

# Design and Hardware-in-the-Loop Test of the Embedded Control System of an Indoor Quadrotor Helicopter

Laszlo Kis	Gergely Regula	Bela Lantos
<i>Budapest University of Technology and Economics Dept. Control Engineering and Information Technology kislac761@gmail.com</i>	<i>Budapest University of Technology and Economics Dept. Control Engineering and Information Technology majom3rg@yahoo.com</i>	<i>Budapest University of Technology and Economics Dept. Control Engineering and Information Technology lantos@iit.bme.hu</i>

## Abstract

*The paper deals with the quick prototype design and hardware-in-the-loop real-time test of the embedded control system of an indoor miniature quadrotor helicopter (UAV) before first flight. The control law is of backstepping type, the sensory system consists of a marker-based vision system outside the helicopter in radio connection with the embedded controller and a 3D inertial measurement unit (IMU) on the helicopter. Extended Kalman filters solve the state estimation problem. Brushless DC motors serve as actuators. For quick prototype design of the embedded controller MATLAB, Simulink, Real-Time Workshop and the MPC555 Target Compiler were used. Because of the critical character of the control of flying systems the whole embedded control system was hardware-in-the-loop tested by using DS1103 of dSPACE for real-time emulation of the indoor helicopter. During test the helicopter and the sensors were mapped to DS1103 and the test was performed in real-time using the helicopter and sensor models. Communication during the test was running using CAN bus protocol between the Freescale MPC555 processor of the embedded controller and the emulation of the helicopter and the sensors on DS1103.*

*Key-Words: Indoor UAV, Embedded controller, Real-time realization, Quick prototype design, Hardware-in-the-loop testing, Marker-based vision and IMU sensors, Extended Kalman filters, Backstepping control.*

## 1. Introduction

Developing an unmanned mini quadrotor helicopter which is able to execute autonomously a mission, for example performing a series of measurements in important positions or completing a surveillance task above a limited territory is one of the tasks for near future. Beside military applications a lot of civil applications are also important, hence research in this field has gained increasing significance. While outdoor helicopters can easily use differential GPS for position and orientation (attitude) measurements, indoor helicopter sensors are usually limited to vision and inertial sensors [1] making the problem more complex.

The paper deals with the indoor case. The structure of the paper is as follows. In section 2 the helicopter and actuator models are derived. Section 3 presents the backstepping control law implemented in the embedded controller. Section 4 deals with the state estimation problem using two stage extended Kalman filters. Section 5 discusses the implementation and the hardware-in-the-loop real-time test of the embedded

controller. Section 6 deals with the integration of the dedicated actuator and sensor processors completing the embedded controller. Section 7 contains the conclusions.

## 2. Helicopter dynamic model

Multicomponent robotic systems can be well described using coordinate systems (frames). Denote  $K_W = K_{H0}$  the helicopter frame in the stop state (world or base inertial frame),  $K_H$  the moving frame of the helicopter center of gravity and  $K_S$  the sensor frame of IMU fixed to the helicopter. For simplicity the vision frame is assumed to be equal to  $K_W$ , hence the vision system measures the position and orientation of  $K_H$  relative to  $K_W$ . The relative position and orientation between frames can be described by a homogeneous transformation  $T$  whose orientation and positions parts are  $A$  and  $p$ , respectively. The graph of the used frames is shown in Fig. 1.

$$K_W = K_{H0} \xrightarrow{T_{W,H}} K_H \xrightarrow{T_{H,S}} K_S$$

**Figure1. The graph of the important frames**

In order to develop the kinematic and dynamic models the results of flight control systems [6] and mini-flying machines [7] can be applied. The position and the orientation of the helicopter with respect to the base frame are denoted by  $\xi = (x, y, z)^T$  and  $\eta = (\Phi, \Theta, \Psi)^T$  respectively where  $\Phi, \Theta, \Psi$  are the Euler angles and the  $A(\Phi, \Theta, \Psi)$  orientation matrix is

