periodica polytechnica

Electrical Engineering
53/3-4 (2009) 151–161
doi: 10.3311/pp.ee.2009-3-4.07
web: http://www.pp.bme.hu/ee

RESEARCH ARTICLE

Backstepping based control design with state estimation and path tracking to an indoor quadrotor helicopter

Gergely Regula / *Béla* Lantos

Received 2010-05-25

Abstract

The article focuses on different aspects (both theoretical and practical) of the development of the control algorithm of a quadrotor helicopter starting from the modelling phase. A new control algorithm is elaborated and supplementary components are described in detail including state estimation and path tracking. The helicopter's dynamic model takes into account the aerodynamic friction, the gyroscopic effect of the rotors and also the motor dynamics. The control algorithm is based on the backstepping approach and is capable of stabilising the model even in case of realistic noises. Vision system and on-board inertial measurement unit provide the measurements and two-level extended Kalman filter based state estimator is used to suppress the measurement noises and to estimate the unmeasured signals. The methods of the software development and real-time testing are also presented with attention to the sources of common errors.

Keywords

UAV · quadrotor helicopter · state estimation · extended Kalman filtering · implementation · hardware-in-the-loop test

Acknowledgement

The research work presented in this article was supported by the Hungarian National Research Program under grant No. OTKA K 71762.

Gergely Regula
Department of Control Engineering and Information Technology, BME, H-1117 Budapest Magyar Tudósok krt. 2., Hungary
e-mail: regula@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

Presently, unmanned aerial vehicles (UAVs) increasingly attract the attention of potential appliers, vehicle professionals and researchers. UAV field seems to step out of the exclusivity of the military applications, a lot of potential civil applications have emerged, research and development in this field have gained increasing significance. Research teams affiliated with the authors are interested in many respects in this field, many efforts are spent in several research areas in connection with the individual and cooperative control of aerial and land vehicles including the unmanned ones. Developing an unmanned mini quadrotor helicopter that is able to execute autonomously a mission, e.g. performing a series of measurements in predefined positions, or completing a surveillance task above a given territory is one of the goals formulated for the near future.

Our project initiated in the spring of 2006 as a cooperation between Budapest University of Technology and Economics Dept. Control Engineering and Information Technology (BME IIT) and the Computer and Automation Research Institute of the Hungarian Academy of Sciences (MTA SZTAKI). In the current phase, the primary goal of this research is to build an autonomous indoor quadrotor helicopter, which will serve later as a research test bed for advanced algorithms in the areas of control, path planning, manoeuvring in formation, position estimation, sensor fusion, embedded real-time (RT) vision system, spatial map building, robust and efficient mechanical and aerodynamic construction.

The development started in five areas: control system algorithms, system architecture, electronic components, mechanical and aerodynamic design and vision system. The first version of the helicopter body was reported in [1], [2]. In 2008 the hardware-in-the-loop tests were successfully performed, which verified that the on-board MPC555 CPU can be programmed via Simulink to perform the backstepping control and extended Kalman filter algorithms together with CAN bus I/O in real time [3].

This article mainly focuses on the development of the control algorithm of the vehicle, introducing a new backstepping based algorithm that has evolved from parallel and our researches' re-

sults. Other important components (state estimation based on extended Kalman filtering and a simple path tracking method) are also presented in later sections. The article is structured as follows. After the description of the system (Section 2), modelling of the helicopter and the rotor system is discussed in Sections 3. Section 4, 5, 6 present the components of the control system, namely, the backstepping based control algorithm, the state estimator and the path tracking method, respectively. Simulations results are presented in Section 7 while Section 8 addresses the implementation methods and software issues. The article ends with a short conclusion and summary of the results.

2 System overview

The control loop of the helicopter requires accurate position and orientation information. The primary sensor for this is a Crossbow MNAV100CA inertial measurement unit (IMU), which provides acceleration and angular velocity measurements. Based on the physical properties of the sensors [4], the raw sensor output is loaded with noise and bias errors, which can build up increasing orientation and position error during integration. In order to reduce the effect of noise and bias, extended Kalman filters are used instead of integration. Thus, for 6 degrees of freedom mobile robots, one needs absolute measurements performed frequently. For outdoor autonomous vehicles, GPS or differential GPS is often used. For indoor vehicles, vision based measurements are widely used to compensate integration errors.

The overview of our system architecture is as follows. The helicopter will have an on-board CPU. The rotors are driven by brushless DC (BLDC) motors. The motor controllers and the IMU are connected to the CPU via CAN bus. Also, a spread spectrum code division multiplexing (CDM) radio link is connected to the CPU, providing bidirectional communication between the quadrotor and the ground station. The ground station sends commands and reference path to the CPU, together with the absolute position measurements. The helicopter sends status information to the ground. The ground station is connected to the computer performing vision algorithms via Ethernet cable. We use commercial, high resolution web cameras for imaging. More details on the sensor and actuator system can be found in [4].

The on-board computer is a phyCORE-MPC555, since it is lightweight, equipped with a floating point unit and can be programmed in MATLAB/Simulink. An alternative goal of the project is to compare high level model based programming of embedded systems with traditional methods.

3 Modelling the quadrotor helicopter's dynamics

3.1 The equations of motion of the helicopter

Let us assume that a frame (coordinate system) K_E fixed to the Earth can be considered as an inertial frame of reference. The frame fixed to the centre of gravity of the helicopter K_H can be described by its position $\xi = (x, y, z)^T$ and orientation (RPY angles) $\eta = (\Phi, \Theta, \Psi)^T$ relative to K_E. The orientation can be described by the matrix R_t in the following way:

$$R_t = \begin{bmatrix} C_\Theta C_\Psi & S_\Phi S_\Theta C_\Psi - C_\Phi S_\Psi & C_\Phi S_\Theta C_\Psi + S_\Phi S_\Psi \\ C_\Theta S_\Psi & S_\Phi S_\Theta S_\Psi + C_\Phi C_\Psi & C_\Phi S_\Theta S_\Psi - S_\Phi C_\Psi \\ -S_\Theta & S_\Phi C_\Theta & C_\Phi C_\Theta \end{bmatrix} \tag{1}$$

where S_x and C_x denote $\sin(x)$ and $\cos(x)$ as usual in robotics.

The relation between $\dot{\xi}$ and $\dot{\eta}$ and translational and angular velocities v and ω of the helicopter take the form

$$\begin{aligned} \dot{\xi} &= R_t v \\ \omega &= R_r \dot{\eta} \end{aligned} \tag{2}$$

where time derivative is denoted by dot and the matrix R_r has the form

$$R_r = \begin{bmatrix} 1 & 0 & -S_\Theta \\ 0 & C_\Phi & S_\Phi C_\Theta \\ 0 & -S_\Phi & C_\Phi C_\Theta \end{bmatrix} \tag{3}$$

It is worth mentioning that the inverse of R_r can be computed as

$$R_r^{-1} = \begin{bmatrix} 1 & S_\Phi T_\Theta & S_\Phi T_\Theta \\ 0 & C_\Phi & -S_\Phi \\ 0 & S_\Phi / C_\Theta & C_\Phi / C_\Theta \end{bmatrix} \tag{4}$$

and the derivative of ω can be written as

$$\beta = \dot{\omega} = R_r \ddot{\eta} + \dot{R}_r \dot{\eta} \tag{5}$$

Applying Newton's laws, the translational and rotational motions of the helicopter in K_H are described by

$$\begin{aligned} \sum F_{ext} &= m\dot{v} + \omega \times (m\dot{v}) \\ \sum T_{ext} &= I\dot{\omega} + \omega \times (I\omega) \end{aligned} \tag{6}$$

where I is the inertia matrix of the helicopter and it is supposed that it can be described by a diagonal matrix $I = \mathrm{diag}(I_x, I_y, I_z)$. $\sum F_{ext}$ and $\sum T_{ext}$ represent the forces and torques respectively applied to the quadrotor helicopter expressed in K_H. These forces and torques are partly caused by the rotation of the rotors (F and T), the aerodynamic friction (F_a and T_a), the gravitational effect (F_g) in the translational motion and the gyroscopic effect (T_g) in the rotational motion.

