In flight collision avoidance for a Mini-UAV robot based on onboard sensors

Marcelo Becker
ABCM Senior Member
Mechatronics Lab. – EESC – USP
e-mail: becker@sc.usp.br
Rafael Coronel B. Sampaio
Mechatronics Lab. – EESC – USP
e-mail: rafaelc@sc.usp.br
Samir Bouabdallah
ASL – IRIS - ETHZ
Vincent de Perrot
STI – I2S – EPFL
Roland Siegwart
ASL – IRIS - ETHZ

The major goal of this research was the development and implementation of an active control system to avoid collisions during the flight for a mini-quadrotor helicopter. However, the design aspects must be seriously studied in order to overcome hardware limitations and achieve control simplification. The controllers of an UAV (Unmanned Aerial Vehicle) robot deal with highly unstable dynamics and strong axes coupling. Furthermore, any additional onboard sensor increases the robot total weight and therefore decreases its operating time. It is necessary to find a balance between onboard electronics and robot operating time. This paper focuses not only on the development and implementation of a collision avoidance controller for a mini-VTOL robot using onboard ultrasound (US) sensors for detecting obstacles and for controlling its altitude, but also on the mathematical model that was essential for the controller developing phases as well. In order to facilitate the controller implementation, we developed a simulation tool in MatLab/Simulink using an accurate dynamic model of OS4 (our mini-VTOL robot). This tool allowed us to simulate and improve the OS4 controllers in different modeled environments and test different approaches. After that, the controllers were embedded in the real robot and the results were encouraging. At present time, this is an ongoing research being carried out with a new mini-VTOL flying robot configuration: a coaxial mini-helicopter.

***Keywords**: VTOL, Obstacle Avoidance, Robotics, Mini-flying robots.*

Introduction

The potential use of flying robots on military and civil applications and the challenges behind their development are attracting the scientific and the industrial community. Thanks to this, unmanned aerial vehicles (UAV) became considerably popular. Today they are being used mainly for surveillance and inspection tasks. Nevertheless recent advances in low-power embedded processors, miniature sensors and control theory are opening new horizons in terms of miniaturization and fields of use. Miniature Flying Robots (MFR) that use the Vertical Taking-Off and Landing concept (VTOL) have many advantages when compared to other mobile robots in complex or cluttered environments. Mini-VTOL can also serve in search-and-rescue missions after earth-quakes, explosions, etc. An aerial robot able to fly in narrow space and collapsed buildings can, for example, search victims of accidents or natural disasters without risking human lives.

Recently, many works in the literature highlighted the mini-VTOL mechanical design and the development of control strategies for maneuvers such as taking-off, hovering, and landing. Kroo et al. (2000) presented interesting results in centimeter-scale quadrotor design and analysis. Other interesting investigations were the ground effect study using a free-vortex wake model (Griffiths and Leishman, 2002) and the flapping concept presented in Deng et al. (2003). Hoffmann et al. (2004) outlined the development of a miniature autonomous flight control system and the creation of a multi-vehicle platform for experimentation and validation of multi-agent control algorithms. One of recent results from EPSON (2008) is a 13.6cm micro-helicopter that is able to hover 3 minutes. It is remotely operated via Bluetooth link. The Swiss Federal Institutes of Technology, EPFL and ETHZ, are also participating with several projects to this scientific challenge (respectively, Aero-EPFL, 2008 and UAV-ETHZ, 2008). At EPFL, the muFly project aims the design of a full autonomous helicopter, which main goal is to achieve a device that can be compared to a bird in size and weight. At ETHZ, the sFly (sFly, 2010)project consists in an effort to make possible that small helicopters can safely fly over metropolitan areas to assist humans in several tasks like surveillance and rescue.

At the Autonomous Systems Lab (ASL) we worked on a quadrotor mini-helicopter named OS4 until 2008. From 2003 to 2005 many goals concerning the mechanical design and control field were achieved (Bouabdallah et al., 2004-a, 2004-b, 2007, Bouabdallah and Siegwart, 2005-a, 2005-b, and 2007 and Bouabdallah, 2007). Numerous approaches have already been developed in the field of obstacle avoidance in mobile robotics. However, most of these methods are not applicable to mini-VTOLs because of their typical low available payload, embedded processing power, and auto-localization issues. Due to these reasons there is a lack of publications about obstacle avoidance procedures for mini-VTOLs. Maybe an exception is the work of Roberts et al. (2007), which describes the experimental results with the Quadrotor, a low weight flying robot endowed with rate gyroscopes, accelerometers, ultrasonic and infrared sensors, a high speed motor controller and a flight computer. The device presents full ability to autonomously take-off and maintain altitude.

Since 2005 we worked on the OS4 (our mini-VTOL flying robot) obstacle avoidance problem. This work presents the development and implementation of an obstacle avoidance controller for this flying robot using four onboard ultrasound (US) sensors for detecting obstacles and one onboard US sensor for controlling its altitude. Initially the OS4 mini-helicopter is introduced. Then, the simulation tool and control techniques developed are presented. The obstacle avoidance algorithms were first implemented and tested in simulated environments developed in MatLab/Simulink using an accurate dynamic model of OS4. Many approaches were considered with single and multiple vertical obstacles and the most promising one was finally embedded on OS4. Next, the implementation procedure and results obtained are addressed, and finally the conclusions are presented.

Nomenclature

a	= lift slope, dimensionless
A	= propeller disk area, m²
b	= thrust coeficient, N.s²
c	= propeller chord, m
C	= propulsion group cost factor, dimensionless
$\overline{C}_d$	= drag coefficient at 70% radial station, dimensionless
C_H	= hub coefficient, dimensionless
C_T	= thrust coefficient, dimensionless
C_Q	= drag coefficient, dimensionless
C_{Rm}	= rolling moment coefficient, dimensionless
D_i	= drag force for each propeller, N

d = drag coeficient, N.m.s²
F = forces, N
F_i = thrust force for each propeller, N
H = hub force, N
I_{xx} = inertia on x axis, kg.m²
I_{yy} = inertia on y axis, kg.m²
I_{zz} = inertia on z axis, kg.m²
J_r = rotor inertia, kg.m²
l = arm lenght, m
m = system overall mass, kg
Q = drag moment, N.m
R = rotation matrix, dimensionless
R_{rad} = propeller radius, m
R_m = rolling moment of a propeller, N.m
v = induced inflow velocity, m/s
T = thrust force, N
V = body linear speed, m/s

Greek Symbols

θ_0 = pitch angle of incidence, rad
θ_{tw} = twist pitch, rad
ψ = yaw angle, rad
ϕ = roll angle, rad
τ = torques, N.m
σ = solidity ratio, dimensionless
λ = inflow ratio, dimensionless
μ = motor advance ratio, dimensionless
ρ = air density, kg/
ω = body angular rate, rad/s
Ω_r = overall residual propeller angular speed, rad/s

Quadrotors

Today several research groups are working on mini-VTOLs based on the quadrotor configuration. Mistler et al. (2001) proposed a non-linear dynamic model and a feedback controller. Altuğ et al. (2002) related the use of visual feedback using one and two cameras (Altuğ et al., 2003) fixed on ground to estimate the quadrotor position and attitude. Hamel et al. (2002) studied the take-off and landing procedures by applying Lyapunov functions. Mokhtari and Benallegue (2004) developed a non-linear dynamic model based on Euler angles. When these angles are associated to Lyapunov functions, it is possible to control the helicopter roll, pitch, and yaw angles. Castillo et al. (2004) used the Lagrangian approach for modeling the quadrotor helicopter. The model was used together with Lyapunov functions and cyclic saturation algorithm to develop its controller. McKerrow (2004) developed a controller for hovering. Earl and D'Andrea (2004) developed a filter for estimating in real-time the roll, pitch, and yaw angles based on data from a gyroscope and a vision system fixed on ground. Tayebi and McGilvray (2004) proposed the use of retro-feeding controller based on quaternions for taking-off, hovering, and landing. In order to compensate the Coriolis and gyroscopic torques, they used PD and PD^2 controllers. Dunfied et al. (2004) developed an artificial neural network based controller to take-off, hover, and land. Guenard et al. (2005) proposed the use of an intuitive strategy based controller for taking-off and landing. In 2006 several studies focused on control techniques for quadrotors were published. The most relevant ones are: Benallegue et al. (2006), Bluteau et al. (2006), Castillo et al. (2006), Coza and Macnab (2006), Guenard et al. (2006), Madani and Benallegue (2006), Tayebi and McGilvray (2006), Voos (2006), and Xu and Ozguner (2006). Briefly, they proposed several control approaches based on Riccati equations, sliding mode technique, robust adaptive-fuzzy technique, and full state backstepping technique. In addition to these works, some papers were published in 2007: Besnard et al. (2007) developed a sliding mode disturbance observer to control a quadrotor; Erginer and Altuğ (2007) implemented a PD control and Tarek and Benallegue (2007-a and 2007-b) proposed a backstepping control and a sliding mode observer for quadrotors; and Voos (2007) used a control system based on a combination of state-dependent Riccati equations and neural networks to control the quadrotors attitude and velocity.

