Preprints of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, July 3-6, 2012

Time-delay Extended State Estimation and Control of a Quadrotor Helicopter

Laszlo Kis and Bela Lantos

***Abstract*— The sensory system of an indoor quadrotor helicopter usually includes several types of sensors. This paper focuses on the case, where the helicopter has onboard inertial measurement unit (IMU) and a vision system on the ground. The vision system uses low-cost webcameras, therefore the image acquisition time implies a delay in the measurement, which should be handled during the state estimation. The dynamic model of the helicopter is also introduced. Based on this model, a non-linear control algorithm is used on the helicopter. Because of the unmodelled components of the system, additional trimmer parameters are added to the control. The results of real maneuvers show the effectiveness of the solution.**

I. INTRODUCTION

The quadrotor control design has been in the focus of researchers in recent years. Many studies have been examined the helicopters with simulation [1],[2] and in real-time as well [3],[4]. The leading researches concentrate on the high level controls, like formation control. A common property of the realizations is the fast, precise and expensive vision system for position and orientation estimation. This paper presents the control theory aspect of an indoor quadrotor helicopter design. Our realization uses a low-cost sensor system, therefore the paper focuses on the upcoming issues, like time delay, motor dynamics and unmodelled behavior.

In the next section the dynamic model is introduced. Then the state estimation is presented in section III. The time delay of the vision system is handled in section IV. The control design is shown in section V. Finally the results of real-time tests are presented in section VI.

II. DYNAMIC MODEL

The scheme of the helicopter can be seen in Fig. 1. $\Omega_{r,i}$ refers to the actual revolution of the rotors. The rotor generated lift forces are noticed with F_i. The helicopter dynamics can be derived from this model. The connection between the lift force and the rotor revolution is

$$F_i = b\Omega_{r,i}^2, \tag{1}$$

where b is a propeller constant. Each motor generates an axis parallel torque, which is

$$\tau_{z,i} = d\Omega_{r,i}^2 \tag{2}$$

where d is a constant referring to the torque of the propeller. If the frame of the helicopter is defined as in Fig. 1, the torques around the axes are

$$\tau_x = lb\left(\Omega_{r,4}^2 - \Omega_{r,2}^2\right) \tag{3}$$
$$\tau_y = lb\left(\Omega_{r,3}^2 - \Omega_{r,1}^2\right) \tag{4}$$
$$\tau_z = d\left(\Omega_{r,1}^2 - \Omega_{r,2}^2 + \Omega_{r,3}^2 - \Omega_{r,4}^2\right), \tag{5}$$

where l is the distance between the motors and the mass centre point.

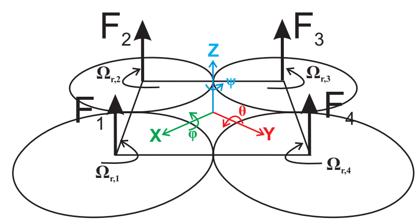

Fig. 1. Forces of the helicopter

A. Movement dynamic

Based on the Newton-Euler equations including the gyroscopic and the rotor inertia effect, the helicopter dynamics is

$$a = \frac{1}{m}\sum F_i - F_g, \tag{6}$$
$$\theta\dot{\omega} = \tau - \omega\times(\theta\omega) - \omega\times\pi_r, \tag{7}$$

where a is the acceleration, F_g is the gravitational force, ω is the angular velocity, θ is the matrix of inertia, $\tau = (\tau_x\ \tau_y\ \tau_z)^T$ and $\pi_r = (0\,0\,\theta_r\Omega_r)^T$. θ_r is the inertia of the rotor and $\Omega_r = \Omega_{r,1} - \Omega_{r,2} + \Omega_{r,3} - \Omega_{r,4}$. Every variable is described in the helicopter frame.

B. Motor dynamic

The second part of the helicopter dynamics is the motor model. The structure of the system can be seen on Fig. 2. The rotation speed is controlled by a low-level controller, which has its own dynamics. After a closed-loop identification, the motor dynamics is a first order system:

$$\dot{\Omega}_{r,i} = \frac{1}{T_m}\left(\Omega_{d,i} - \Omega_{r,i}\right) \tag{8}$$

where $\Omega_{d,i}$ is the desired speed of rotation calculated by the main-level control and T_m is a motor time constant.