$$\begin{aligned} \sum F_{ext} &= F + F_a + F_g \\ \sum T_{ext} &= T + T_a + T_g \end{aligned} \tag{7}$$

The helicopter has four actuators (four brushless DC motors) which exert a lift force proportional to the square of the angular velocities Ω_i of the actuators ($f_i = b\Omega_i^2$). The BLDC motors' reference signals can be programmed in Ω_i. The resulting torque and lift force are

$$\begin{aligned} T &= \begin{pmatrix} lb(\Omega_4^2 - \Omega_2^2) \\ lb(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{pmatrix} \\ f &= f_1 + f_2 + f_3 + f_4 = b\sum_{i=1}^{4} \Omega_i^2 \end{aligned} \tag{8}$$

where l, b, d are helicopter and rotor constants. The force F can then be rewritten as $F = (0, 0, f)^T$.

The gravitational force points to the negative z-axis, therefore $F_g = -mR_t^T(0, 0, g)^T = -mR_t^T G$. The gyroscopic effect can be modelled as

$$T_g = -(\omega \times k)I_r(\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3) = -\omega \times (I_r\Omega_r) \quad (9)$$

where I_r is the rotor inertia and k is the third unit vector.

The aerodynamic friction at low speeds can well be approximated by the linear formulas $F_a = -K_t v$ and $T_a = -K_r\omega$.

Using the equations above we can derive the equations of motion of the helicopter:

$$\begin{aligned} F &= mR_t^T\ddot{\xi} - K_t R_t^T\dot{\xi} - mR_t^T G \\ T &= IR_r\ddot{\eta} + I\left(\frac{\partial R_r}{\partial\Phi}\dot{\Phi} + \frac{\partial R_r}{\partial\Theta}\dot{\Theta}\right)\dot{\eta} + \\ &\quad + K_r R_r\dot{\eta} + (R_r\dot{\eta}) \times (IR_r\dot{\eta} + I_r\Omega_r) \end{aligned} \quad (10)$$

3.2 Simplified dynamic equations

A simplified model of the quadrotor helicopter can be obtained by neglecting certain effects and applying reasonable approximations. The purpose of the construction of such a model is to reduce the complexity of the controller while keeping its performance.

Since the helicopter's motion is planned to be relatively slow, it is reasonable to neglect all the aerodynamic effects, namely, K_t and K_r are approximately zero matrices. The other simplification is also related to the low speeds. Slow motion in lateral directions means little roll and pitch angle changes, therefore R_r can be approximated by a 3-by-3 unit matrix. Such simplification cannot be applied to R_t.

Consequently, the dynamic equations in (10) become

$$\begin{aligned} F &\approx mR_t^T\ddot{\xi} - mR_t^T G \\ T &\approx I\ddot{\eta} + \dot{\eta} \times (I\dot{\eta} + I_r\Omega_r) \end{aligned} \quad (11)$$

The six equations in detail are the ones that can be found in [5] and [3].

$$\begin{aligned} m\ddot{x} &\approx (C_\Phi S_\Theta C_\Psi + S_\Phi S_\Psi) f \\ m\ddot{y} &\approx (C_\Phi S_\Theta S_\Psi - S_\Phi C_\Psi) f \\ m\ddot{z} &\approx C_\Phi C_\Theta f - mg \\ I_x\ddot{\Phi} &\approx \dot{\Theta}\dot{\Psi}(I_y - I_z) - I_r\dot{\Theta}\Omega_r + T_1 \\ I_y\ddot{\Theta} &\approx \dot{\Psi}\dot{\Phi}(I_z - I_x) - I_r\dot{\Phi}\Omega_r + T_2 \\ I_z\ddot{\Psi} &\approx \dot{\Phi}\dot{\Theta}(I_x - I_y) + T_3 \end{aligned} \quad (12)$$

3.3 Rotor dynamics

The four BLDC motors' dynamics can be described as ($k = 1, \ldots, 4$):

$$\begin{aligned} L\dot{i}_k &= u_{m,k} - Ri_k - k_e\Omega_k \\ I_r\dot{\Omega}_k &= k_m i_k - k_r\Omega_k^2 - k_s \end{aligned} \quad (13)$$

where k_e, k_m and k_s represent the back EMF constant, the motor torque constant and the friction constant, respectively. If the motors' inductance is negligible, (13) can be rewritten to

$$\dot{\Omega}_k = -k_{\Omega,0} - k_{\Omega,1}\Omega_k - k_{\Omega,2}\Omega_k^2 + k_u u_{m,k} \quad (14)$$

4 The construction of the helicopter's control algorithm

4.1 Related research

There are numerous control algorithms that can be applied to a quadrotor helicopter including linear [6], nonlinear and even soft computing techniques [7]. Among the nonlinear control algorithms, the backstepping approach has gained the most attention, although several other methods are elaborated including sliding mode [5] and feedback linearisation control algorithms [8].

These pieces of research differ from each other not only on the control algorithm, but also on the types of simplification of the dynamic model of the helicopter. Several methods exist for dynamic models that retain the basic behaviour of the vehicle. Some neglect the rotor dynamics assuming the transients of the rotors are fast compared to those of the helicopter, some others do not consider the aerodynamics or the gyroscopic effect.

A full state backstepping algorithm is presented in [9], where the control law is obtained step by step through three virtual subsystems' stabilisation. The quadrotor dynamic model described in the previous section is similar to that in this work. In [5], [2] and [3], a backstepping algorithm is applied to simplified helicopter dynamic model. These are the base of the algorithm that is presented in this section.

The following parts of the present article focus on the construction of such an algorithm that is capable of explicitly handling all the effects appearing in (10), while being ignorant to realistic measurement noises.

4.2 Applying a backstepping algorithm to the helicopter

The control algorithm has evolved from the results of [5] and our research [3]. The algorithm presented in this part intends to exploit the advantages of two approaches, that are the ability to control a dynamic model with the least possible simplification and the good handling of measurement noises experienced in the case of our earlier algorithm based on [3].

First, we have to reformulate the equations (10) and (13).

$$\begin{aligned} \ddot{\xi} &= f_\xi + g_\xi u_\xi \\ \ddot{\eta} &= f_\eta + g_\eta u_\eta \\ \dot{\Omega}_k &= f_{\Omega,k} + g_{\Omega,k}u_{\Omega,k} \end{aligned} \quad (15)$$

where f_ξ, g_ξ and u_ξ are

$$\begin{aligned} f_\xi &= -G - \frac{1}{m}R_t K_t R_t^T\dot{\xi} \\ g_\xi &= \frac{1}{m}\operatorname{diag}(r_{t,3}) \\ u_\xi &= (f, f, f)^T \end{aligned} \quad (16)$$

while f_η, g_η and u_η stand for

$$\begin{aligned} f_\eta &= (I R_r)^{-1} \left[-I \left(\frac{\partial R_r}{\partial \Phi} \dot{\Phi} + \frac{\partial R_r}{\partial \Theta} \dot{\Theta} \right) \dot{\eta} - \right. \\ &\left. -K_r R_r \dot{\eta} - (R_r \dot{\eta}) \times (I R_r \dot{\eta} + I_r \Omega_r) \right] \\ g_\eta &= (I R_r)^{-1} \\ u_\eta &= T \end{aligned} \tag{17}$$

and $f_{\Omega,k}$, $g_{\Omega,k}$ and $u_{\Omega,k}$ yield

$$\begin{aligned} f_{\Omega,k} &= -k_{\Omega,0} - k_{\Omega,1}\Omega_k - k_{\Omega,2}\Omega_k^2 \\ g_{\Omega,k} &= k_u \\ u_{\Omega,k} &= u_{m,k} \end{aligned} \tag{18}$$

Since the vector F contains only one nonzero element,

$$g_\xi u_\xi = \frac{1}{m} r_{t,3} f = \frac{1}{m} \operatorname{diag}(r_{t,3}) u_\xi \tag{19}$$

where $r_{t,3}$ is the third column of R_t.