Recent works have discussed new approaches for many issues involving quadrotors. Nicol et al. (2008) proposes the implementation of one robust neural network approach for the quadrotor control. Lee et al. (2009) used Lyapunov based approach to control the device. Stepaniak et al. (2009) describe the full development of one electronic board to brushless speed control systems and also provide a very accurate model for the quadrotor so that it can either be remotely controlled or even fly autonomously. Huang et al. (2009) extends previous works on aerodynamics effects concerning quadrotors beyond hovering conditions. Courbon et al. (2009) presents a vision-based navigation strategy for autonomous flight. Kim et al. (2010) discuss the implementation of classic controllers for hovering maneuvers. Scaramuzza et al. (2010) proposes a novel vision based technique for localizing the aerial vehicle using a monocular downward camera.

The great majority of the publications focused on control techniques that allow the stable flight of the mini-VTOLs.

OS4 Mini-VTOL

OS4 (Fig. 1) is a small-scale helicopter with four rotors in cross configuration and represents the result of the design methodology developed at ASL (Bouabdallah et al., 2007). This figure shows the OS4 sensors, actuators and electronics: (a) inertial measurement unit, (b) altitude sensor below the robot, (c) obstacle avoidance sensor with tubes, (d) mini camera below the robot, (e) DSP 30F6014A (Microchip), (f) mother board, (g) motor module, (h) propeller, (i) battery, (j) remote control (RC) antenna, (k) Wi-Fi dongle.

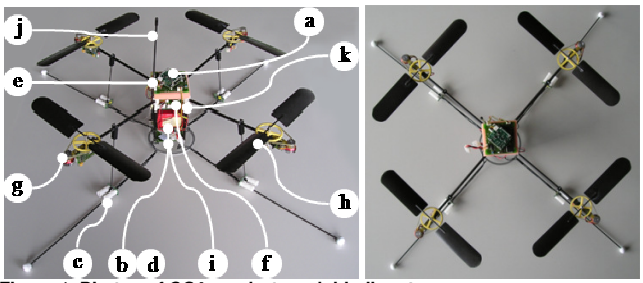


Figure 1. Photos of OS4 quadrotor mini-helicopter.

The OS4 total span is 800mm (300mm diameter propeller), about 200mm in height, the four arms are tilted by 5° with respect to the horizontal plane, and its total mass is about 650g. Its battery (Lithium-Polymer) takes almost one-half of the total mass. In contrast to this, the actuators take only one-third, thanks to brushless DC (BLDC) technology. They present 60W of 66W average power consumption. However, the last one depends on flight conditions and represents a weighted average between the equilibrium (40W) and the worst possible inclination state (120W) without losing altitude. For yaw angle and lateral displacements estimation we used a lightweight vision sensor. The GPS signal weakness and precision in cluttered environments made it difficult to be used. On the other hand, the surrounding metallic structures strongly disturb the IMU magnetic based yaw estimation. Thus, it was necessary to develop a lightweight visual positioning module. Embedding the controller for our application is definitely advisable as it avoids all the delays and the discontinuities in wireless connections. A miniature computer module (CM), based on Geode 1200 processor running at 266MHz

with 128MB of RAM and flash memory was developed. The computer module is x86 compatible and does offer all standard PC interfaces. The whole computer is 44g in mass, 56mm by 71mm in size and runs a Debian-based minimalist Linux distribution. The controller includes a microcontroller for Bluetooth chip interfacing with the computer module. The same MCU was used to decode the Pulse Position Modulation (PPM) signal picked-up from a 1.6g, 5 channels commercially available RC receiver. Due to this, it was possible to change the number of channels conveniently and control the robot using a standard remote control. Finally, a wireless LAN USB adapter was added. On the groundside, a standard Ground Control Software (GCS) for all our flying robots was designed. Presently, it permits environment visualization, waypoints and flight plans management as well as data logging and controller parameters tuning.

OS4 is equipped with a sonar-based obstacle avoidance system composed of four miniature ultrasound range finders (US) in cross configuration and altitude sonar (all sonars are SRF10 model, SRF10 Sensor, 2008), as it can be seen in Fig 1. The following table (Table 1) presents the OS4 parameters in detail.

Table 1. OS4 constructive parameters.

Name	Parameter	Value
Overall Mass	m	0.650kg
Inertia on x axis	I_{xx}	7.5e-3kg.m^2
Inertia on y axis	I_{yy}	7.5e-3kg.m^2
Inertia on z axis	I_{zz}	1.3e-2kg.m^2
Thrust Coefficient	b	3.13e-5Ns^2
Drag Coefficient	d	7.5e-7Nms^2
Propeller Radius	R_{rad}	0.15m
Propeller Chord	C	0.04m
Pitch Angle of Incidence	θ_0	0.26rad
Twist Pitch	θ_{tw}	0.045rad
Rotor Inertia	J_r	6e-5kg.m^2
Arm Length	L	0.23m

OS4 Dynamical Modeling

In order to obtain the OS4 dynamic model, we wrote the physical equations, got the parameters from its CAD model, and identified only the dynamics of the actuators which were considered important in the case of a quadrotor. This approach makes it easy to build dynamic models of instable systems, since we do not need to perform closed loop identification in flight. During the OS4 project we used several methods to obtain the models needed to simulate different behaviors. For instance, while Euler-Lagrange formalism and DC motor equations were used to model the test bench, Newton-Euler formalism (Murray et al., 1994), model identification, and blade element and momentum theories were used to model the OS4 quadrotor. In addition to this, Tait-Bryan angles were used for the parameterization. In the end, the OS4 model was implemented in a simulator (next section). The OS4 model developed in this section is a result of the following assumptions:

- The structure is supposed to be rigid and symmetrical;
- The Center of Gravity (CoG) and the body fixed frame origin are assumed to be coincident;
- The propellers are supposed to be rigid;
- Thrust and Drag Forces are considered proportional to the square of propeller speed.

Helicopters are considered complex mechanical systems because they encompass an enormous range of physical effects from the aerodynamics and the mechanics domains (Done and Balmford, 2001). Due to this, a good quadrotor model should consider as much as possible important effects, including the gyroscopic ones. A short list of the main effects acting on a helicopter is briefly described in Table 2 (Mullhaupt, 1999).

Table 2. Main physical effects acting on a helicopter.

Effect	Source	Formulation
Aerodynamics	Propeller rotation and blades flapping	$C\Omega^2$
Inertial Counter Torques	Change in propeller rotation speed	$J\dot{\Omega}$
Gravity	Center of Mass position	-
Gyroscopic	Change in orientation of the rigid body and propeller plane	$I\theta\psi$ $J\Omega_r\theta,\phi$
Friction	Any helicopter motion	$C\dot{\phi},\dot{\theta},\dot{\psi}$

where: J is the inertia; I, inertia moment; Ω, propeller angular rate; C, propulsion group cost factor; Ω_r, overall residual propeller angular speed; ϕ, roll angle; θ, pitch angle; and ψ, yaw angle.