As the connection between $\Omega_{r,i}$ and F_i or τ_i in (1) and (2) is pure quadratic, the main-level control should deal with the dynamics of $\Omega_{r,i}^2$. This is in the non-linear form of

$$\frac{d}{dt}\left(\Omega_{r,i}^2\right) = \frac{2}{T_m}\left(\Omega_{r,i}\Omega_{d,i} - \Omega_{r,i}^2\right). \tag{9}$$

L. Kis and B. Lantos is with Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, H-1117 Budapest, Hungary {lkis,lantos}@iit.bme.hu

978-1-4673-2529-5/12/$31.00 ©2012 IEEE

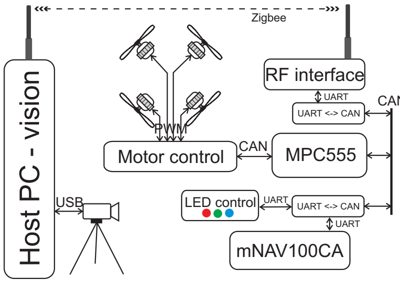

Fig. 2. Structure of the whole system

This formula can be linearized around the operating point $\Omega_{r,i,0} \approx \Omega_{d,i,0}$ resulting in the linearized dynamics

$$\frac{d}{dt}\left(\Omega_{r,i}^2\right) = \frac{1}{T_m}\left(\Omega_{d,i}^2 - \Omega_{r,i}^2\right) \tag{10}$$

C. Full dynamic

The first approximation has already done with the linearization in the motor model. An other one is also useful. The θ inertial matrix can be approximated with a diagonal one, where the diagonal elements are $\theta_x, \theta_y, \theta_z$. This approach is comes from the symmetry of the helicopter body and benefits in the complexity of the control. Finally, the full dynamic model described in the helicopter frame is

$$\frac{d}{dt}\left(\Omega_{r,i}^2\right) = \frac{1}{T_m}\left(\Omega_{d,i}^2 - \Omega_{r,i}^2\right) \quad i = 1...4 \tag{11}$$

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \frac{b}{m}\left(\Omega_{r,1}^2 + \Omega_{r,2}^2 + \Omega_{r,3}^2 + \Omega_{r,4}^2\right)\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} - F_g \tag{12}$$

$$\begin{pmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{pmatrix} = \begin{pmatrix} \frac{lb}{\theta_x}\left(\Omega_{r,4}^2 - \Omega_{r,2}^2\right) + \frac{\theta_y - \theta_z}{\theta_x}\omega_y\omega_z - \frac{\theta_r}{\theta_x}\omega_y\Omega_r \\ \frac{lb}{\theta_y}\left(\Omega_{r,3}^2 - \Omega_{r,1}^2\right) + \frac{\theta_z - \theta_x}{\theta_y}\omega_x\omega_z + \frac{\theta_r}{\theta_y}\omega_x\Omega_r \\ \frac{d}{\theta_z}\left(\Omega_{r,1}^2 - \Omega_{r,2}^2 + \Omega_{r,3}^2 - \Omega_{r,4}^2\right) + \frac{\theta_x - \theta_y}{\theta_z}\omega_x\omega_y \end{pmatrix} \tag{13}$$

III. STATE ESTIMATION

Until this point all the variables were described in the helicopter frame. The only variable in (11)-(13) which is not known or cannot be measured directly is F_g. This value is known relative to the Earth. Therefore the actual state of the helicopter frame relative to the Earth should be known. This attitude is the first goal of the state estimation. The second one is to estimate the actual position and velocity which are needed for the position control.

Three frames will be used in this section. Their geometry can be seen in Fig. 3. K_W is the World coordinate system fixed to the Earth. In this frame the gravity vector is $(0\,0\,g)^T$, where g is the gravitational acceleration. K_H is the helicopter frame, same as in Fig. 1. The position of the origin of K_H in K_W is described by p and the orientation is represented by the quaternion denoted with q. These two values are measured by the vision system.