The construction of the control law is similar to that presented in [5]. Since the helicopter is underactuated, the concept is that the helicopter is required to track a path defined by its (x_d, y_d, z_d, Ψ_d) coordinates. The helicopter's roll and pitch angles are stabilised to 0 internally. The control algorithm can be divided into three main parts. At first, the translational part of the vehicle dynamics is controlled, which then produces the two missing reference signals to the attitude control system. The third part is responsible for generating the input signals of the BLDC motors. The hierarchical structure of the controller is shown in Fig. 1, where indices d and m denote desired and measured values, respectively. The speed ratio of the three parts of the hierarchical structure depends on the physical properties of the components, especially on the measurement frequency of the sensors. The ideal values of the sampling times for position and orientation control are between 10-30 ms . Kalman filters can tolerate the difference of measurement frequencies of the position and orientation (vision system) and acceleration and velocity (IMU). The sampling time of the motor control is set to 10 ms .

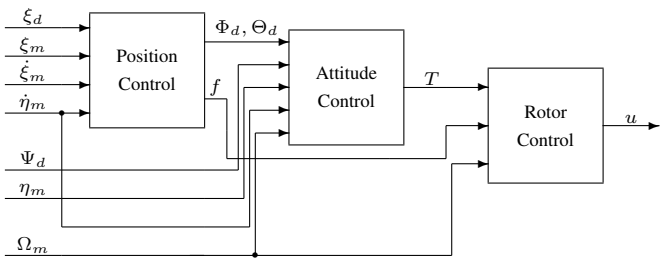


Fig. 1. The hierarchical structure of the controller

4.2.1 Position control

The concept of backstepping control will be explained for position control. First, let us define the path tracking error

$$q_{\xi_1} = \xi_d - \xi \tag{20}$$

Applying Lyapunov's theorem we construct a positive definite function whose time derivative is negative definite.

$$V(q_{\xi_1}) = \frac{1}{2} q_{\xi_1}^T q_{\xi_1} \tag{21}$$

$$\dot{V}(q_{\xi_1}) = q_{\xi_1}^T \dot{q}_{\xi_1} = q_{\xi_1}^T (\dot{\xi}_d - \dot{\xi}) \tag{22}$$

If we were free to choose

$$\dot{\xi} = \dot{\xi}_d + A_{\xi_1} q_{\xi_1} \tag{23}$$

then the time derivative of the Lyapunov function would be

$$\dot{V}(q_{\xi_1}) = -q_{\xi_1}^T A_{\xi_1} q_{\xi_1} < 0 \tag{24}$$

if the matrix A_{ξ_1} were positive definite. Therefore, introducing a virtual tracking error

$$q_{\xi_2} := \dot{\xi} - \dot{\xi}_d - A_{\xi_1} q_{\xi_1} = -\dot{q}_{\xi_1} - A_{\xi_1} q_{\xi_1} \tag{25}$$

and an augmented Lyapunov function

$$V(q_{\xi_1}, q_{\xi_2}) = \frac{1}{2} \left(q_{\xi_1}^T q_{\xi_1} + q_{\xi_2}^T q_{\xi_2} \right) \tag{26}$$

results in the possibility to find the actual lift force f needed. The augmented Lyapunov function's time derivative is

$$\begin{aligned} \dot{V}(q_{\xi_1}, q_{\xi_2}) &= q_{\xi_1}^T \dot{q}_{\xi_1} + q_{\xi_2}^T \dot{q}_{\xi_2} = q_{\xi_1}^T (-\dot{q}_{\xi_1} - A_{\xi_1} q_{\xi_1}) + \\ &+ q_{\xi_2}^T (\ddot{\xi} - \ddot{\xi}_d - A_{\xi_1} (-\dot{q}_{\xi_1} - A_{\xi_1} q_{\xi_1})) = \\ &= -q_{\xi_1}^T A_{\xi_1} q_{\xi_1} - q_{\xi_2}^T q_{\xi_1} + q_{\xi_2}^T (f_\xi + g_\xi u_\xi) - \\ &- q_{\xi_2}^T [\ddot{\xi}_d - A_{\xi_1} (q_{\xi_2} + A_{\xi_1} q_{\xi_1})] \end{aligned} \tag{27}$$

We are now free to choose

$$\begin{aligned} u_\xi &= g_\xi^{-1} [q_{\xi_1} - f_\xi + \ddot{\xi}_d - A_{\xi_1} (q_{\xi_2} + A_{\xi_1} q_{\xi_1}) - A_{\xi_2} q_{\xi_2}] = \\ &= g_\xi^{-1} [\ddot{\xi}_d - f_\xi + (I_3 + A_{\xi_2} A_{\xi_1}) q_{\xi_1} + (A_{\xi_2} + A_{\xi_1}) \dot{q}_{\xi_1}] \end{aligned} \tag{28}$$

where I_3 is a unit matrix. It could be assumed here that $\ddot{\xi}_d$ is negligible as in [5]. However, the further goal is that the helicopter tracks certain waypoints, which means that it is in continuous motion. Therefore, $\ddot{\xi}_d$ does have an important role in the control. If A_{ξ_2} is positive definite, the time derivative of the Lyapunov function is

$$\dot{V}(q_{\xi_1}, q_{\xi_2}) = -q_{\xi_1}^T A_{\xi_1} q_{\xi_1} - q_{\xi_2}^T A_{\xi_2} q_{\xi_2} < 0 \tag{29}$$

Applying the control law (28) to (15) results in

$$\ddot{\xi} = \ddot{\xi}_d + (I_3 + A_{\xi_2} A_{\xi_1}) q_{\xi_1} + (A_{\xi_2} + A_{\xi_1}) \dot{q}_{\xi_1} \tag{30}$$

which is equivalent to

$$0 = (I_3 + A_{\xi_2} A_{\xi_1}) q_{\xi_1} + (A_{\xi_2} + A_{\xi_1}) \dot{q}_{\xi_1} + \ddot{q}_{\xi_1} \tag{31}$$

Assuming positive definite and diagonal A_{ξ_1}, A_{ξ_2} matrices with diagonal elements $a_{\xi_1,i}$, $a_{\xi_2,i}$, the characteristic equations have the form

$$s^2 + (a_{\xi_2,i} + a_{\xi_1,i})s + (1 + a_{\xi_2,i}a_{\xi_1,i}) = 0 \tag{32}$$

which guarantees stability.

This means that the errors exponentially converge to zero if the calculated values of f_ξ and g_ξ are close to the real ones.