The OS4 model was developed based on successive steps as presented in previous papers (Bouabdallah, 2004-a, Bouabdallah and Siegwart, 2005-a and 2005-b). Its last version includes Hub Forces (H), Rolling Moments (R_m), and variable aerodynamical coefficients. This makes the model more realistic especially in forward flight.

Let us consider an earth-fixed frame E and a body-fixed frame B as presented in Fig. 2. Using Euler angles parameterization, airframe orientation in space is given by a rotation R from B to E, where $R \in$ SO3 is the rotation matrix. The frame system (Fig. 2) is in conformity with the N, E, D (North, East, Down) standard, following by the way the coordinate system of our inertial sensor (3DM-GX1). In this figure, ω_i represents the motor angular rate for each propeller; D_i, drag force for each propeller; F_i, Thrust Force for each propeller; x, y, z, OS4 position in body coordinate frame; X, Y, Z, OS4 position in earth coordinate frame; and ϕ, θ, ψ, respectively, OS4 roll, pitch and yaw angles.

Then Eq. (1) represents the dynamics of a rigid body under external forces applied to the center of mass:

$$\begin{bmatrix} mI_{3x3} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times I\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix}. \quad (1)$$

where: m is the overall OS4 mass; V, body linear speed; ω, body angular rate; F, forces; and τ, torques.

Aerodynamic forces and moments are derived using a combination of momentum and blade element theory (Leishman, 2006). Leishman work was based on the work of Gary Fay during

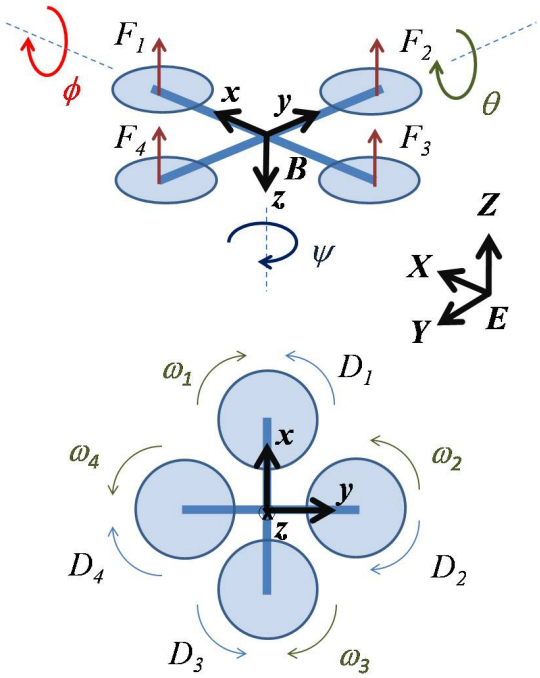


Figure 2. OS4 local coordinate system.

Mesicopter project (Fay, 2001). For an easier reading of the equations below, we recall some symbols: σ represents solidity ratio; λ, inflow ratio; a, lift slope; v, induced velocity; μ, motor advance ratio; and ρ, air density. The Thrust Force (T) is the resultant of the vertical forces acting on all the blade elements:

$$\begin{cases} T = C_T \rho A (\Omega R_{rad})^2 \\ \dfrac{C_T}{\sigma a} = \left(\dfrac{1}{6} + \dfrac{1}{4}\mu^2\right)\theta_0 - (1+\mu^2)\dfrac{\theta_{t\omega}}{8} - \dfrac{1}{4}\lambda \end{cases}. \quad (2)$$

where: C_T is the thrust coefficient; R_{rad}, rotor radius; A, propeller disk area; θ_0, pitch of incidence; and $\theta_{t\omega}$, twist pitch.

The Hub Force (H) is resultant of horizontal forces acting on all blade elements:

$$\begin{cases} H = C_H \rho A (\Omega R_{rad})^2 \\ \dfrac{C_H}{\sigma a} = \dfrac{1}{4a}\mu \overline{C}_d + \dfrac{1}{4}\lambda\mu\left(\theta_0 - \dfrac{\theta_{t\omega}}{2}\right) \end{cases}. \quad (3)$$

where: $\overline{C}_d$ is the drag coefficient at 70% radial station and C_H, the hub coefficient.

In addition to this, Drag Moment (Q), i.e., the moment about the rotor shaft caused by the aerodynamic forces acting on the blade elements, is given by Eq. (4). One may notice that the horizontal forces acting on the rotor are multiplied by the moment arm and integrated over the rotor. Drag moment determines the power required to spin the rotor.

$$\begin{cases} Q = C_Q \rho A (\Omega R_{rad})^2 R_{rad} \\ \dfrac{C_Q}{\sigma a} = \dfrac{1}{8a}(1+\mu^2)\overline{C}_d + \lambda\left(\dfrac{1}{6}\theta_0 - \dfrac{\theta_{t\omega}}{8} - \dfrac{1}{4}\lambda\right) \end{cases}. \quad (4)$$

where: C_Q is the drag coefficient.

The rolling moment of a propeller (R_m) - Eq. (5) - exists in forward flight when the advancing blade is producing more lift than the retreating one. It is the integration over the entire rotor of the lift of each section acting at a given radius. This should not be confused with propeller radius or the rotation matrix R or the overall rolling moment which is caused by a number of other effects.

$$\begin{cases} R_m = C_{R_m} \rho A (\Omega R_{rad})^2 R_{rad} \\ \dfrac{C_{R_m}}{\sigma a} = -\mu\left(\dfrac{1}{6}\theta_0 - \dfrac{\theta_{t\omega}}{8} - \dfrac{1}{8}\lambda\right) \end{cases}. \quad (5)$$

where C_{Rm} is the rolling moment coefficient.

Helicopters operating near the ground (~ at half rotor diameter) experience thrust augmentation due to better rotor efficiency. This is related to a reduction of induced airflow velocity. This is called Ground Effect. Literature presents different approaches to deal with this effect, for instance, by using adaptive techniques (Guenard et al., 2006). However, the principal aim in this project is to find a model of this effect for OS4 to improve the autonomous take-off and landing controllers. The goal is to obtain a simple model capturing mainly the variation of induced inflow velocity (v).

Cheeseman states (Cheeseman and Bennet, 1957) that at constant power, the Thrust Force out of ground effect (OGE) is equal to the Thrust Force in ground effect (IGE), i.e., $T_{OGE v_i OGE} = T_{IGE v_i IGE}$. The velocity induced at the rotor center by its image is $\delta v_i = A v_i / 16\pi z^2$, where z is the altitude. Cheeseman obtained Eq. (6) by assuming that v_i and δv_i are constant over the disk which allows $v_{i,IGE} = v_i - \delta v_i$.

$$\frac{T_{IGE}}{T_{OGE}} = \frac{1}{1 - \dfrac{R_{rad}^2}{16 z^2}}. \quad (6)$$

Another simple way to proceed is to consider that the inflow ratio in ground effect (IGE) is $\lambda_{IGE} = (v_{i,OGE} - \delta v_i - \dot{z})/\Omega R_{rad}$, where the variation of induced velocity is $\delta v_i = v_i/(4z/R_{rad})^2$. We can then rewrite the thrust coefficient (Eq. 2) IGE as follows:

$$\begin{cases} T_{IGE} = C_T^{IGE} \rho A (\Omega R_{rad})^2 \\ \dfrac{C_T^{IGE}}{\sigma a} = \dfrac{C_T^{OGE}}{\sigma a} + \dfrac{\delta v_i}{4\Omega R_{rad}} \end{cases}. \quad (7)$$

Then we compared the variation of inflow velocity in and out of ground effect using OS4 simulator. The influence is perceptible for $z/R_{rad} \approx 2$ but becomes important near $z/R_{rad} < 1$. It seems that in the case of a quadrotor the ground effect influence is already present at one rotor diameter and becomes really important at one rotor radius. In order to empirically verify this assumption, we conducted a simple experiment which proved that a quadrotor deprived of altitude control is able to hover at a constant altitude at nearly one rotor diameter from the ground. It is clear that this result is only an indication of validity and does not constitute a formal proof. Quadrotor motion is obviously caused by a series of forces and moments coming from different physical effects. This model considers the following ones, presented in Tab. 3.