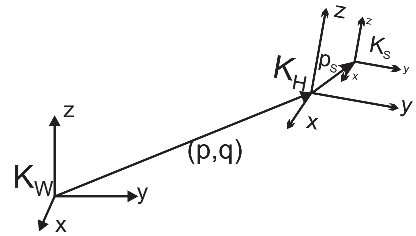

Fig. 3. Frames of the state estimation

K_S is the sensor coordinate system. The onboard sensors are a 3D accelerometer and a 3D angular velocity sensor. Their measurements are known in K_S. It is assumed, that the axes of K_S and K_H are parallel, but their origin is not the same. The position vector between the two origin is p_S known in K_H.

For state estimation two level filters are used.

A. Orientation estimation

First part of the state estimation is the orientation estimation based on the measurements of the angular velocity sensor and the data provided by the vision system. In this section it is assumed that the measurements are available immediately. This assumption is not correct, but it will be handled in the next section by introducing time delay.

The orientation between K_W and K_H is described with quaternion. The quaternion vector $q = (q_1\, q_2\, q_3\, q_4)^T$ is defined as in [5],

$$q = \begin{pmatrix} t\, sin(\alpha/2) \\ cos(\alpha/2) \end{pmatrix} \tag{14}$$

where t is the unit vector of the rotation axis, α is the angle of rotation and the quaternion derivate by the time[5] is

$$\dot{q} = \underbrace{\frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}}_{\Gamma}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{15}$$

The following error model for the measurement of the angular velocity sensor is introduced:

$$\omega_r = \omega_m + \omega_b + \omega_n, \tag{16}$$

where ω_r is the real value of the angular velocity, ω_m is the measured angular velocity, which is the output of the sensor, ω_b is an additional bias, ω_n is a zero mean white noise.

The state estimation is based on the kinematic model of the moving body. Its orientation part is:

$$\dot{q} = \Gamma(\omega_m + \omega_b + \omega_n), \tag{17}$$

where ω_b is unknown and hard to measure, therefore it should also be estimated based on the dynamics:

$$\dot{\omega}_b = \omega_{b,n}, \tag{18}$$

where $\omega_{b,n}$ is a zero mean white noise.

B. Estimator realization

The state estimation is performed on the MPC555 micro-controller in discrete time, hence (17) and (18), should be discretized with T_S sampling time:

$$x_k = (q_k^T \ \omega_{b,k}^T)^T \tag{19}$$

$$u_k = \omega_m \tag{20}$$

$$v_k = \begin{bmatrix} \Gamma_k \omega_n \\ \omega_{b,n} \end{bmatrix} \tag{21}$$

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} I & T_S\Gamma_k \\ 0 & I \end{bmatrix} x_k + T_S \begin{bmatrix} \Gamma_k \\ 0 \end{bmatrix} u_k + T_S v_k = \\ &= f(x_k, u_k, v_k), \end{aligned} \tag{22}$$

where I is the identity matrix, x_k is the state vector, u_k is the input of the kinematic model, v_k is a zero mean white noise and $f(x_k, u_k, v_k)$ is a non-linear function of the state variables.

For the state estimation the output of the measured kinematic model is also needed. In this case this measurement is the orientation from the vision system, denoted with q_m. The image processing system measures slower than the IMU, despite the state estimation runs as fast as the IMU, but the correction based on the vision system is slower. This measurement can be connected to previous description with the formula:

$$y_k = q_m + z_k \tag{23}$$

$$y_k = [I \ \ 0] x_k + z_k, \tag{24}$$

where y_k is the measured kinematic model output and z_k is the zero mean white measurement noise of the vision system.

As the function $f(x_k, u_k, v_k)$ is non-linear the state estimation is performed with extended Kalman filter (EKF). Its algorithm is the modified form of the one in [6]

$$A_{k-1} = \frac{\partial f(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial x} \quad B_{k-1} = \frac{\partial f(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial v} \tag{25}$$

$$C_k = [I \ \ 0] \tag{26}$$

$$\bar{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0) \tag{27}$$

$$M_k = A_{k-1}\Sigma_{k-1}A_{k-1}^T + B_{k-1}R_v B^T{}_{k-1} \tag{28}$$

$$G_k = M_k C_k^T (C_k M_k C_k^T + R_z)^{-1} \tag{29}$$

$$\Sigma_k = M_k - G_k C_k M_k \tag{30}$$

$$\hat{x}_k = \bar{x}_k + \delta_k G_k (y_k - C_k \bar{x}_k), \tag{31}$$

where R_v and R_z are the covariance matrices of $T_S v_k$ and z_k respectively and δ_k is one if the vision system produced new measurement at time k and zero otherwise.