Algebraic manipulations can be performed in $g_\xi u_\xi$. The third component of u_ξ is the lift force f. The other two components are for different purposes. Multiplying the formula of (28) in brackets by $\frac{m}{u}$ instead of the reciprocal of the appropriate element of g_ξ yields an expression for the third column of R_t. Since this change has no effect on the stability, then if the entire controlled system is stable, g_ξ has to be convergent and its limit is $(0, 0, 1)^T$. Therefore, the reference signals Φ_d and Θ_d can be obtained as follows. First we modify g_ξ and u_ξ:

$$\tilde{g}_\xi = \frac{1}{m}\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & C_\Phi C_\Theta \end{bmatrix} \tag{33}$$

$$\tilde{u}_\xi = \begin{pmatrix} C_\Phi S_\Theta C_\Psi + S_\Phi S_\Psi \\ C_\Phi S_\Theta S_\Psi - S_\Phi C_\Psi \\ f \end{pmatrix} = \begin{pmatrix} u_{\xi_x} \\ u_{\xi_y} \\ f \end{pmatrix} \tag{34}$$

We can extract f as before and then we obtain u_{ξ_x} and u_{ξ_y} using the elements of $\tilde{g}_\xi$ and

$$\begin{aligned} S_{\Phi_d} &= S_\Psi u_{\xi_x} - C_\Psi u_{\xi_y} \\ S_{\Theta_d} &= \frac{C_\Psi u_{\xi_x} + S_\Psi u_{\xi_y}}{C_\Phi} \end{aligned} \tag{35}$$

yield Φ_d and Θ_d. The reason why these signals can be considered as reference signals is that as the helicopter approaches the desired coordinates, they converge to zero. Conversely, if the helicopter follows the appropriate attitude and lift force, it will get to the desired position and orientation.

4.2.2 Attitude control

The design is similar to that described in the previous part. Again, let us define the attitude error

$$q_{\eta_1} = \eta_d - \eta \tag{36}$$

and introduce a virtual tracking error

$$q_{\eta_2} = \dot{\eta} - \dot{\eta}_d - A_{\eta_1} q_{\eta_1} = -\dot{q}_{\eta_1} - A_{\eta_1} q_{\eta_1} \tag{37}$$

Following the same steps the result is $T = u_\eta$ where

$$u_\eta = g_\eta^{-1}[\ddot{\eta}_d - f_\eta + (I_3 + A_{\eta_2}A_{\eta_1})q_{\eta_1} + (A_{\eta_2} + A_{\eta_1})\dot{q}_{\eta_1}] \tag{38}$$

4.2.3 A simplified control of the position and attitude

According to (10) and (11), a change in f_ξ, f_η and g_η results in simpler controller equations that are formally identical to (28) and (38).

$$\begin{aligned} f_\xi &= -G \\ f_\eta &= -I^{-1}\left[\dot{\eta} \times (I\dot{\eta} + I_r\Omega_r)\right] \\ g_\eta &= I^{-1} \end{aligned} \tag{39}$$

A further simplification is to assume that the second derivatives of the reference signals are zero ($\ddot{\xi}_d = 0$ and $\ddot{\eta}_d = 0$), thus these terms disappear from the control laws. In the following sections it will be shown that these simplifications do not deteriorate the overall performance of the control system.

4.2.4 Rotor control

There is a slight difference in the calculation of u_m compared to the other inputs since in the third equation of (15) only the first derivative of Ω_k appears. This means that there is no need for the virtual error q_{m_2} and the Lyapunov function remains $V(q_{m_1}) = \frac{1}{2}q_{m_1}^T q_{m_1}$. However, it is worth including the derivative of q_{m_1} similarly as in the previous sections because of the error dynamics.

$$\begin{aligned} u_m = g_m^{-1}[&\dot{\Omega}_d - f_m + (I_4 + A_{m_2}A_{m_1})q_{m_1} + \\ &+ (A_{m_2} + A_{m_1})\dot{q}_{m_1}] \end{aligned} \tag{40}$$

with q_{m_1} and f_m being

$$q_{m_1} = \begin{pmatrix} \Omega_{1_d} - \Omega_1 \\ \Omega_{2_d} - \Omega_2 \\ \Omega_{3_d} - \Omega_3 \\ \Omega_{4_d} - \Omega_4 \end{pmatrix} \quad \text{and} \quad f_m = \begin{pmatrix} f_{m,1} \\ f_{m,2} \\ f_{m,3} \\ f_{m,4} \end{pmatrix} \tag{41}$$

Since the four motors are considered to be identical, g_m can be any of $g_{m,k}$-s and therefore it is a scalar. It is worth noticing that since T and f are linear combinations of Ω_k^2, hence Ω_{k_d} are the element-wise square roots of

$$\begin{pmatrix} \Omega_{1_d}^2 \\ \Omega_{2_d}^2 \\ \Omega_{3_d}^2 \\ \Omega_{4_d}^2 \end{pmatrix} = \begin{bmatrix} 0 & -(2lb)^{-1} & -(4d)^{-1} & (4d)^{-1} \\ -(2lb)^{-1} & 0 & (4d)^{-1} & (4d)^{-1} \\ 0 & (2lb)^{-1} & -(4d)^{-1} & (4d)^{-1} \\ (2lb)^{-1} & 0 & (4d)^{-1} & (4d)^{-1} \end{bmatrix} \begin{pmatrix} T \\ f \end{pmatrix} \tag{42}$$

The time derivative of the Lyapunov function becomes

$$\begin{aligned} \dot{V}(q_{m_1}) &= q_{m_1}^T \dot{q}_{m_1} = \\ &= -q_{m_1}^T[(I + A_{m_2}A_{m_1})q_{m_1} - (A_{m_2} + A_{m_1})\dot{q}_{m_1}] = \\ &= -q_{m_1}^T(I + A_{m_2} + A_{m_1})^{-1}(I + A_{m_2}A_{m_1})q_{m_1} < 0 \end{aligned} \tag{43}$$

if A_{m_1} and A_{m_2} are positive definite matrices.

4.2.5 Tuning of controller parameters

The parameters of the controllers are related to coefficients of low order characteristic polynomials or elements of positive definite diagonal matrices, hence their choice is simple. The numerical values are immediately related to the speed of the control. It should also be taken into account that the increased speed of the control can cause saturation in the actuators. Simulation experiments can help in parameter tuning (see the results in Tab. 3).

5 Estimating the non-measured signals

The control algorithm introduced in the previous section requires the signals shown in Table 1 for the computation of the control inputs. The signals measured by the sensor system are marked by asterisk (*). Since there is a signal that is not measured and long time tests show that the inertial sensor's signals not only contain noises, but also an offset (bias) that changes slowly, a state estimator is included in the control algorithm.

Tab. 1. The signals required by the control algorithm

Signal	*Meaning*
ξ	Position*
$\dot{\xi}$	Velocity
η	Attitude*
$\dot{\eta}$	Angular velocity*
Ω	Angular velocities of the rotors*

The state estimator consists of two hierarchically structured extended Kalman filters that are responsible for the estimation of the attitude and position related signals (EKF_1 and EKF_2, respectively). The structure of the state estimator can be seen in Fig. 2. The role of the block in the middle of the diagram is to transform the measured acceleration from the sensor frame to the helicopter frame. This block separates the two EKFs.

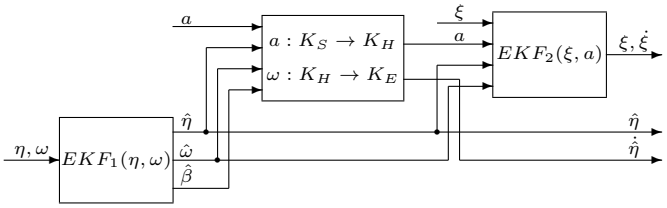


Fig. 2. The structure of the state estimator

Before the description of the state estimation, it is necessary to consider that the inertial sensor's frame is not identical to that of the helicopters. Therefore, the measured acceleration and angular velocity have to be transformed. In the following subsections, A_s (orthonormed) and p_s represent the angular and spatial displacement of the two frames.

