that some oscillation appears by moving from the operating point. This is because of the non-linearity of the system.

It can also be noted that according to the test in [7], beside these low level errors the upper level control algorithms (backstepping etc.) can assure proper system behaviour.

7 Conclusion

In this paper the development, analysis and realization of the sensor and actuator systems of an indoor quadrotor helicopter were presented. The sensor system has two separate parts, an inertial measurement unit and a vision system.

Two stage extended Kalman filters were used as sensor fusion algorithms to merge the two different sources of information. A calibration method was developed for the IMU and based on the types of errors the performance of the vision system was increased. The actuator system has four BLDC motors with commercial motor drivers. The identification method and several control solutions (including PI control, two degrees of freedom control and control with state feedback and load estimation) were presented. Finally the operations of these systems were shown in real-time environment.

High level control algorithms will be presented in another paper. Developments are in progress to increase the speed of the vision system based on the use of graphical signal processors (GPUs) of the PC. The final goal is to develop and experimentally realize the formation control of a set of indoor UAVs.

References

1 **Bouabdallah S, Siegwart R**, *Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor*, IEEE International Conference on Robotics and Automation (Barcelona, Spain), 2005.

2 **Das A, Subbarao K, Lewis F**, *Dynamic Inversion of Quadrotor with Zero-Dynamics Stabilization* (2008).

3 **Hanford S D, Long L N, Horn J F**, *A Small Semi-Autonomous Rotary-Wing Unmanned Air Vehicle (UAV)*, American Institute of Aeronautics and Astronautics, Infotech@Aerospace Conference, 2005.Paper No. 2005-7077

4 **Hartley R I**, *An Investigation of the Essential Matrix*, GE CRD, Schenectady, NY 12301, USA, Tech. Rep. (1993).

5 **Hoffmann G, Rajnarayan D G, Waslander S L, Dostal D, Jang J S, Tomlin C J**, *The Stanford Testbed of Autonomous Rotorcraft for Multi Agent Control (Starmac)*, Digital Avionics Systems Conference, 2004.

6 **How Jonathan P, Bethke B, Frank A, Dale D, Vian J**, *Real-Time Indoor Autonomous Vehicle Test Environment*, IEEE Control System Magazine,2008 April (2008), 51–64.

7 **Kis L, Regula G, Lantos B**, *Design and Hardware-in-the-Loop Test of the Embedded Control System of an Indoor Quadrotor Helicopter*, 6^{th} Workshop on Intelligent Embedded Systems (Regensburg, Germany), 2008.

8 **Kis L, Prohaszka Z, Regula G**, *Calibration and Testing Issues of the Vision, Inertial Measurement and Control System of an Autonomous Indoor Quadrotor Helicopter*, 17^{th} International Workshop on Robotics in Alpe-Adria-Danube Region (Ancona,Italy), 2008.

9 **Madani T, Benallegue A**, *Backstepping Control for a Quadrotor Helicopter*, International Conference on Intelligent Robots and Systems (Beijing, China), 2006.

10 **Molnar Cs**, *Sensorless Steering and Revolution Control of BLDC Motor Driving a Quadrotor Helicopter*, MSc Thesis (Budapest University of Technology and Economics, Department of Control Engineering and Information Technology), 2007.

11 **Pounds P, Mahony R, Hynes P, Roberts J**, *Design of a Four-Rotor Aerial Robot*, Australasian Conference on Robotics and Automation, 2002.

12 **Soumelidis A, Gáspár P, Bauer P, Lantos B, Prohászka Z**, *Design of an Embedded Microcomputer Based Mini Quadrotor UAV*, European Control Conference (Kos, Greece), 2007.