C. Position estimation

According to [5] the rotation matrix between K_H and K_W is

$$R_{H,W} = \begin{bmatrix} \hat{q}_1^2 + \hat{q}_4^2 - \hat{q}_2^2 - \hat{q}_3^2 & 2(\hat{q}_1\hat{q}_2 - \hat{q}_3\hat{q}_4) & 2(\hat{q}_1\hat{q}_3 + \hat{q}_2\hat{q}_4) \\ 2(\hat{q}_1\hat{q}_2 + \hat{q}_3\hat{q}_4) & \hat{q}_2^2 + \hat{q}_4^2 - \hat{q}_1^2 - \hat{q}_3^2 & 2(\hat{q}_2\hat{q}_3 - \hat{q}_1\hat{q}_4) \\ 2(\hat{q}_1\hat{q}_3 - \hat{q}_2\hat{q}_4) & 2(\hat{q}_2\hat{q}_3 + \hat{q}_1\hat{q}_4) & \hat{q}_3^2 + \hat{q}_4^2 - \hat{q}_1^2 - \hat{q}_2^2 \end{bmatrix} \tag{32}$$

where $\hat{q}_i$ is the actual i^{th} component of the quaternion in $\hat{x}_k$ in EKF of the orientation estimation. Position and velocity estimation are performed in a KF assuming the results of EKF for orientation estimation are already available.

The IMU input of the orientation estimation is in K_S. Assuming that the axes of K_H and K_S are parallel and the body of the helicopter is rigid, the results are valid in K_S and K_H as well.

The position subsystem is also use the kinematic model of a moving body. In this case measured acceleration from the IMU is in K_S. According to [6] its transformation to K_H requires the angular acceleration, which is hard to estimate precisely. Therefore the state estimation is performed in K_S, hence the vision system should measure the position of the origin of K_S. In this case the kinematic model is:

$$\dot{v}_S = -\omega \times v_S + a_{m,S} - R_{H,W}^T (0\,0\,g)^T + a_b + a_n \tag{33}$$

$$\dot{a}_b = a_{b,n} \tag{34}$$

$$\dot{\rho}_W = R_{H,W} v_H + v_n, \tag{35}$$

where a refers to acceleration, v to velocity, ρ to sensor position, indices b to bias, n to noise. Notations S and W mean that the values are in K_S and K_W respectively.

Similarly to the orientation estimation the discrete time model can be evolved using the Euler method beside the following choice:

$$x_k = \left(v_{S,k}^T \ a_{b,k}^T \ \rho_{W,k}^T\right)^T \quad v_k = \left(a_{n,k}^T \ a_{b,n,k}^T \ v_{n,k}^T\right)^T \quad u_k = a_{m,S,k} \tag{36}$$

In this case the discrete time kinematic model will be linear in the state variables, therefore only (26)-(31) should be executed.

The dynamic model is described for the mass center of the helicopter. For a proper control, its position and velocity should be estimated. Therefore every value should be transform to K_W:

$$\hat{v}_W = R_{H,W} \left(\hat{v}_S + (\omega_m - \omega_b) \times p_S\right) \tag{37}$$

$$\hat{p}_W = \hat{\rho}_W - R_{H,W} p_S, \tag{38}$$

where $\hat{v}$ and $\hat{\rho}$ are the velocity and position component of position estimation's result.

IV. TIME-DELAY HANDLING

In section III it was assumed that the data from the vision system is available immediately. This assumption need to be checked. Let us introduce an other approach. The IMU is on the board of the helicopter, it has direct connection to the main processing unit. The communication can be fast enough and according to [7] the IMU need no time-consuming signal processing. Therefore these measurements can be considered immediately available.

The position and orientation can be estimated by the numeric integration of the IMU data. This approach has the disadvantage that the result will drift because of the bias error of the measurement. But according to [7] after the start of integration, the drift error is small enough to use in the following test.