5.1 Estimating the attitude and angular velocity of the helicopter

As indicated before, the IMU's outputs consist of three components: the real values and an additional bias and noise with zero mean. In the sensor frame, it can be written as

$$\omega_m = \omega_s = \omega_{s,0} + \omega_{s,b} + \omega_{s,n} \tag{44}$$

The real value of the angular velocity in the helicopter's frame is

$$\omega = A_s \omega_{s,0} = A_s(\omega_s - \omega_{s,b} - \omega_{s,n}) \tag{45}$$

which can be transformed to the reference frame K_E as

$$\dot{\eta} = R_r^{-1}\omega = R_r^{-1} A_s(\omega_s - \omega_{s,b} - \omega_{s,n}) \tag{46}$$

Since the bias changes slowly, it can be assumed that its time derivative is close to zero. This can be formulated by the aid of small virtual noise that affects the change of the bias's value (see [10] for another application of the technique).

$$\dot{\omega}_{s,b} = \omega_{s,b,n} \tag{47}$$

As the measurements of the positioning system do not contain offset, the third equation that can be used for state estimation along with (46) and (47) is the following:

$$\eta_m = \eta + \eta_n \tag{48}$$

If T_s denotes the sampling time, these three equations can be transformed to discrete time using the Euler's formula as

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + T_s R_{r,k} A_s(-x_{2,k} + u_k + w_{1,k}) \\ x_{2,k+1} &= x_{2,k} + T_s w_{2,k} \\ y_k &= x_{1,k} + z_k \end{aligned} \tag{49}$$

with the following notations:

$$\begin{aligned} x_1 &= \eta & x_2 &= \omega_{s,b} & x &= (x_1^T, x_2^T)^T \\ w_1 &= -\omega_{s,n} & w_2 &= \omega_{s,b,n} & w &= (w_1^T, w_2^T)^T \\ u &= \omega_m & y &= \eta_m & z &= \eta_n \end{aligned} \tag{50}$$

The equations (49) can be rewritten to the form of

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k) \\ y_k &= g(x_k, z_k) \end{aligned}$$

Assuming w and z are not correlated, the EKF algorithm can be performed by introducing the following notations:

$$\begin{aligned} R_{w,k-1} &= E[w_{k-1} w_{k-1}^T] & R_{z,k} &= E[z_k z_k^T] \\ A_{k-1} &= \frac{\partial f(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial x} & B_{w,k-1} &= \frac{\partial f(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial w} \\ C_k &= \frac{\partial g(\bar{x}_k, 0)}{\partial x} & C_{z,k} &= \frac{\partial g(\bar{x}_k, 0)}{\partial z} \end{aligned} \tag{51}$$

where $\hat{x}_k$ is the estimated value of x. The well-known steps of the extended Kalman filter algorithm are:

1. Prediction:

$$\begin{aligned} \bar{x}_k &= f(\hat{x}_{k-1}, u_{k-1}, 0) \\ M_k &= A_{k-1} \Sigma_{k-1} A_{k-1}^T + B_{w,k-1} R_{w,k-1} B_{w,k-1}^T \end{aligned} \tag{52}$$

2. Time update:

$$\begin{aligned} S_k &= C_k M_k C_k^T + C_{z,k} R_{z,k} C_{z,k}^T \\ G_k &= M_k C_k^T S_k^{-1} \\ \Sigma_k &= M_k - G_k S_k G_k^T \\ \hat{x}_k &= \bar{x}_k + G_k(y_k - g(\bar{x}_k, 0)) \end{aligned} \tag{53}$$

5.2 Estimating the position and velocity of the helicopter

The estimation method is similar to that in the previous subsection. First, it has to be considered what an accelerometer senses. Its output not only contains the three components of the acceleration, but also the effect of the gravity. Using the same notations as before, it can be formulated as

$$a_s = a_{s,0} + a_{s,b} + a_{s,n} - A_s^T R_t^T g \tag{54}$$

As the quadrotor's frame is not an inertial frame, the connection between the acceleration in K_S and K_H is the following:

$$a_{s,0} = A_s^T (a + \beta \times p_s + \omega \times (\omega \times p_s)) \tag{55}$$

From this equation, the acceleration in K_H (a) can be obtained as

$$a = A_s a_s - \beta \times p_s - \omega \times (\omega \times p_s) + R_t^T g - A_s a_{s,b} - A_s a_{s,n} \tag{56}$$

The first part of (56) can be interpreted as a transformed value of the accelerometer's output.

$$a_t = A_s a_s - \beta \times p_s - \omega \times (\omega \times p_s) + R_t^T g \tag{57}$$

This transformation is performed by the central block in Fig. 2. The equation is also an explanation why the two EKFs need to be arranged hierarchically.

Applying the differentiation rule in a moving frame ($\ddot{\xi} = \dot{v} + \dot{\eta} \times v$) once again and making the same assumptions about the bias and the positioning system yields the equations that can be used for the state estimation.

$$\begin{aligned} \dot{v} &= -\omega \times v - A_s a_{s,b} + a_t + A_s a_{s,n} \\ \dot{a}_{s,b} &= a_{s,b,n} \\ \ddot{\xi} &= R_t v + v_{\xi,n} \\ \xi_m &= \xi + \xi_n \end{aligned} \tag{58}$$

Following the same steps as previously, the discrete time equations of the system above are

$$\begin{aligned} x_{1,k+1} &= (I_3 - T_s[\omega\times]_k)x_{1,k} - T_s A_s x_{2,k} + T_s u_k + T_s A_s w_{1,k} \\ x_{2,k+1} &= x_{2,k} + T_s w_{2,k} \\ x_{3,k+1} &= x_{3,k} + T_s R_{t,k} x_{1,k} + T_s w_{3,k} \\ y_k &= x_{3,k} + T_s z_k \end{aligned} \tag{59}$$

with the notations

$$\begin{aligned} x_1 &= v & x_2 &= a_{s,b} & x_3 &= \xi \\ w_1 &= -a_{s,n} & w_2 &= a_{s,b,n} & w_3 &= v_{\xi,n} \\ u &= a_t & y &= \xi_m & z &= \xi_n \end{aligned} \tag{60}$$

The matrix $[\omega\times]$ represents the cross product and takes the form of

$$\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{61}$$

From here the EKF can easily be formed. In order to find $\beta = \dot{\omega}$ needed in a_t, numerical differentiation can be used.

6 Path tracking

6.1 The tracking algorithm

The purpose of the control design is to track a predefined trajectory with the smallest possible tracking error. In practice a navigation point must be approximated with a predefined accuracy considering the positions and orientations. Apart from the tracking error it is also important to keep the helicopter in continuous motion. In other words, the helicopter should not slow down when it reaches a waypoint but move towards the next one.

These principles can be formulated by setting up certain conditions. If one of them is satisfied, the quadrotor helicopter may advance towards the next navigation point in the algorithm.

The conditions for position tracking are

$$\begin{aligned} &\sum_{j=1}^{3} (\xi_{d_j}^{(i+1)} - \xi_j)^2 < \Delta\xi_0 \\ &\sum_{j=1}^{3} (\xi_{d_j}^{(i+1)} - \xi_j)^2 < \lambda \sum_{j=1}^{3} (\xi_{d_j}^{(i+1)} - \xi_{d_j}^{(i)})^2 \end{aligned} \tag{62}$$

where ξ is the current position, $\Delta\xi_0$ is a predefined constant distance, $\xi_{d_j}^{(i)}$ and $\xi_{d_j}^{(i+1)}$ are the coordinates of two consecutive navigation points of the trajectory (the helicopter has already stepped towards $\xi_{d_j}^{(i+1)}$). The first condition ensures that the helicopter will remain in the proximity of the navigation points, while the other one is responsible for keeping the motion continuous. $\lambda = (3/4)^2$ is a suitable compromise if the navigation points are close to each other. In the proximity of obstacles, the navigation points should be chosen denser. Obstacle avoidance is a high level motion design problem, which is not part of this article.