More detailed information concerning this model can be found in Appendix B at Bouabdallah (2007). The equations of motion (Eq. 8 to 13) are derived from Eq. (1) and all the forces and moments listed in Tab. 3.

Table 3. Physical Effects considered in OS4 Dynamical Modeling.

Physical Effect Considered		Equation
Rolling Moments	Body gyro effect	$\dot{\theta}\dot{\psi}(I_{yy}-I_{zz})$
	Propeller gyro effect	$J_r\dot{\theta}\Omega_r$
	Roll actuator action	$l(-T_2+T_4)$
	Hub moment due to sideward flight	$h\left(\sum_{i=1}^{4}H_{y_i}\right)$
	Rolling moment due to forward flight	$(-1)^{i+1}\sum_{i=1}^{4}R_{m_i}x_i$
Pitching Moments	Body gyro effect	$\dot{\phi}\dot{\psi}(I_{zz}-I_{xx})$
	Propeller gyro effect	$J_r\dot{\phi}\Omega_r$
	Pitch actuator action	$l(T_1-T_3)$
	Hub moment due to forward flight	$h\left(\sum_{i=1}^{4}H_{x_i}\right)$
	Rolling moment due to sideward flight	$(-1)^{i+1}\sum_{i=1}^{4}R_{m_i}y_i$
Yawing Moments	Body gyro effect	$\dot{\phi}\dot{\theta}(I_{xx}-I_{yy})$
	Inertial counter-torque	$J_r\dot{\Omega}_r$
	Counter-torque unbalance	$(-1)^{i}\sum_{i=1}^{4}Q_i$
	Hub force unbalance in forward flight	$l(H_{x_2}-H_{x_4})$
	Hub force unbalance in sideward flight	$l(-H_{y_1}+H_{y_3})$
Forces along z Axis	Actuators action*	$c\psi c\phi\left(\sum_{i=1}^{4}T_i\right)$
	Weight	mg
Forces along x Axis	Actuators action*	$(s\psi s\phi+c\psi s\theta c\phi)\left(\sum_{i=1}^{4}T_i\right)$
	Hub force in x axis	$-\sum_{i=1}^{4}H_{x_i}$
	Friction	$\frac{1}{2}C_xA_c\rho\dot{x}\lvert\dot{x}\rvert$
Forces along y Axis	Actuators action*	$(-c\psi s\phi+s\psi s\theta c\phi)\left(\sum_{i=1}^{4}T_i\right)$
	Hub force in x axis	$-\sum_{i=1}^{4}H_{y_i}$
	Friction	$\frac{1}{2}C_yA_c\rho\dot{y}\lvert\dot{y}\rvert$

* where: *s* represents sine function and *c*, cosine.

OS4 is equipped with four fixed-pitch rotors (no swash plate), each one includes a Brush-Less Direct Current (BLDC) motor, a one-stage gearbox and a propeller. The entire rotor dynamics was identified and validated using the MatLab Identification Toolbox.

$$I_{xx}\ddot{\phi}=\dot{\theta}\dot{\psi}(I_{yy}-I_{zz})+J_r\dot{\theta}\Omega_r+\ldots \\ l(-T_2+T_4)-h\left(\sum_{i=1}^{4}H_{y_i}\right)+(-1)^{i+1}\left(\sum_{i=1}^{4}R_mx_i\right), \quad (8)$$

$$I_{yy}\ddot{\theta}=\dot{\phi}\dot{\psi}(I_{zz}-I_{xx})-J_r\dot{\phi}\Omega_r+l(T_1-T_3)+\ldots \\ h\left(\sum_{i=1}^{4}H_{x_i}\right)+(-1)^{i+1}\left(\sum_{i=1}^{4}R_my_i\right), \quad (9)$$

$$I_{zz}\ddot{\psi}=\dot{\theta}\dot{\phi}(I_{xx}-I_{yy})+J_r\dot{\Omega}_r+(-1)^{i}\left(\sum_{i=1}^{4}Q_i\right)+\ldots \\ l(H_{x_2}-H_{x_4})+l(-H_{y_1}+H_{y_3}), \quad (10)$$

$$m\ddot{z}=mg-(\cos\psi\cos\phi)\sum_{i=1}^{4}T_i, \quad (11)$$

$$m\ddot{x}=(\sin\psi\sin\phi+\cos\psi\sin\theta\cos\phi)\sum_{i=1}^{4}T_i-\ldots \\ \sum_{i=1}^{4}H_{x_i}-\frac{1}{2}C_xA_c\rho\dot{x}\lvert\dot{x}\rvert, \quad (12)$$

$$m\ddot{y}=(-\cos\psi\sin\phi+\sin\psi\sin\theta\cos\phi)\sum_{i=1}^{4}T_i-\ldots \\ \sum_{i=1}^{4}H_{y_i}-\frac{1}{2}C_yA_c\rho\dot{y}\lvert\dot{y}\rvert. \quad (13)$$

A first-order transfer function (Eq. 14) is sufficient to reproduce the dynamics between the propeller speed set-point and its true speed. It is worthwhile to note the non-unity gain in Eq. (14). This is visible in Fig. 3 that superimposes the model output and the sensor data on a step input. In fact, sensorless BLDC motors require a minimum speed to run. Thus, the set-point does not start from zero. The motor used does not incorporate hall effect sensors; the identification was carried out using a reflective encoder placed under the propeller gear.

$$G(s)=\frac{0.936}{0.178s+1}. \quad (14)$$

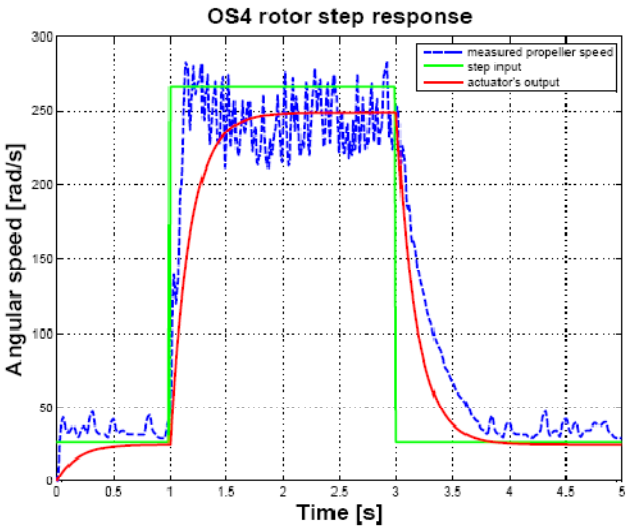


Fig. 3. Rotor and model step response, measured at propeller's shaft.

OS4 Simulator

Aiming to assist the control design phase, we developed a simulation and analysis tool based on MatLab/Simulink. This enables the use of model-based design from the application definition, to the controller design and simulation. The simulator is used for control and obstacle avoidance simulations and visualizations. The user has many options in order to execute the

simulation by selecting the desired combination between mini-VTOL model, sensors, controllers and environments in the libraries (Fig. 4).

It is possible for example to combine various types and quantities of sensors with different control approaches in different environments. Another interesting characteristic of the tool is that the libraries accept the inclusion of new models for sensors, mini-VTOLs, controllers and environments. The results can be visualized using graphical interfaces. Simulink model considers OS4 dynamical model developed in the previous section. So, it utilizes hub forces and rolling moments based on the literature (Done and Balmford, 2001 and Fay, 2001), and in addition to this, we implemented air friction model and included inertial counter-torques in yaw dynamics. The whole dynamical model is a composition of all these effects in one mathematical representation. We use a first-order actuator dynamics captured by identification. A first-order model is a reasonable simplification that was validated with different sets of data. The dynamics simulator includes all the delays measured and the noise estimated on the real robot. The results in simulation were satisfying and we are confident that they are close to reality. In fact, we used exactly the same MatLab controller parameters in the real flying experiments (Bouabdallah et al., 2007).