The first data of Fig. 4 is the measurement from the camera. The second one is from the numeric integration of the angular velocity based on:

$$\hat{q}_{k+1} = \hat{q}_k + T_S \Gamma \omega_k \tag{39}$$

$$\hat{q}_{k+1} = \frac{\hat{q}_{k+1}}{\|\hat{q}_{k+1}\|} \tag{40}$$

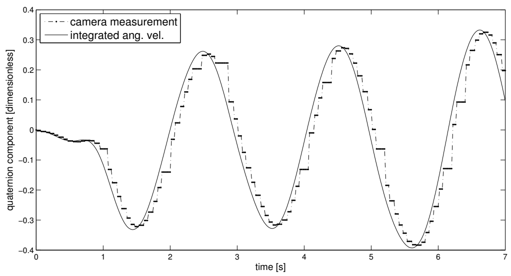

Fig. 4. Time delay of the vision system

For the sake of distinctness only the third element of the quaternion is shown in Fig. 4. It can be seen that the vision-loop (helicopter movement-camera-vision processing-RF communication-onboard processing) has significant delay. Its amount is comparable to the time constant of the motor dynamic hence this problem should be handled.

A. Orientation handling

In this solution it will be assumed that the amount of time delay is the integer multiple of the sampling time, denoted with d. Now in the k^{th} sampling point, the most relevant known coherent measurement pairs are from the $(k-\delta)^{th}$ sampling point. Therefore the EKF in (25)-(31) should be calculated using the following choice:

$$x_k = (q_{k-\delta}^T \ \omega_{b,(k-\delta)}^T)^T \quad u_k = \omega_{m,(k-\delta)} \quad y_k = q_{m,k} \tag{41}$$

The first four element of $\hat{x}_k$ is the estimated orientation of the $(k-\delta)^{th}$ sampling time. At time k the measurement between $k-\delta$ and k sampling points are available only from the IMU. Therefore a useful approximation may be the integration of the IMU data. In the case of orientation

$$\hat{q}_{i+1} = \hat{q}_k + T_S \Gamma(\omega_{m,i} + \hat{\omega}_{b,k-\delta}) \tag{42}$$

$$\hat{q}_{i+1} = \frac{\hat{q}_{i+1}}{\|\hat{q}_{i+1}\|} \tag{43}$$

can be repeated for d times. The result will be the coherent approximation of the orientation of the k^{th} sampling point.

The result of this method can be seen on Fig. 5. In this test, the reference is the numeric integration of the angular velocity. The estimations were performed for the same set of data with the estimator in Section III and with the estimator with time-delay handling. The rare vision measurement is also shown. For every orientation estimation and measurement the α value in (14) is calculated and the difference from the reference is shown.

It can be seen that estimation with time-delay handling performs better than the normal extended Kalman filter. The remaining error is mainly because of the fact that the time-delay is not exactly integer multiple of the sampling time.

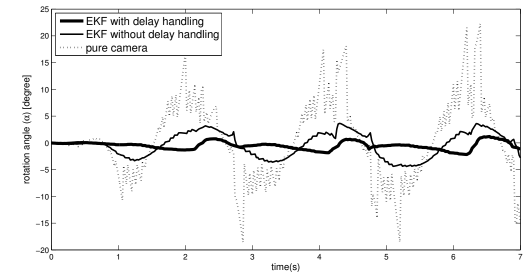

Fig. 5. The error of the different estimators

B. Position handling

The time-delay handling in the case of position estimation is similar to the orientation. The Kalman filter should be used with the $(k-\delta)^{th}$ sample, then the velocity and position should be numerically integrated from the IMU measurement. Therefore the following choice is suggested:

$$x_k = \left(v_{S,(k-\delta)}^T \ a_{b,(k-\delta)}^T \ \rho_{W,(k-\delta)}^T \right)^T \tag{44}$$

$$u_k = a_{m,S,(k-\delta)} \tag{45}$$

$$y_k = \rho_{m,k} \tag{46}$$

The recursive integration is:

$$v_{S,i+1} = v_{S,i} + T_S(a_{m,i} + \hat{a}_{b,k-\delta} - R_{H,W,i}^T (0\,0\,g)^T) \tag{47}$$

$$\rho_{W,i+1} = \rho_{W,i} + T_S v_{S,i+1}, \tag{48}$$

where $R_{H,W,i}$ is the rotation matrix based on $\hat{q}_i$

Finally, transformation (37)-(38) between K_S and K_H is used.