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

The angular velocity can be written in the form  $\omega = \Gamma \dot{\eta}$  where  $\dot{\omega} = \Gamma \ddot{\eta} + \dot{\Gamma} \dot{\eta}$  and

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

The helicopter has 4 actuators, each actuator exerts a lift force proportional with the square of the angular velocity of the rotor ( $f_i = b\Omega_i^2$ ). Each actuator is a brushless DC motor with own controller whose reference signal can be programmed in  $\Omega_i$  [2]. The resulting lift force ( $u$ ) and driving torque ( $\tau$ ) are defined by

$$u = f_1 + f_2 + f_3 + f_4 = b \sum_{i=1}^4 \Omega_i^2$$

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

where  $l, b, d$  are helicopter and motor constants. The gyroscopic effect can be modeled as

$$\tau_G = -I_r(\omega \times k)(\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3) =$$

$$=: -I_r(\omega \times k)\Omega$$

where  $I_r$  is the rotor inertia and  $k$  is the third unit vector. Denote  $I$  the inertia matrix of the helicopter then the differential equation of the helicopter is

$$m\ddot{\xi} = AF_{ext} + F_g$$

$$I\ddot{\omega} + \omega \times (I\omega) = \tau_{ext}$$

where  $F_{ext}$  and  $\tau_{ext}$  are the external force and torque, respectively, in  $K_H$  and  $F_g = (0, 0, -mg)^T$  is the gravity force in  $K_W$ . We assume in the sequel that

$I = \text{diag}(I_x, I_y, I_z)$  and  $\Phi, \Theta \approx 0$ ,  $\Gamma \approx I_3$  (unite matrix),  $\omega \approx (\dot{\Phi}, \dot{\Theta}, \dot{\Psi})^T$ . Hence the differential equation of the helicopter can be written in the simplified form

$$\begin{aligned}\ddot{x} &= (C_\phi S_\theta C_\psi + S_\phi S_\psi) \frac{u}{m} =: u_x \frac{u}{m} & \ddot{\Phi} &= \frac{I_y - I_z}{I_y} \dot{\Theta} \dot{\Psi} - \frac{I_r}{I_x} \dot{\Theta} \dot{\Omega} + \frac{1}{I_x} \tau_x \\ \ddot{y} &= (C_\phi S_\theta S_\psi - S_\phi C_\psi) \frac{u}{m} =: u_y \frac{u}{m} & \ddot{\Theta} &= \frac{I_z - I_x}{I_y} \dot{\Phi} \dot{\Psi} + \frac{I_r}{I_y} \dot{\Phi} \dot{\Omega} + \frac{1}{I_y} \tau_y \\ \ddot{z} &= -g + C_\phi C_\theta \frac{u}{m} & \ddot{\Psi} &= \frac{I_x - I_y}{I_z} \dot{\Phi} \dot{\Theta} + \frac{1}{I_z} \tau_z\end{aligned}$$

From the differential equations the state equations can be easily written down if the state is chosen as  $x := (x, y, z, \dot{x}, \dot{y}, \dot{z}, \Phi, \Theta, \Psi, \dot{\Phi}, \dot{\Theta}, \dot{\Psi})^T$ .

### 3. Backstepping control law

There are different approaches of backstepping control of helicopters [4], [5]. We choose a simple form based on the following principle.

Let a SISO subsystem and the variables  $q_1$  and  $q_2$  be defined by

$$\begin{aligned}\ddot{x} &= f(x, \dot{x}) + bu \\ q_1 &:= x_d - x \\ q_2 &:= \dot{x} - \dot{x}_d - \alpha_1 q_1 = \dot{q}_1 - \alpha_1 q_1 \Rightarrow \dot{q}_1 = -\alpha_1 q_1 - q_2 \\ \dot{q}_2 &= \ddot{x} - \ddot{x}_d - \alpha_2 \dot{q}_1 =: q_1 - \alpha_2 q_2\end{aligned}$$

where  $x_d$  is the reference signal (set point) and  $\alpha_1, \alpha_2 > 0$ , then

$$\ddot{x} = q_1 - \alpha_2 q_2 - \alpha_1(\alpha_1 q_1 + q_2) + \ddot{x}_d = f(x, \dot{x}) + bu$$