The tracking of the yaw angle is somewhat different since it might be important how the attitude of the helicopter behaves during flight. Therefore, the only condition is that the absolute value of the yaw angle error has to be lower than a predefined limit ($\Delta\Psi_0$).

$$\left| \Psi_d^{(i+1)} - \Psi \right| < \Delta\Psi_0 \tag{63}$$

As a further refinement, Ψ_d^{i+1} should be chosen such that the helicopter rotates in the desired direction. If Ψ_d values can be outside the interval $[0, 2\pi)$, this can also be taken into account.

6.2 Ensuring the smooth motion of the helicopter

Abrupt changes may occur between two navigation points during the manoeuvre. In order to guarantee the smooth motion, the predefined path must be refined by using a filtering procedure in order to avoid the risk of numerical problems for large tracking errors. A block with the following transfer function is able to perform this task:

$$W(s) = \frac{\gamma \, |p_s|^{n+1}}{(s + \gamma p_s)(s + p_s)^n} \tag{64}$$

with $0 < p_s$ and $0 < \gamma$. By setting $n = 5$ and $\gamma = 3$, the control inputs will still remain smooth and the transient will mostly be determined by p_s.

7 Preliminary simulations

Flying systems are complex ones, hence thorough test of their control system before the first real flight is highly suggested. Before implementing the control algorithm on the embedded target, it was tested using hardware-in-the-loop method.

The mechanical parameters of the helicopter and the BLDC motors with the rotors are based on the planned dimensions, the masses of purchased elements. These values are summarised in Table 2.

Tab. 2. The physical parameters of the helicopter and the motors

Parameter	*Value*
l	0.23 m
b	$1.759{\cdot}10^{-5}$kgm
d	$1.759{\cdot}10^{-6}$kg m^2
I_x, I_y	$1.32{\cdot}10^{-2}$kg m^2
I_z	$2.33{\cdot}10^{-2}$kg·m^2
I_r	$5.626 \cdot 10^{-5}$kg·m^2
m	1.4kg
K_t	diag([0.1 0.1 0.15])N s/m
K_r	diag([0.1 0.1 0.15])Nm s
$k_{\Omega,0}$	94.37s^{-2}
$k_{\Omega,1}$	3.02s^{-1}
$k_{\Omega,2}$	0.005
k_u	139.44V/s^{-2}

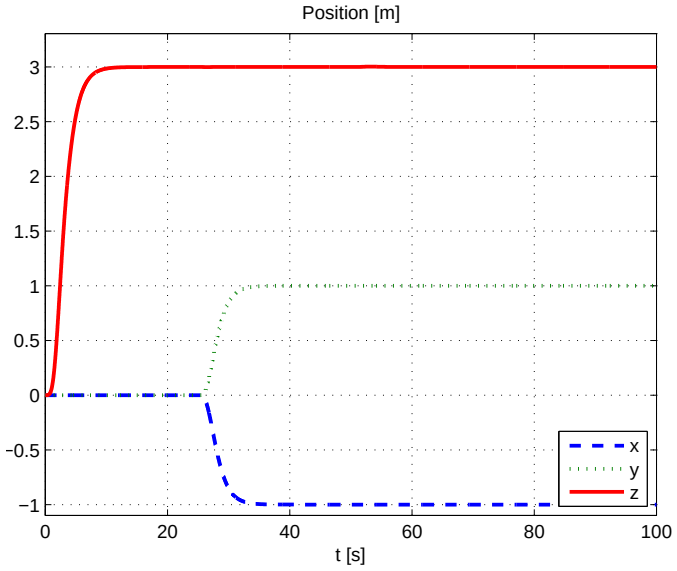
7.1 Simulations performed using simulink

The components are included in the control loop gradually from the simplest case to the most complex. The simplest simulation contained only the model of the quadrotor helicopter and the controller. No measurement noises were considered and the helicopter only had to reach a single spatial point with a defined yaw angle. The reference signals were smoothed using the same method presented in the previous section. The parameters of the controller could be tuned this way (see Table 3 for the exact values). The desired sample time of the control algorithm was set to $T_s = 10$ ms, since the maximum operating frequency of the IMU is 100 Hz. Therefore, this was the fixed step size during the simulations.

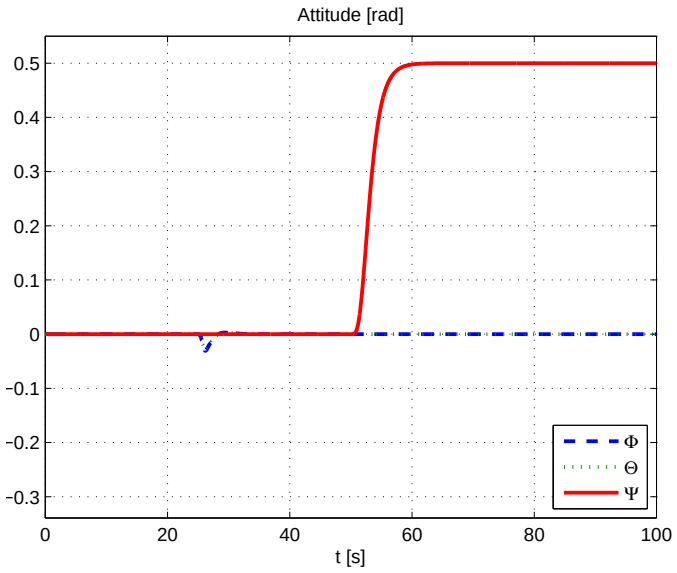
Tab. 3. The physical parameters of the helicopter

Parameter	*Value*
A_{ξ_1}	diag(2, 1.6, 1.6)
A_{ξ_2}	diag(1.2, 1.2, 1.2)
A_{η_1}	diag(12, 12, 12)
A_{η_2}	diag(8, 8, 8)
A_{m_1}	diag(0.04, 0.04, 0.04, 0.04)
A_{m_2}	diag(0.016, 0.016, 0.016, 0.016)

An example is shown below, including some signals of interest. Fig. 3 shows the position and attitude of the quadrotor helicopter, while the control inputs of the motors and the actual angular velocities of the rotors appear in Fig. 4.



(a) Position

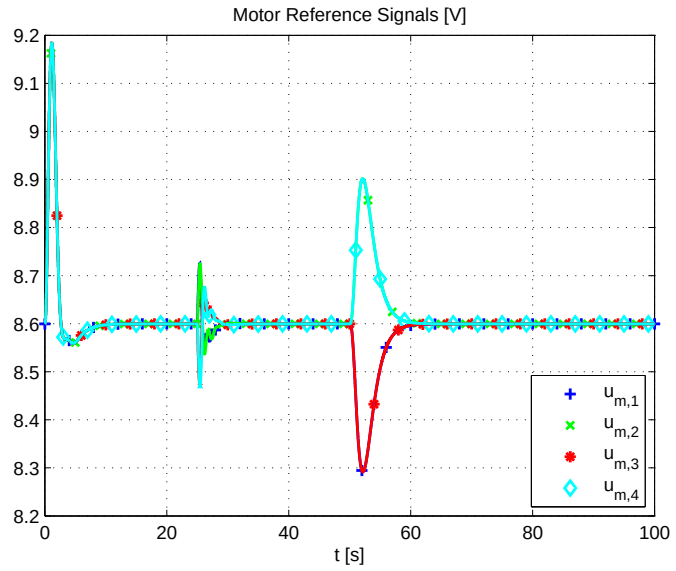


(b) Attitude

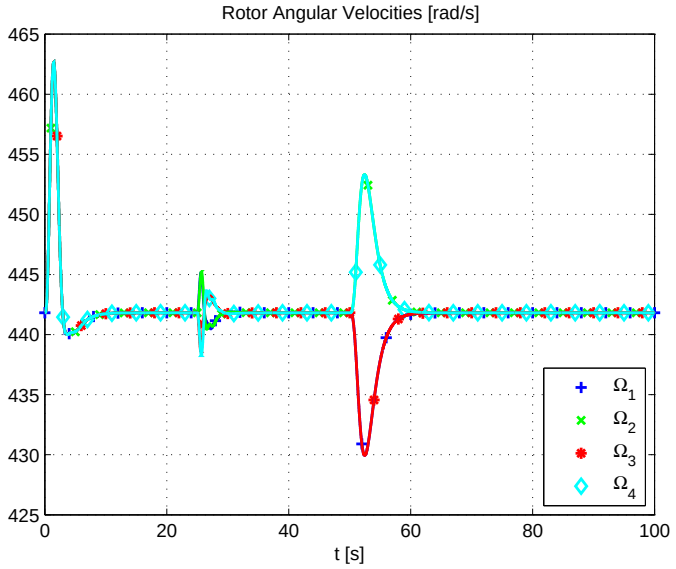
Fig. 3. The position and attitude of the helicopter during flight