Concerning the simulator, Fig. 5 presents a simplified diagram of the MatLab/Simulink simulator developed. Figures 6 and 7 show typical results obtained during the simulations, respectively the OS4 mini-helicopter attitude angles and its 3D path in an environment with obstacles.

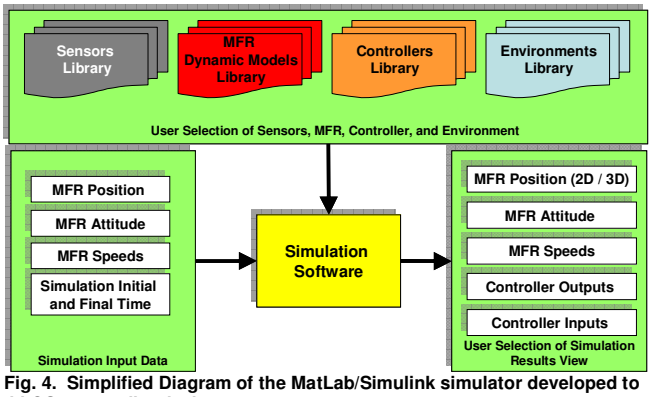


Fig. 4. Simplified Diagram of the MatLab/Simulink simulator developed to aid OS4 controller design.

OS4 Controllers

Since the beginning of the OS4 project at ASL in 2003, we have developed and tested several approaches for controlling. Five techniques were explored from theoretical development to final experiments. In the beginning we tested on OS4 two linear controllers, a PID and an LQR, based on a simplified model.

We obtained an autonomous hover flight behavior (Bouabdallah et al., 2004-a). Later we reinforced the control using backstepping techniques (Bouabdallah and Siegwart, 2005-a). Another improvement was introduced thanks to integral backstepping. With this technique, OS4 was able to perform autonomous hovering with altitude control and autonomous take-off and landing (Bouabdallah et al., 2007). After the evaluation of all the control approaches tested during the project, it became clear that the way to follow was a combination between PID and Backstepping into the so-called Integral Backstepping. The goal was to bring together the robustness against disturbances offered by Backstepping and robustness against model uncertainties offered by the integral action. This shall permit more complex flight maneuvers than a simple hovering. After a phase of extensive simulation and experimentation, Integral Backstepping was proposed as a single approach for attitude, altitude and position control.

The difficulties encountered in OS4 control included sensor quality, yaw drift, and robustness against large disturbances and model uncertainties. Sensor noise is inherent to micro IMUs and is dramatically amplified on helicopters. This degrades sensor accuracy and accelerates drift. Yaw drift is one of the most annoying issues as the contribution of yaw control in the overall control is important. The best robustness against large disturbances was achieved using backstepping technique, while model uncertainties were cancelled thanks to integral action. Thus, integral backstepping has been proposed for full control of quadrotors.

Thanks to this technique, OS4 has been able to perform autonomous hovering with altitude control and autonomous take-off and landing. More detailed information concerning the OS4 controllers may be found in Bouabdallah and Siegwart (2007) and Bouabdallah (2007).

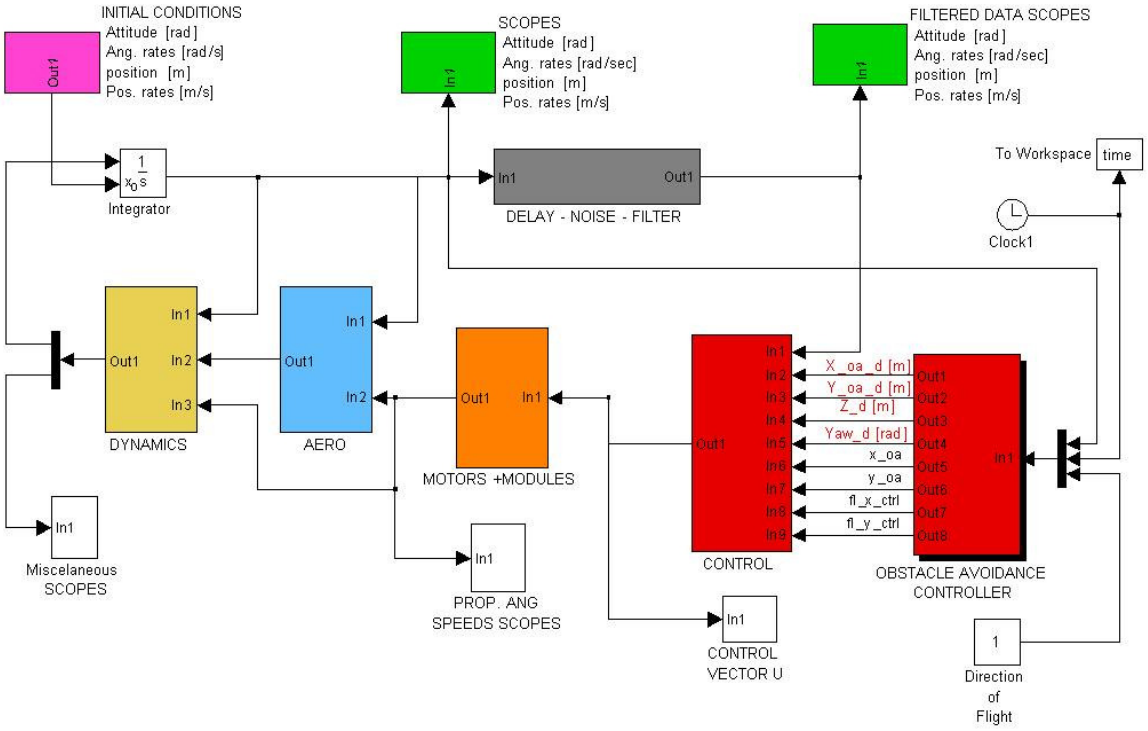


Fig. 5. MatLab /Simulink OS4 model. Observe that this model includes the obstacle avoidance controller.

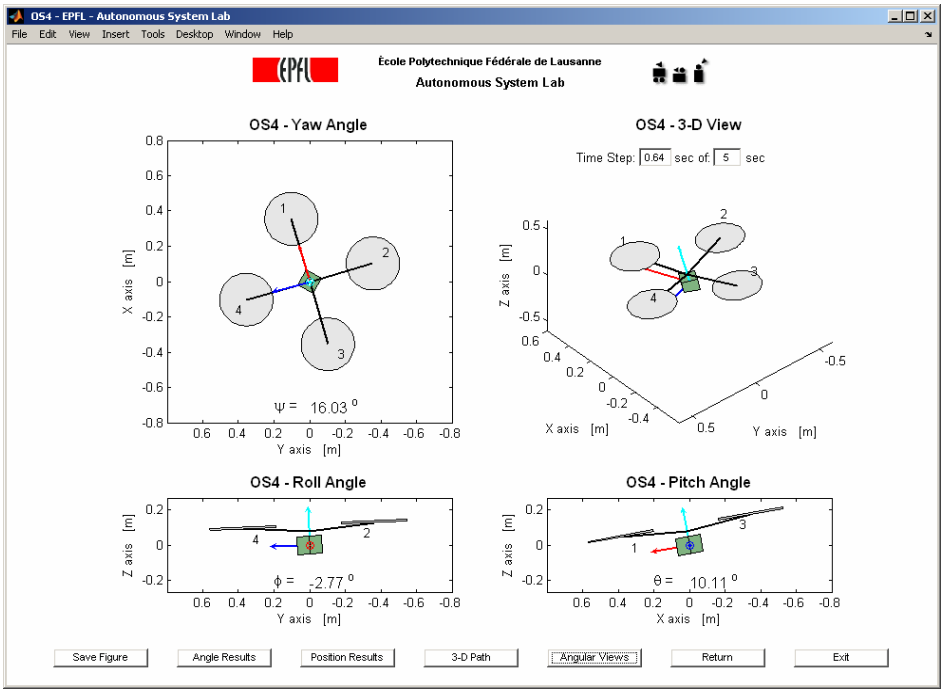


Fig. 6. OS4 attitude angles view during the simulation.