Let assume  $\ddot{x}_d \approx 0$  and choose the control as

$$u = \frac{1}{b} \{q_1 - \alpha_2 q_2 - \alpha_1(\alpha_1 q_1 + q_2) - f(x, \dot{x})\}$$

The system is asymptotically stable according to the Lyapunov function ( $V$ ) and its derivative ( $\dot{V}$ ):

$$V(q_1, q_2) = \frac{1}{2}(q_1^2 + q_2^2) \quad \dot{V} = -\alpha_1 q_1^2 - \alpha_2 q_2^2 < 0$$

The method can separately be applied for every subsystem of the helicopter resulting in the composite backstepping controllers

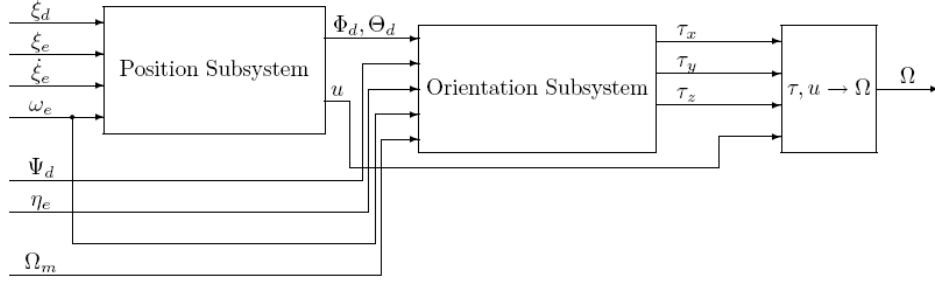
$$\begin{aligned}\tau_x &= \frac{1}{b_1} \{q_1 - \alpha_2 q_2 - \alpha_1(\alpha_1 q_1 + q_2) - a_1 \dot{\Theta} \dot{\Psi} - a_2 \dot{\Theta} \dot{\Omega}\} & u &= \frac{m}{C_\phi C_\theta} \{q_7 - \alpha_8 q_8 - \alpha_7(\alpha_7 q_7 + q_8) + g\} \\ \tau_y &= \frac{1}{b_2} \{q_3 - \alpha_4 q_4 - \alpha_3(\alpha_3 q_3 + q_4) - a_3 \dot{\Psi} \dot{\Phi} - a_4 \dot{\Phi} \dot{\Omega}\} & u_x &= \frac{m}{u} \{q_9 - \alpha_{10} q_{10} - \alpha_9(\alpha_9 q_9 + q_{10})\} \\ \tau_z &= \frac{1}{b_3} \{q_5 - \alpha_6 q_6 - \alpha_5(\alpha_5 q_5 + q_6) - a_5 \dot{\Phi} \dot{\Theta}\} & u_y &= \frac{m}{u} \{q_{11} - \alpha_{12} q_{12} - \alpha_{11}(\alpha_{11} q_{11} + q_{12})\}\end{aligned}$$

$$S_\phi = S_\psi u_x - C_\psi u_y, \quad S_\theta = \frac{C_\psi u_x + S_\psi u_y}{C_\phi} \Rightarrow \Phi_d, \Theta_d \text{ for Orientation Subsystem}$$

For simplification some new notations were introduced here, for example  $a_1 = (I_y - I_z)/I_x$ ,  $b_1 = l/I_x$ ,  $u_x = C_\phi S_\theta C_\psi + S_\phi S_\psi$ ,  $\ddot{x} = u_x u/m$  etc. From  $u, \tau_x, \tau_y, \tau_z$  and the force and torque actuator equations  $\Omega_i, i=1, \dots, 4$  can be easily determined.