Control behaviours using the simplified and more complex dynamic models were compared. According to the simulation results, the simplified position and attitude control performed similarly to the more complex model, although the latter performed better. This can be explained by the low speeds of the helicopter and the small roll and pitch angles during flight. Therefore, it is reasonable to use it instead of the more complex algorithm if hardware resources are limited. Fig. 5 shows the tracking errors of x and Ψ during the simple test (measurable states) presented above. The dotted lines correspond to the motion of the helicopter controlled by the simpler controller.

As a final result of the integration of all the components including state estimation, a tracking of a complex trajectory is presented in Fig. 6 and Fig. 7. These simulations were per-



(a) Motor Reference Signals



(b) Rotor Angular Velocities

Fig. 4. The control inputs and the angular velocities of the helicopter

formed by setting the two EKFs initial parameters as shown in Table 4 (I_3 is a 3-by-3 unit matrix). The parameters were set according to the results of the analysis of the sensor signals [11].

7.2 Real-Time Tests

Real-time tests were performed using the hardware-in-the-loop method. The tests were aided by a dSPACE DS1103 board. First the model of the helicopter, the sensor and the vision system's measurements were emulated on the board, while further experiments included the real IMU and vision system. The communication channel between the MPC555 and the DS1103 board was the CAN bus, as in the final version of the helicopter. The scheme of the tests can be seen in Fig. 8.

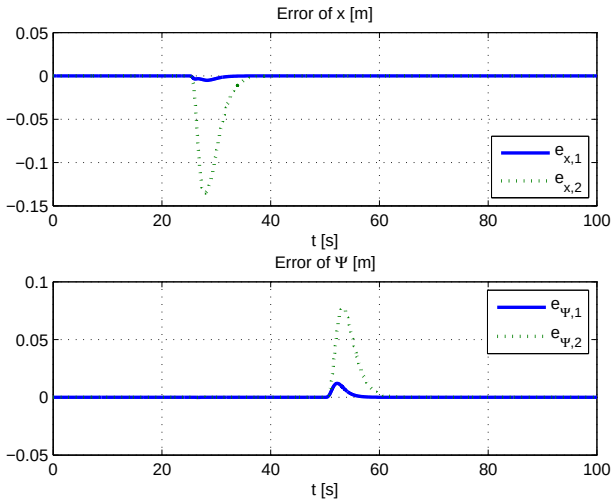


Fig. 5. Comparison of the tracking errors using simplified (index 2) and more complex (index 1) dynamic models

Tab. 4. The initial parameters of the EKFs

Parameter	*Value*
$Q_{\omega,0}$	$\mathrm{diag}(\sigma_\omega I_3, \sigma_{\omega,b} I_3)$
$R_{\omega,0}$	$\mathrm{diag}(9 \cdot 10{-}6\sigma_\eta I_3)$
$Q_{a,0}$	$\mathrm{diag}(\sigma_a I_3, \sigma_{a,b} I_3, \sigma_v I_3)$
$R_{a,0}$	$\mathrm{diag}(9 \cdot 10^{-6}\sigma_\xi I_3)$
σ_ω	$(2 \cdot 10^{-2}\mathrm{s}^{-1})^2$
$\sigma_{\omega,b}$	$(5 \cdot 10^{-4}\mathrm{s}^{-1})^2$
σ_η	$(\pi/180\mathrm{rad})^2$
σ_a	$(10^{-2}\mathrm{m/s}^2)^2$
$\sigma_{a,b}$	$(2 \cdot 10{-}3m/s)^2$
σ_v	$(2.5 \cdot 10^{-3}\mathrm{m/s})^2$
σ_ξ	$(2 \cdot 10{-}2\mathrm{m})^2$

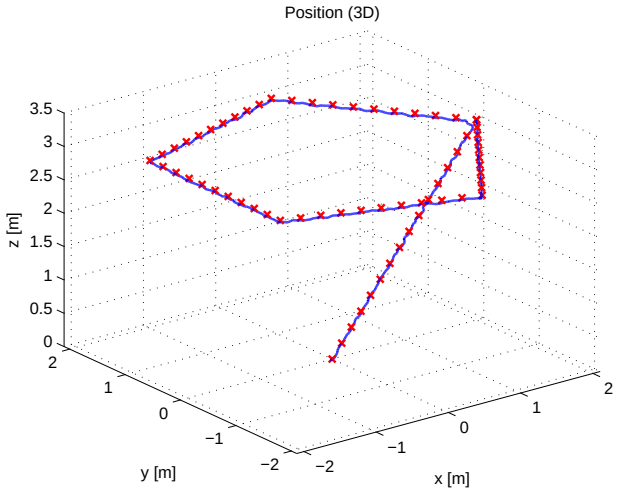
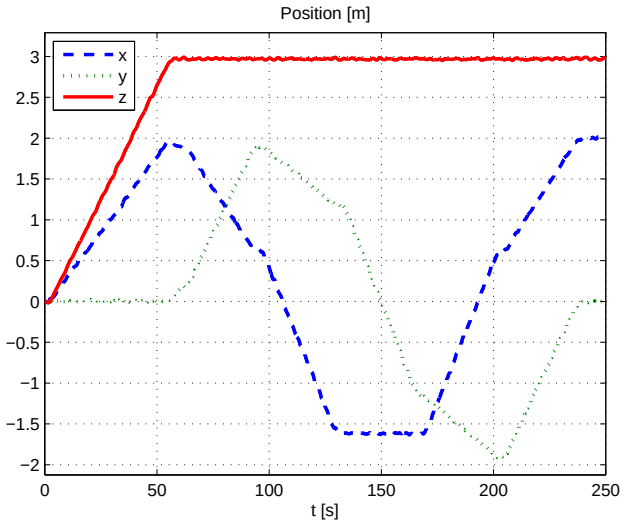


Fig. 6. A complex path tracking

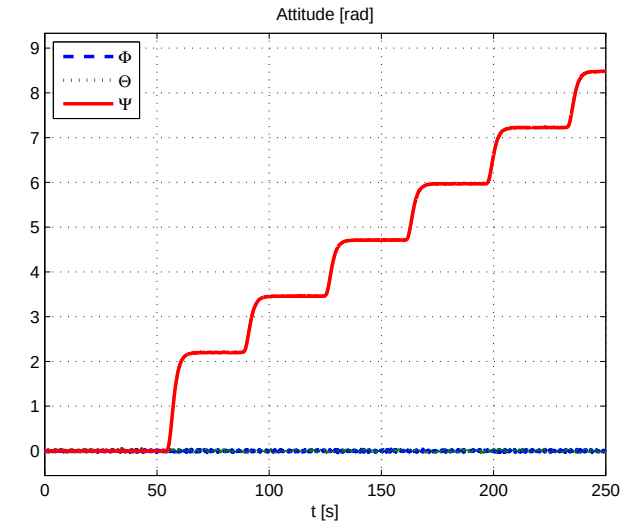
8 Implementation

8.1 Software Environment

The central unit is a Freescale MPC555 microcontroller mounted on a board produced by Phytec. The processor is equipped with a 64-bit floating point unit. The control algorithm



(a) Position



(b) Attitude

Fig. 7. The position and attitude of the helicopter during flight

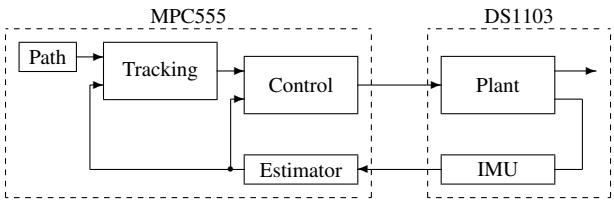


Fig. 8. Hardware-in-the-loop test setup

is designed using MATLAB/Simulink environment with Real-Time Workshop, Real-Time Workshop Embedded Coder and Embedded Target for Motorola MPC555. The generated code is compiled by MetroWerks CodeWarrior cross-development tool.