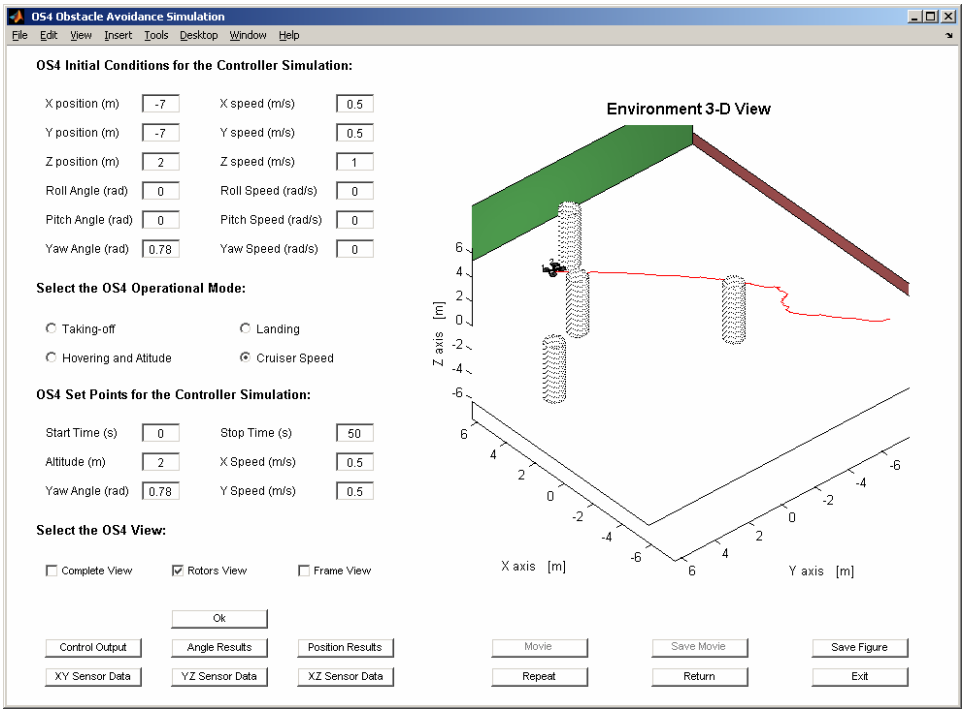


Fig. 7. OS4 3D path during the simulation. In this case, cylinders represent static obstacles in the environment.

Obstacle Avoidance

Literature provides several works focused on navigation and obstacle avoidance procedures for helicopters. Undoubtedly, some of the early pioneers in autonomous navigation for helicopters worked at NASA Ames Research Center. In 1980 and 1990 decades they have published a series of papers highlighting some techniques developed for automatic Nap-Of-the-Earth flights such as computer vision (Sridhar and Cheng, 1988), integration of active and passive sensors (Cheng and Sridhar, 1990), design of control strategies tested in 3D computer simulations (Cheng, 1990; Cheng and Lam, 1992; Zelenka et al., 1993). In the beginning, the authors developed 2D models of the environment and latter on they extended the path search techniques to 3D in order to obtain a low-altitude guidance system for military helicopters.

Zapata and Lépinay (1999) applied the Deformable Virtual Zone (DVZ) approach, originally designed for land and submarine mobile robots, to helicopters. They performed some simulated experiments in an 3D environment with several obstacles using an extremely simple helicopter model and a graphic simulator implemented in MatLab. Based on the simulations, they concluded that the DVZ can be considered an efficient algorithm to obtain local and reactive obstacle avoidance behaviors. In addition to this, they emphasized two main problems found to implement the DVZ procedure onboard an electric small-size helicopter: the need of a helicopter complex dynamic model and an efficient onboard perception system to model the environment that surrounds the helicopter. Since then, many researchers worked on reliable dynamic models for helicopters with different rotor configurations (co-axial, main and tail, 2 rotors, 4 rotors, etc.) and a large quantity of normal and small-size sensors were developed. Today it is possible to implement a set of onboard sensors for small-size helicopters at reasonable prices. The combined onboard use of GPS/INS navigation system, cameras, laser scanners, and powerful processors makes the sensor fusion for scene 3D mapping, estimation of the helicopter state, detection of obstacles possible. For instance, Kanade et al. (2004) implemented a promising real-time 3D vision system onboard in Yamaha R50 helicopter for outdoor applications. Unfortunately, this is not the case for micro and mini-helicopters. Development in micro and mini-size sensors is still needed to allow their onboard use. Due to this issue, there is a lack of micro and mini-helicopters able to navigate autonomously based only on onboard sensors. This is clearly a strong restriction factor for the use of micro and mini-helicopters. Some authors try to overcome this limitation by using external sensors placed in the environment. Nevertheless this solution is not self-contained and it requires more calibrated sensors and higher data transmission rates or an external host computer.

Thanks to recent advances on sensor fields, simulation tools, and UAV VTOL-like models mentioned above, several authors reported interesting applications in obstacle avoidance for outdoor use (most of them developed only simulations). Bae and Kim (2004) simulated obstacle avoidance methods for UAVs based on chaos trajectory surfaces. The combined use of optical flow and stereo-based navigation for UAVs in urban canyons was reported in Hrabar et al. (2005). They used the optical flow from a pair of sideways-looking cameras to stay centered in a canyon and initiate turns at junctions, while stereo vision from a forward-facing stereo head was used to avoid obstacles. He et al. (2006) proposed a hierarchical framework to deal with uncertainty and noise in motion field analysis, so as to develop a low-complexity and reliable vision analysis system for UAV navigation. Wang et al. (2007) addressed the formation flying of multiple UAVs navigating through an obstacle-laden environment using Grossberg neural networks (GNN). They carried out several simulations and concluded that the onboard implementation in small-size UAVs of a modified GNN is feasible for real-time applications in obstacle-rich environments. Zengin and Dogan (2007) ran simulations of a gradient search algorithm for real-time target tracking for autonomous UAVs. As their strategies in decision making considered the UAV dynamic constraints, the simulation results were very realistic (all heading and speed commands were feasible). Andert and Goormann (2007) combined grid and feature-based occupancy mapping using data extracted by an UAV with stereo vision (maxiARTIS). They produced useful maps of the UAV surrounds. Further improvements

are necessary to obtain reliable global maps that can be used for autonomous path planning in real-time applications. Boivin et al. (2008) designed and simulated a decentralized control strategy with cooperation to engage UAVs towards several targets while avoiding static obstacles detected en route. The algorithms were based on a predictive control scheme and the UAVs dynamic constrains were taken into account. Hrabar (2008) presented a novel combination of techniques that could allow a UAV to navigate safely in outdoor environments while performing tasks (for instance, the inspection of power lines). He combined probabilistic roadmaps and D* Lite approaches for path planning with stereo-based occupancy mapping for dynamic replanning. He carried out several experiments in simulation and with a cable array robot and the system achieved promising results. However, the system failure rate was too high for the desired application. Paul et al. (2008) proposed a potential fields-based solution for collision and obstacle free formation flight of UAV groups. In order to verify the algorithm performance, they did 3D simulations using a simplified helicopter model implemented in MatLab/Simulink.