The backstepping controller has the form shown in Fig. 2. In the block scheme the indices  $d$ ,  $m$  and  $e$  refer to the desired, measured and estimated values, respectively.

The controller was implemented on Freescale MPC555 processor having floating point instructions.



**Figure 2. Controller structure based on backstepping method**

#### 4. State estimation

The state estimation is realized using two stage extended Kalman filters (EKF). For state estimation the measurements of the IMU and those of the marker-based vision system are used.

The IMU is of type Crossbow MNAV100CA producing 3D acceleration and angular velocity measurements. Because of technical constraints the IMU is shifted by  $p_s$  with respect to the center of gravity of the helicopter. Between  $K_H$  and the sensor frame  $K_S$  there is a homogeneous transformation  $T_{H,S}$  whose orientation and position parts are  $A_S$  and  $p_s$ , respectively. The maximal allowed sampling frequency of IMU is 100Hz (sampling time 0.01s).

The marker-based vision system is based on the motion stereo principle and is running on a host computer supervising the whole system [3]. The vision system is working with 10 frame/sec and is in connection with the embedded controller on the helicopter where the coupling is managed by an extra processor converting radio signals into CAN protocol. Position and orientation (in form of Euler angles) are expressed in  $K_W$ , the measurement frequency is 10Hz (sampling time 0.1s).

##### 4.1. Estimation of the orientation and the angular velocity of the helicopter

Collect the angular velocity components measured by the IMU in the vector  $\rho_S \in K_S$  which beside the correct value  $\rho_{S,0}$  also contains the unknown bias and a Gaussian noise and transform them into the helicopter frame  $K_H$  :

$$\rho_S = \rho_{S,0} + \rho_{S,b} + \rho_{S,n}$$

$$A_S \rho_S = A_S \rho_{S,0} + A_S \rho_{S,b} + A_S \rho_{S,n}$$

$$\rho := A_S \rho_{S,0} = A_S \rho_S - A_S \rho_{S,b} + A_S \rho_{S,n}$$

(the sign of the noise is not important). Here  $\rho \in K_H$  is the angular velocity vector  $\rho = (P, Q, R)^T$  from which the derivative of the Euler angles can be determined by using

$$\Gamma^{-1} =: F(\Phi, \Theta) = \begin{bmatrix} 1 & T_\Theta S_\Phi & T_\Theta C_\Phi \\ 0 & C_\Phi & -S_\Phi \\ 0 & S_\Phi / C_\Theta & C_\Phi / C_\Theta \end{bmatrix}$$

$$\begin{pmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{pmatrix} = F(\Phi, \Theta) \rho = -F A_S \rho_{S,b} + F A_S \rho_S + F A_S \rho_{S,n}$$

$$\dot{\rho}_{S,b} = \rho_{S,b,n}$$

$$\begin{pmatrix} \Phi \\ \Theta \\ \Psi \end{pmatrix}_m = \begin{pmatrix} \Phi \\ \Theta \\ \Psi \end{pmatrix}_n + \begin{pmatrix} \Phi \\ \Theta \\ \Psi \end{pmatrix}_n$$

Here  $\rho_S$  is measured by the IMU while  $(\Phi, \Theta, \Psi)_m^T$  by the marker-based vision system.

Denote  $T$  the sampling time of IMU then the continuous time model can be converted to the discrete time one using Euler approximation:

$$x_1 = (\Phi, \Theta, \Psi)^T, \quad x_2 = \rho_{S,b}, \quad x = (x_1^T, x_2^T)^T,$$

$$u = \rho_S, \quad w_1 = \rho_{S,n}, \quad w_2 = \rho_{S,b,n}, \quad w = (w_1^T, w_2^T)^T$$