The full structure of the state estimation can be seen in Fig. 6.

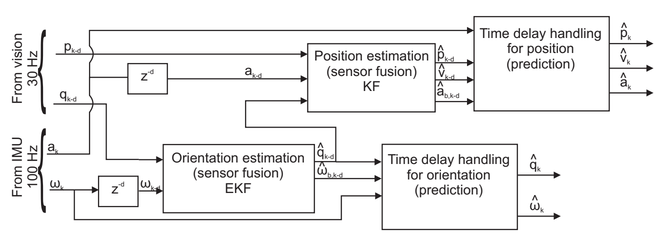

Fig. 6. The structure of the state estimation

V. CONTROL DESIGN

For the control the dynamic model is separated into four part. The control design is performed in force-torque space and the desired force and torque are transformed to the desired revolution of the rotors based on the inverse of (1) and (3)-(5). Fig. 7 shows the scheme of the control.

An approximation is applied in the control design. Let $(\varphi \ \vartheta \ \psi)^T$ denote the roll-pitch-yaw (Euler) angles calculated

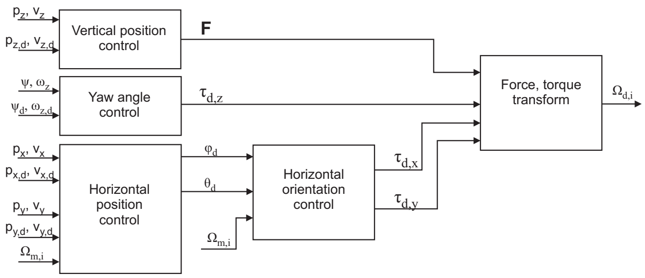

Fig. 7. Structure of the control

from $R_{H,W}^T$. For movements satisfying $\varphi \approx 0$ and $\vartheta \approx 0$, it was assumed that

$$(\dot{\varphi} \quad \dot{\vartheta} \quad \dot{\psi})^T \approx (\omega_x \quad \omega_y \quad \omega_z)^T \tag{49}$$

The trim values appearing in our control algorithms play similar role like V_r in [1]. Unfortunately the proof of theorem 1 in [1] has errors for Case 2 hence the results are not comparable.

A. Horizontal orientation subsystem

During the development process, different types of control methods were tested on the helicopter like in [2], [6] and [8]. Based on the experiments it can be said that the precision of this part of control is the key of the successful control. In our tests the following control produced the best results.

In this level two parallel control is designed for two very similar dynamic model, which is derived from (11) and (13). The first is the control around the X axis:

$$c_x = \frac{\theta_y - \theta_z}{\theta_x}\dot{\vartheta}\dot{\psi} - \frac{\theta_r}{\theta_x}\dot{\vartheta}\Omega_r \tag{50}$$

$$\ddot{\varphi} = \frac{\tau_{r,x}}{\theta_x} + c_x \tag{51}$$

$$\dot{\tau}_{r,x} = \frac{1}{T_m}\left(\tau_{d,x} - \tau_{r,x}\right) \tag{52}$$

The second is the control around the Y axis:

$$c_y = \frac{\theta_z - \theta_x}{\theta_y}\dot{\varphi}\dot{\psi} - \frac{\theta_r}{\theta_y}\dot{\varphi}\Omega_r \tag{53}$$

$$\ddot{\vartheta} = \frac{\tau_{r,y}}{\theta_y} + c_y \tag{54}$$

$$\dot{\tau}_{r,y} = \frac{1}{T_m}\left(\tau_{d,y} - \tau_{r,y}\right). \tag{55}$$

The control synthesis is shown only for the X axis. Let e be the control error $e = \varphi - \varphi_d$. The behavior of the closed loop dynamic is defined with the prescribed characteristic equation

$$e^{(3)} + k_1\ddot{e} + k_2\dot{e} + k_3 e + k_4\int e dt = 0, \tag{56}$$

where k_i is control parameter.