Hardware-in-the-loop tests are performed with the use of a dSPACE DS1103 board that is a powerful means of rapid prototype development. The software package includes a Simulink block library and ControlDesk, which provides a graphical user interface that controls program flow, data monitoring and collection. The collected data can easily be analysed then in MATLAB. Result of the hardware-in-the-loop tests for a complex spiral motion is illustrated in Fig. 9.

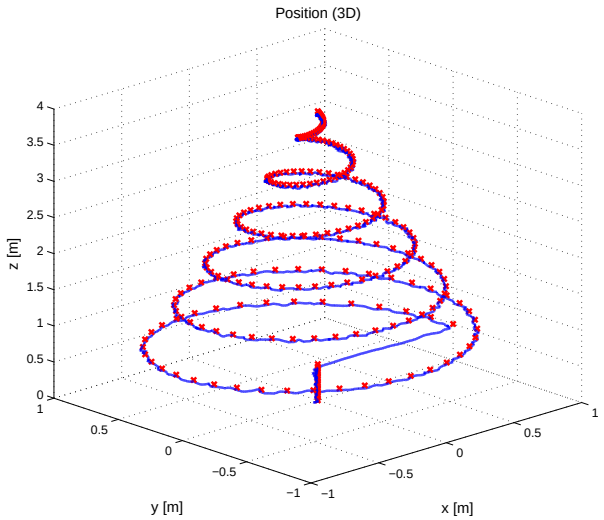


Fig. 9. Real-time test results for a complex motion

8.2 Communication

The majority of Simulink's blocks are supported MATLAB's Target Language Compiler, while the Embedded Target for Motorola MPC555 includes blocks that can be used for handling peripherals such as communication via serial or CAN interface. However, experiments show that serial communication causes a significant delay (20ms-30ms) in the signal propagation. Since CAN communication does not cause such delays, there is an extra component in the system which is responsible for converting all the serial packets into CAN packets. Wireless communication and the IMU's output is affected by this problem.

The MPC555 contains 16 buffers that can be used for transmitting or receiving CAN packets, while the number of data to be transmitted in each cycle exceeds the buffer number. Since the packet size is limited to 8 bytes, groups of the measurement data need to be transmitted in each packet. It is also crucial to maintain data integrity during the hardware-in-the-loop test, since starting the calculation of the control inputs before receiving all measurement data may make the control loop unstable. To ensure that the time delay between receiving the sensor data and starting the calculation of the new control inputs is minimal, data acquisition (checking the buffers for new packets) on the target processor is performed at a higher frequency compared to that of the control algorithm.

8.3 The sample time of the control algorithm

Preliminary calculations showed that the MPC555 is theoretically capable of performing the computation of the control inputs every 10ms. However, Execution Profiling shows that calculating the control inputs using double precision floating point numbers takes slightly more than 20ms. Using single precision

numbers does not cause significant deterioration in the calculations, however, it saves about 5ms.

8.4 Software related issues

We faced minor software problems during the development that can be avoided with little attention. If the Simulink model to be run on the dSPACE DS1103 board contains several S-functions written in C that contain global variables inside, then these variables should have different names in order to avoid unexpected behaviour during execution. The reason is that during the compilation procedure these variables are overwritten by each other.

Sampling times are also of high importance, especially when the model contains multiple sample times. It is not a good practice to set the sampling time property "inherited" of any Simulink block. For the same reason the usage of discrete time blocks that do not have sampling time property (like discrete derivative blocks) is not recommended.

9 Conclusion

In this article the theoretical foundations and the real-time realisation of the embedded control system of an indoor helicopter (UAV) were presented. The components of the controller were described in detail. A backstepping algorithm is responsible for stabilising the quadrotor helicopter. From theoretical point of view, our results differ from earlier ones in that backstepping is integrated with state estimation based on advanced sensors. The state estimator consists of two extended Kalman filters. The embedded controller was realised by using a Freescale MPC555 processor, a Crossbow MNAV100CA IMU and a marker-based vision system developed by BME IIT. Quick prototype design of the controller was performed based on MATLAB/Simulink, Real-Time Workshop and MPC555 Target Compiler. Hardware-in-the-loop real-time tests were performed using the DS1103 board of dSPACE which emulated the helicopter and the sensor system of the indoor helicopter. A CAN network served as communication channel between the components of the units of the distributed control system. Path tracking results show the effectiveness of the embedded control system under real-time conditions.

References

1 **Soumelidis A, Gáspár P, Bauer P, Regula G, Lantos B, Prohászka Z**, *Design of an Embedded Microcomputer Based Mini Quadrotor UAV*, Proceeding of the European Control Conference ECC'07 (2007), 2236-2241.

2 **Soumelidis A, Gáspár P, Regula G, Lantos B**, *Control of an Experimental Mini Quad-Rotor UAV*, Proceeding of the 16th Mediterranean Conference on Control and Automation (2008), 1252-1257.

3 **Kis L, Regula G, Lantos B**, *Design and Hardware-in-the-Loop Test of the Embedded Control System of an Indoor Quadrotor Helicopter*, Proceeding of the 6th Workshop on Intelligent Solutions in Embedded Systems WISES'08 (2008), 35-44.

4 **Kis L, Lantos B**, *Sensor-Fusion and Actuator System of a Quadrotor Helicopter*, Accepted for publication in Periodica Polytechnica El. Eng (2009).

5 **Bouabdallah S, Siegwart R**, *Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor*, Proceeding of the IEEE International Conference on Robotics and Automation (2005), 2247–2252.

6 **Bouabdallah S, Noth A, Siegwart R**, *PID vs LQ control techniques applied to an indoor micro quadrotor*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'04 **3** (2004).

7 **Coza C, Macnab C J B**, *A New Robust Adaptive-Fuzzy Control Method Applied to Quadrotor Helicopter Stabilization*, Proceeding of the Annual meeting of the North American Fuzzy Information Processing Society NAFIPS'06 (2006), 454–458.

8 **Das A, Subbarao K, Lewis F**, *Dynamic Inversion of Quadrotor with Zero-Dynamics Stabilization*, Proceeding of the 17th IEEE Conference on Control Applications (2008), 1189–1194.

9 **Madani T, Benallegue A**, *Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique*, Proceedings of the 45th IEEE Conference on Decision and Control (2006), 1515–1520.

10 **Lantos B**, *State Estimation Based on Image Processing and Inertial Sensors for Indoor Autonomous Systems*, BME IIT, 2007.

11 **Kis L, Lantos B**, *Calibration and Testing Issues of the Vision, Inertial Measurement and Control System of an Autonomous Indoor Quadrotor Helicopter*, 17th International Workshop on Robotics in Alpe-Adria-Danube region (2008).