$$x_{1,k+1} = x_{1,k} - TF_k A_S x_{2,k} + TF_k A_S u_k + TF_k A_S w_{1,k}$$

$$x_{2,k+1} = x_{2,k} + T w_{2,k}$$

$$y_k = x_{1,k} + z_k$$

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

$$y_k = g(x_k, z_k)$$

Assume that  $w$  and  $z$  are uncorrelated and introduce the notations

$$R_{w,k} = E[w_k w_k^T], \quad R_{z,k} = E[z_k z_k^T]$$

$$A_k = \frac{\partial f(\hat{x}_k, u_k, 0)}{\partial x}, \quad B_{w,k} = \frac{\partial f(\hat{x}_k, u_k, 0)}{\partial w}$$

$$C_k = \frac{\partial g(\hat{x}_k, 0)}{\partial x}, \quad C_{z,k} = \frac{\partial g(\hat{x}_k, 0)}{\partial z}$$

The EKF performs the following computations:

1. Prediction:

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

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

2. Time update:

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

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

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

$$\Sigma_k = M_k - G_k S_k G_k^T$$

Since the sampling time is 10ms according to IMU while the image processing produce orientation data in every 100ms hence if orientation data are not arrived then in the time update  $\hat{x}_k = \bar{x}_k + 0$  is performed.

#### 4.2. Estimation of the velocity and the position of the helicopter

The 3D acceleration measurements of IMU can be modeled by

$$a_S = a_{S,0} + a_{S,b} + a_{S,n} - A_S^{-1} A(\Phi, \Theta, \Psi)^{-1} g$$

$$a_{S,0} = A_S^{-1} (a + \varepsilon \times p_S + \omega \times (\omega \times p_S))$$

$$a_m := A_S a_S - \varepsilon \times p_S - \omega \times (\omega \times p_S) + A(\Phi, \Theta, \Psi)^{-1} g$$

where  $\varepsilon$  is the angular acceleration (not measured).

Denote  $v \in K_H$  the velocity of the helicopter then using the differentiation rule in moving frames it yields

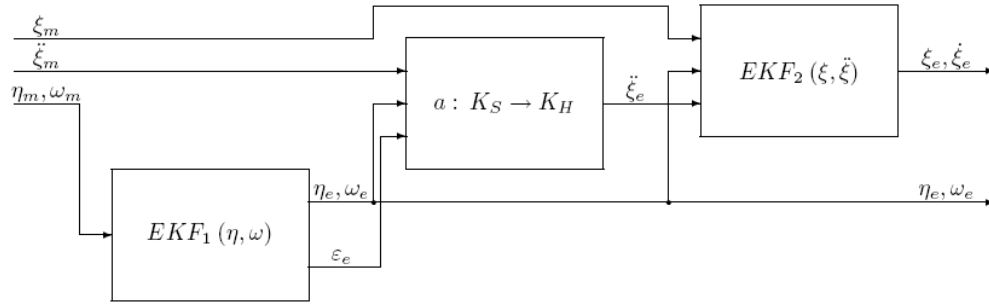
$$\begin{aligned}\dot{v} &= -\omega \times v - A_S a_{S,b} + a_m + A_S a_{S,n} \\ \dot{a}_{S,b} &= a_{S,b,n} \\ \dot{p} &= A(\Phi, \Theta, \Psi)v + v_{p,n} \\ p_m &= p + p_n\end{aligned}$$

The discrete time equivalent of this model can easily be derived and results in

$$\begin{aligned}x &= (v^T, a_{S,b}^T, p^T)^T, u = a_m, w = (a_{S,n}^T, a_{S,b,n}^T, v_{p,n}^T)^T \\ y &= p_m, z = p_n \\ x_{k+1} &= f(x_k, u_k, w_k) \\ y_k &= g(x_k, z_k)\end{aligned}$$

From here the EKF can easily be formed. In order to find  $\varepsilon$  needed in  $a_m$  numerical differentiation was used, while IMU gives measurements of  $a_s$  and the vision system those of  $p_m$ .

The structure of the two stage extended Kalman filter is shown in Fig. 3. In the block scheme the indices  $m$  and  $e$  refer to measured and estimated values, respectively.