The design of an in-flight collision avoidance controller for micro and mini-VTOLs relying only on onboard sensors is challenging. Most of the literature brings simulated results essentially. Bouktir et al. (2008) achieved promising results in simulation. They proposed a method that is able to generate time-optimal trajectories for a micro quadrotor based on its trajectory parameterization and using a nonlinear optimization technique. However, the authors did not mention how the UAV and onboard sensor models were implemented (delays, noises, etc.), neither any information concerning the sensors characteristics or onboard computational needs. Concerning the full implementation of an UAV, Roberts et al. (2007) presents a full autonomous indoor and hands-off mini-UAV. They have presented outstanding results with the "Quadrotor", achieving their goals which were to make the UAV automatically take-off, control constant altitude level, accomplish the obstacle avoidance demands, autonomously treat the anti-drift problem and land in safe. Besides, one important point lies in the fact that authors achieved that using very simple sensing and control strategies.

Obstacle Avoidance Sensors and Controller

Due to the lack of researches that focus on real onboard implementation of such navigation and obstacle avoidance systems in mini-helicopters (particularly when it comes to mini-quadrotors), we decided to concentrate our attention on this topic. In this work indoor environments are foregrounded. In order to develop and implement obstacle avoidance procedures onboard our OS4 mini-helicopter, we firstly verified the available sensors on the market. They should be as light as possible and provide low power consumption in order to reduce their impact on the OS4 overall flight autonomy. Natural candidates to provide the necessary perception system were US sensors, linear cameras, IMUs, and gyroscopes. We checked several sensor datasheets and prices and tested some sensors trying to find the best balance between their range, noise, power consumption, size, and weight.

We have chosen US SRF10 sensors (SRF10 Sensor, 2008). In the end we decided to add a plastic cylinder on the sensor emitter-receiver end to reduce the sensor cone angle and increase its maximum range (Bouabdallah, 2007). This kind of sensor, by default, is very sensible to objects whose position is out of the line of vision. However, this is not an advantage in our case. Their maximum reading frequency was 15Hz (acquisition every 65ms). We used 5 US sensors: four on OS4 structure (US sensor #1 to #4) to detect obstacles and one vertically downwards assembled to measure its altitude (US sensor #5). Only one US sensor was fired a time. The ultrasound sensors are used aiming to provide the collision avoidance system with real-time information.

As our goal is to prevent situations of collision, where we don't know the exact position and the speed of the quadrotor, the obstacle itself is the reference point for evasive closed-loop maneuvers. In this case, it is more advantageous that the ultrasound sensors are able to detect obstacles from a long range of distance and with accuracy than that it has a large field of vision, so that the collision can be precisely avoided.

For this purpose, the above mentioned tubes were used to cover the sensor in order to limit its wide field of vision and increase its range. Many tubes of different compositions where used such as plastic, compact foam, paper and cloth. The best results were obtained using a 3cm plastic tube which allowed us to increase the sensor's range of vision, as it can be seen from Fig. 8. Still, the length of the tube must be observed for the sensor's appropriate behavior.

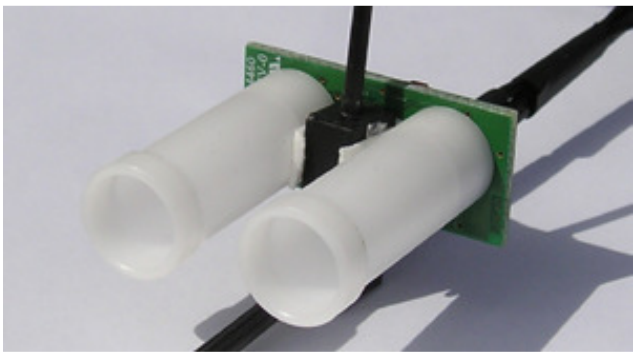


Fig. 8. Plastic tube recovering the US aiming the operation improvement.

The volume of detection of the US sensor with and without the tubes can be evaluated from Fig. 9, whereas the orange line represents the sensing volume without the tubes and the blue series were acquired with the use of them. Without the plastic tubes, the sensing volume is broader at the three axes. When they were used, they straiten the angle of vision indeed. However, the range of vision increases considerably along the Y axis, which is desirable.

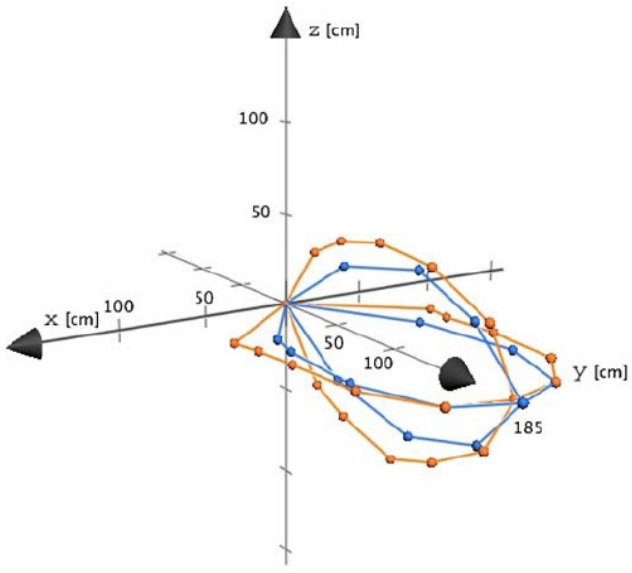


Fig. 9. Volume captured by the US with the plastic tubes (blue series) and without the plastic tubes (orange series).

An important aspect concerning the use of the ultrasound sensor refers to its position in the structure of the quadrotor. Three positions were considered, taking into account the sensor's angle of vision, as it can be seen from Fig. 10. By its simplicity, the sensor was positioned at the intersection point that the vertical stem and the skids form (position 3 in Fig. 10). Thus, the angle of vision is around 41° (which is still broad), and free of the blades interruption.

The main disadvantage of this set is that, at 41°, there are four dead zones in which the sensors cannot accurately detect the obstacle. One possible solution for such problem would be to bring the sensors to positions 1 and 2 (Fig. 10). However, it would make the blades to be permanently at the field of vision. Furthermore, the minimum angle of vision required to allow good results is 35°. Figure 11 shows the position of the US in the quadrotor's skids.

There are several sensors that could solve this problem indeed. However, the ultrasound is appropriated thanks to its low weight. Besides, the direction of flight is imposed to always be in the sensors direction, which can help improving the quadrotor's vision.

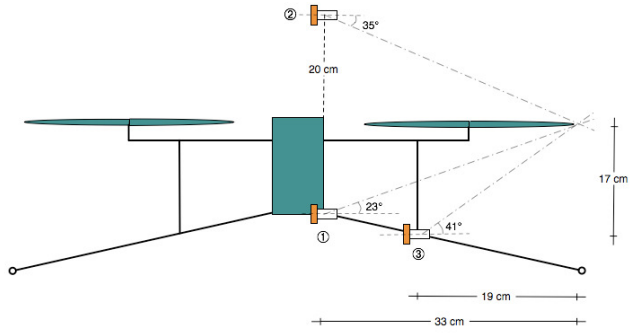


Fig. 10. All three possible positions for the ultrasound sensor.

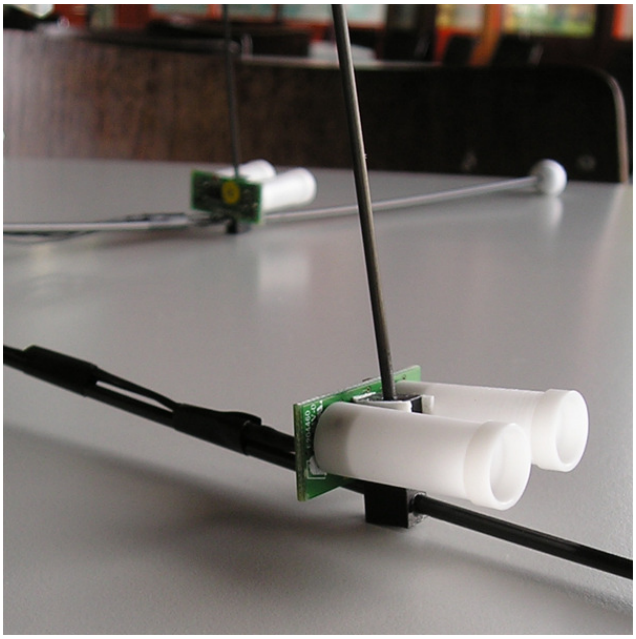


Fig. 11. Two of the four ultrasounds placed at the base of the quadrotor's skids.