$$e = \varphi - \varphi_d \tag{57}$$

$$\dot{e} = \dot{\varphi} - \dot{\varphi}_d \tag{58}$$

$$\ddot{e} = \frac{\tau_{r,x}}{\theta_x} + c_x - \ddot{\varphi}_d \tag{59}$$

$$e^{(3)} = \frac{1}{\theta_x T_m}\left(\tau_{d,x} - \tau_{r,x}\right) + \dot{c}_x - \varphi_d^{(3)} \tag{60}$$

The desired torque is

$$\tau_{d,x} = \tau_{r,x} + \theta_x T_m \left(\varphi_d^{(3)} - \dot{c}_x - k_1\left(\frac{\tau_{r,x}}{\theta_x} + c_x\right) - k_2\left(\dot{\varphi} - \dot{\varphi}_d\right) - \right.$$
$$\left. - k_3\left(\varphi - \varphi_d\right) - k_4\int\left(\varphi - \varphi_d\right)dt\right) \tag{61}$$

In practice $\varphi_d^{(3)} - \dot{c}$ is approximated with 0. Also based on the experiments, an additional trimmer parameter was introduced. Its reason is that the parameter b in (1) is not the same for every propeller. The control law after this modification is

$$\tau_{d,x} = \tau_{r,x} - \tau_{trim,x} + \theta_x T_m\left(-k_1\left(\frac{\tau_{r,x} - \tau_{trim,x}}{\theta_x} + c_x\right) - \right.$$
$$\left. - k_2\left(\dot{\varphi} - \dot{\varphi}_d\right) - k_3\left(\varphi - \varphi_d\right) - k_4\int\left(\varphi - \varphi_d\right)dt\right) \tag{62}$$

B. Horizontal position subsystem

This part of the control is easier to design in K_W. Therefore let U_x and U_y define the angles between the X-Y plane of K_H and the X and Y axes of K_W. Let's introduce the following approximation

$$U_x \approx \frac{F_x}{F} \qquad U_y \approx \frac{F_y}{F}, \tag{63}$$

where F is amount of the sum of the lift forces and F_x, F_y are the X and Y component of the external force in K_W. Then the dynamic model is

$$\ddot{x} = \frac{F}{m}U_x \qquad \ddot{y} = \frac{F}{m}U_y \tag{64}$$

The control synthesis is similar to that of horizontal orientation subsystem. The control law for U_x is

$$U_{x,d} = \frac{m}{F}\left(\ddot{x}_d - k_1(\dot{x} - \dot{x}_d) - k_2(x - x_d) - k_3\int(x - x_d)dt\right) \tag{65}$$

Finally $(U_x\, U_y)^T$ can be transformed to K_H with

$$\begin{pmatrix} \vartheta \\ -\varphi \end{pmatrix} = \begin{bmatrix} cos(\psi) & -sin(\psi) \\ sin(\psi) & cos(\psi) \end{bmatrix}\begin{pmatrix} U_x \\ U_y \end{pmatrix}. \tag{66}$$

C. Altitude control

In this subsystem the dynamic model in K_W is

$$\ddot{z} = \frac{\cos(\varphi)\cos(\vartheta)}{m}F - g \tag{67}$$

Similarly to the previous controls the control law is

$$F_d = \frac{m}{\cos(\varphi)\cos(\vartheta)}(g + \ddot{z}_d - k_1(\dot{z} - \dot{z}_d) - k_2(z - z_d) -$$
$$- k_3\int(z - z_d)dt) \tag{68}$$

D. Yaw control

In the case of yaw control the following model is used:

$$\ddot{\psi} = \frac{\tau_z}{\theta_z} \tag{69}$$

In our case $\theta_x \approx \theta_y$ because of the symmetry of the helicopter body. Hence the non-linear part in (13) can be neglected.