**Figure 3. The structure of the state estimation**

## 5. Real-time realization and test of the embedded controller based on MPC555 processor

An embedded system is a special-purpose computer system designed to perform some dedicated functions, often with real-time constraints. It is usually embedded as part of a complete device, in our case the indoor UAV, including hardware and mechanical parts. Embedded controller of a quadrotor helicopter integrates microcontrollers, actuators, vision and inertial sensors. Earlier small indoor UAVs were realized using 8-bit microcontrollers, limited degree of freedom gyroscope and accelerometer (MEMS), serial servo controllers, R/C transmitter and receiver and lightweight rechargeable battery allowing approximately 15 minutes flight duration [8].

Our main goal is to increase the intelligence and control properties by using an advanced microprocessor (Freescale MPC555) allowing quick floating point computations and high speed data transfer amongst system components.

The development of the control algorithm has been carried out in MATLAB Simulink environment, since code generation to the target processor can be done in a convenient way with its additional components (Real-Time Workshop and Target Language Compiler).

In order to validate the control algorithm, i.e. check if it can operate at the desired frequency, hardware-in-the-loop tests were performed that involved the emulation of the quadrotor helicopter's model and the flow of the sensory data.

### 5.1. Software realization of the embedded controller

Since the backstepping based algorithm realizes a point-to-point control, the rotorcraft is able to follow a path defined by a series of points in the Cartesian coordinate system using an additional path tracking algorithm. The tracking algorithm is responsible for providing the control algorithm with the actual reference signals (3D position coordinates and yaw angle) and for keeping the helicopter in continuous motion.

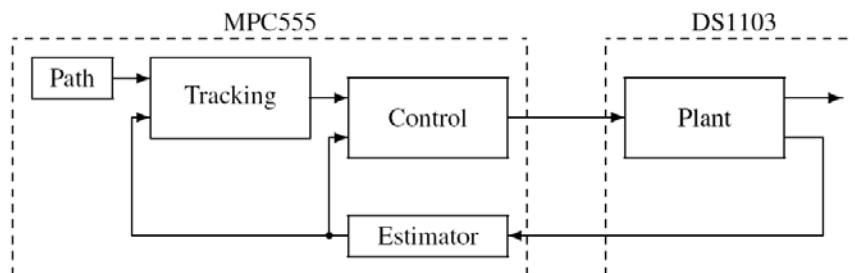
Sudden changes in the reference signals may cause numerical instability, therefore the reference signals are smoothed by third order filters, which ensure that the second derivatives of the state variables remain smooth.

Due to the high complexity of the control algorithm the planned 10 ms sampling time couldn't be realized, 30 ms was used during the tests instead.

### 5.2. Controller and test system

Hardware-in-the loop tests were aided by a dSPACE DS1103 board which is specially designed for rapid control prototyping. The DS1103 is fully programmable from the Simulink block diagram environment and all the signals can be monitored during the real-time tests and saved to workspace for further analysis.

The helicopter's model described in section 2 was implemented on the test board, as well as all the sensory measurements and the outputs of the image processing algorithm. Communication between the DS1103 board and the MPC555 was carried out via CAN interface, described in details in section 6. The scheme of the distributed system is shown in Fig. 4.



**Figure 4. The simplified scheme of the distributed system**

Testing the control in a distributed system brings up several issues that must be addressed. The operation of the two units has to be synchronised and data consistency must be maintained.

A start mechanism is an essential part both in the MPC555 and in the DS1103. It is obvious that inadequate rotor angular velocity values would cause unwanted behavior in the helicopter motion (diving), while the lack of measurement data would cause extended Kalman filters to work improperly, decreasing the quality of the control algorithm. The solution is to start the control algorithm at the MPC555 side when it receives the first adequate, i.e. nonzero measurement data from the DS1103, while at the DS1103 a default control input is applied to the helicopter's model until the first calculated control input arrives from the MPC555.

The other issue of high importance is the data consistency during the calculation of the control inputs. Simulation results show that starting the calculation of the control inputs before receiving all the measurement data at each step can result in instability of the closed control loop. To avoid malfunctioning, data sent from the dSPACE unit are timestamped and the MPC555 starts every cycle only after receiving all the data. This method also has an additional beneficial effect, i.e. synchronizes the two separate units' operation.

### 5.3. Test results

Several tests have been performed to investigate the capabilities of the proposed control algorithm for paths of different kind including maneuvers in horizontal plane consisting of straight lines and special spiral-formed paths.

Fig. 5. shows the results of a test of the latter type of path. The crosses show the waypoint coordinates while the helicopter's trajectory is signed with continuous line.

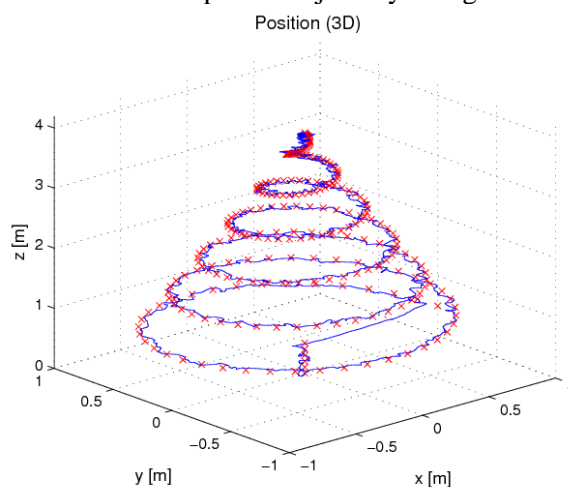


Figure 5. Test results during spiral-formed path

## 6. Integration of actuator and sensor system using dedicated processors

The difference between the hardware-in-the-loop test and the real-time operation is that data flows from the IMU and the vision processing system are not synchronized. To decrease this difference accurate and reliable communication is needed.

Communication among parts of the helicopter is realized on CAN bus. The IMU and the RF unit has no CAN interface, therefore dedicated processors are needed to convert serial data to CAN packages. Four brushless DC motors are controlled by separated microcontrollers. These units receive actuator signals and produce information about the actual revolution. On the Crossbow IMU an offline calibration is used to remove permanent errors and another calibration before start to decrease varying errors. Marker-based vision system is running on a host PC, which communicates with the helicopter via radio connection.

### 6.1. CAN communication

There are six different critical signals in the communication: actuator, measured revolution, acceleration and angular velocity from IMU, position and orientation from the marker-based vision system.



Other signals are also propagated on the CAN bus, like starting or landing, path information, battery information and data logging. But the propagation time of these signals are not as critical as the first six ones.

$\Omega_i, i=1, \dots, 4$  can be put into one CAN package. This is received by the four motor control units in the same time. It solves the synchronization problem of the actuator controls. This package got the highest priority, the worst case propagation time (WCPT) is 0.5ms beside 500kbit/s communication speed. Actual value of motor revolution is measured by the motor controllers, propagated via four CAN packages. These are periodic signals with a known time period, hence WCPT can be calculated (1.5ms). Acceleration and angular velocity values need two CAN packages. Its WCPT is 2ms, which is under the sample time of the state estimation. Position and orientation data from the host PC have an unknown delay because of RF channel, hence WCPT cannot be calculated, but these packages are transmitted with the highest possible priority.

## **6.2. Actuator control**

We use four sensorless brushless DC motors as actuators. Each one has its own motor control unit, described in [2], realized on Atmel AT90CAN128 microcontroller. The control algorithm uses a PI method. Each module is able to measure actual revolution with a 10ms sampling time.