Based on our experiments in this part of the control it has no significant difference whether the motor dynamic is controlled or not. An additional trimmer parameter is used here similarly to the horizontal orientation control. The control law for this subsystem is

$$\begin{aligned}\tau_{d,z} = \tau_{trim,z} + \theta_z(\ddot{\psi}_d - k_1(\dot{\psi} - \dot{\psi}_d) - k_2(\psi - \psi_d) - \\ - k_3 \int (\psi - \psi_d)dt)\end{aligned} \tag{70}$$

VI. RESULTS

Experimental results from real flights are shown in this section. In the recent state of our realization, we use only one camera as a vision system, therefore the operation volume is a cone with 1m radius and 2m height. The starting transient of the integrators are too large beside this volume, therefore they were switched off during the tests. The task of the first test is that the helicopter should hold its position. The result is shown in Fig. 8. The working area was too small for lateral path follow tests.

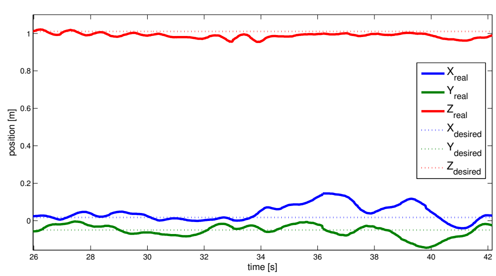

Fig. 8. Position hold test

The second test is of the altitude control beside a sine wave reference signal. The result is shown in Fig. 9.

The third test is the path follow of the yaw control. The result is in Fig. 10.

VII. CONCLUSION

This paper presents the state estimation and control system of an indoor quadrotor helicopter. The state estimation performs as sensor fusion between the IMU and the vision system using extended and linear Kalman filter methods. The estimator handles the time delay of the vision system. The control is based on the dynamics of the quadrotor including the motor dynamics. The control is extended with trimmer parameters based on experimental results. The system is tested during real flights.

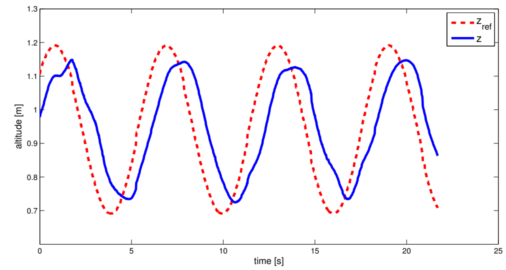

Fig. 9. Test of the altitude control

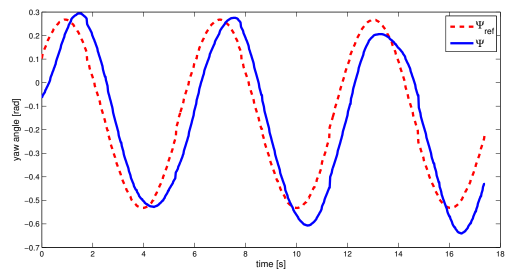

Fig. 10. Yaw control test

ACKNOWLEDGMENTS

This work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Hungary Development Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

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

REFERENCES

[1] A. Das, K. Subbarao, F. Lewis, Dynamic inversion with zero-dynamics stabilisation for quadrotor control. *IET Contr. Theory & Applications*, 303-314, 2009.

[2] S. Bouabdallah, A. Noth, R.Siegwart PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Sendai 2004

[3] S. Bouabdallah and R. Siegwart, Full Control of a Quadrotor *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, 2007

[4] G. M. Hoffmann, H. Huang, S. L. Waslander C. J. Tomlin, Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2004 Hilton Head, South Carolina

[5] J. Farrell and M. Barth, The Global Positioning System and Inertial Navigation, *McGraw-Hill*, ISBN 0-07-022045-X.

[6] L. Kis, G. Regula, B. Lantos, Design and Hardware-in-the-Loop Test of the Embedded Control System of an Indoor Quadrotor Helicopter, *6th Workshop on Intelligent Solutions in Embedded Systems*, Regesburg, Germany, 2008

[7] L. Kis and B. Lantos, Sensor Fusion and Actuator System of a Quadrotor Helicopter, *Periodica Polytechnica-Electrical Engineering*, 2009:(53/3-4) pp. 139-150.

[8] L. Kis, B. Lantos,Quadrotor Control Based on Partial Sensor Data, *INTERNATIONAL Journal of Mechanics and Control* 12:(1) pp. 43-50. (2011)