## **6.3. IMU**

Offline calibration has two steps. The first step can remove bias and gain error from acceleration value and it is able to solve the problem that the axes of the accelerometers are not perpendicular. The second step eliminates the bias error and gravity effect from measured angular velocity. This order is needed to eliminate gravity error from angular velocity sensor. Measurements for calibration are taken in stationary position with different orientation. At least 10 values are needed for calibration.

Online calibration is executed before start. This algorithm supposes that the starting orientation of the helicopter is known from the marker based vision system. It decreases the starting biases from the acceleration and angular velocity values, so the initial value of these biases in the state estimation algorithm can start from zero.

Before the calibration the 1G gravity vectors were measured between 0.87G and 1.12G in different orientations. 18 points were used for calibration and the measured values were between 0.98G and 1.01G after calibration. Before angular rate calibration the error in stationary position was  $\pm 1.5$ degree/s in different orientations. After calibration this error decreased to  $\pm 0.1$ degree/s.

The IMU has RS-232 interface for communication, an Atmel AT90CAN128 microcontroller converts signals to CAN packages.

## **6.4. Vision system and radio connection**

We use a marker-based vision system, described in [3], to calculate position and orientation data. Algorithm is running in MATLAB/Simulink environment over Windows platform on a host communicating with the task of RF transmitter. Calculation speed in usual situations is more than 10 frame/s.

The Simulink code communicates with an other task on the host, written in C. This task sends the position and orientation data to the RF unit and high level commands for the control algorithm.

The RF unit is a MaxStream XBee module, working within the ISM 2.4Ghz frequency band, providing a transparent serial communication between PC and the helicopter. Its serial interface is converted to CAN by an Atmel AT90CAN128 microcontroller.

## 7. Conclusion

In this paper the theoretical foundations and the real-time realization of the embedded control system of an indoor helicopter (UAV) were presented. The control law is of backstepping character, the state estimation is based on two stage extended Kalman filter. The embedded controller was realized by using Freescale MPC555 processor, Crossbow MNAV100CA as IMU and own developed marker-based vision system. Quick prototype design of the controller was performed based on MATLAB/Simulink, Real-Time Workshop and MPC555 Target Compiler. The hardware-in-the-loop real-time test was presented using the DS1103 board of dSPACE which emulated the helicopter and the sensory system of the indoor helicopter during the test. Communication happened using CAN protocol among the system components. Composite path tracking results show the effectiveness of the embedded control system under real-time conditions.

## 8. Acknowledgement

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

## 9. References

- [1] A. Soumelidis, P. Gaspar, P. Bauer, B. Lantos and Z. Prohaszka, "Design of an Embedded Microcomputer Based Mini Quadrotor UAV", *European Control Conference*, Kos, Greece, 2007.
- [2] Cs. Molnar, "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.
- [3] Sz. Matusik, "Real-Time Position and Orientation Determination Module for the Indoor Work of a Quadrotor Helicopter", *MSc Thesis*, Budapest University of Technology and Economics, Department of Control Engineering and Information Technology, 2007.
- [4] S. Bouabdallah and R. Siegwart, "Backstepping and Sliding-Mode Techniques Applied to an Indoor Micro Quadrotor", *Proc. Int. Conf. Robotics and Automation*, Barcelona, Spain, pp. 2247-2252, 2005
- [5] T. Madani and A. Benallegue, "Control of Quadrotor Mini-Helicopter via Full State Backstepping Technique", *Proc. IEEE Conference on Decision and Control*, pp. 1515-1520, 2006.
- [6] B.L. Stevens and F.L. Lewis, "Aircraft Control and Simulation", John Wiley & Sons, Inc., New York, 1992.
- [7] P. Castillo, R. Lozano and A.E. Dzul, "Modelling and Control of Mini-Flying Machines", Springer-Verlag, London, 2005.
- [8] S.D. Hanford, L.N. Long and J.F. Horn, "A Small Semi-Autonomous Rotary-Wing Unmanned Air Vehicle (UAV)", *AIAA InfoTech&Aerospace Conference*, Washington D.C., 2005, AIAA Paper NO. 2005-7077.