The Inertial Measurement Unity embedded in the OS4 is the Microstrain 3DM-GX1 (MicroStrain IMU, 2008) which acts as an accelerometer, a gyroscope and also as a magnetometer, simultaneously. Figure 12 shows a close view of de device. We firstly tried to use the IMU accelerations that were acquired at 76Hz to estimate the OS4 speeds. The idea was to store the IMU data onboard and then numerically integrate it. However, this data was extremely noisy (Fig. 13) and even after adding a filter, we got some speed peaks that exceeded 15m/s (we expected a signal close to 0 m/s because the helicopter was hovering). Then, it was necessary to calculate the average value of the last 5 measurements in order to minimize the high frequency noise which is present while data is acquired. As it can be seen from Fig. 14, the measured signal related to the acceleration in X axis does not represent the real acceleration.

After the above mentioned procedures, the speed is numerically calculated by integrating the accelerations, as follows:

$$\dot{x}(t) = \int_0^t \ddot{x}\,(t) \cdot dt \qquad (15)$$

$$\dot{y}(t) = \int_0^t \ddot{y}\,(t) \cdot dt \qquad (16)$$

Fig. 12. Microstrain 3DM-GX1 Inertial Measurement unity.

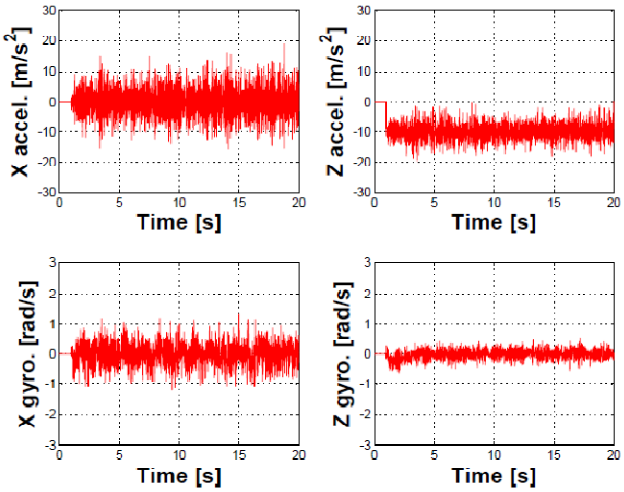


Fig. 13. OS4 Accelerations (m/s²) and Angular Speeds (rad/s) during experiment.

As it is notable from Fig. 15, the filtering process provides the boarding computer with a more realistic data that income from the sensor, which contributes directly for an optimized control process.

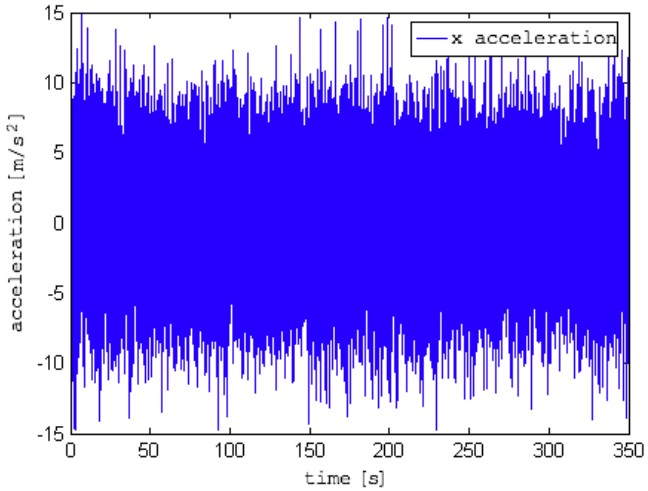


Fig. 14. Noisy linear acceleration along X axis during measurement process.

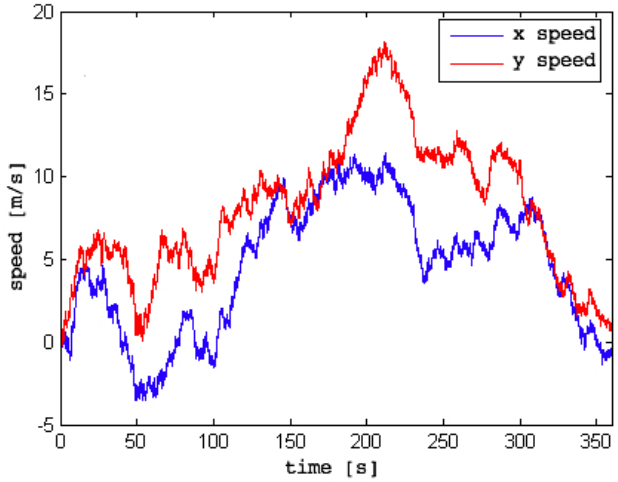
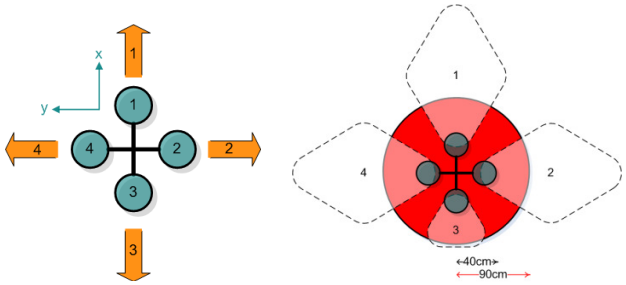


Fig.15. Estimated linear speed by numeric integration with the use of filtering (average of last 5 measures).

Next, we decided to use our OS4 Simulator to test different approaches before implementing them onboard the mini-helicopter. We modeled the sensors and included their delays and noises. Apart from that, we introduced the obstacle avoidance controller (OAC) into the Simulink model and inserted indoor environment and sensor libraries. Thanks to our reliable OS4 dynamical model and OS4 Simulator we could simulate OS4 behavior and verify its controllability while avoiding obstacles in indoor environments. Depending on the environment selected, OS4 would negotiate its path with mobile and/or static obstacles. Obstacles were modeled as vertical cylinders with different diameters and heights. It was also possible to select the desired OS4 behavior during the OAC simulation (check Fig. 7: "Select the OS4 Operational Mode"), for instance: hovering or keeping cruiser speed while not avoiding obstacles, landing, taking-off, etc.

Aiming to simplify the procedure, we decided to keep the helicopter altitude (z) constant during the OAC maneuvers (it moves in a quasi-horizontal plane with a fixed altitude). This would reduce the path planning complexity to a 2D problem. We also restricted its flight direction: OS4 can move only on the four directions where the US sensors were placed (Fig. 16-a). To increase the flight safety, a 90cm-radius security zone is constantly maintained between the helicopter and the environment (Fig. 16-b). This security zone assures a 50cm-distance between the helicopter rotors and any obstacle. If an obstacle is detected inside the security zone, a safety loop (that runs in parallel to the OAC) interferes in the helicopter flight control and generates an evasive maneuver. This maneuver is obtained by selecting a predefined pitch and/or roll angle(s) that would avoid a collision between the helicopter and the obstacle(s).

During the OAC simulation phase using the MatLab/Simulink tool, we developed and simulated several obstacle avoidance approaches (Becker et al., 2006 and Bouabdallah, 2007). All of them were based on the premise that during the real implementation phase we would install onboard sensors that would provide OAC with data concerning OS4 position and/or speed (see next section, implementation, for details about the sensors). Therefore the developed approaches can be divided into two categories: relative position and speed-based approaches. The first OAC category has as output the desired OS4 relative positions (x_d, y_d), and the second one, the desired OS4 speeds ($\dot{x}_d$, $\dot{y}_d$).



(a) (b)

Fig. 16. Visualization of the 4 flight directions (a) and the security zone around the helicopter (b).

Due to this, we also designed two additional controllers: the position and the speed controllers (both are represented in Fig. 17). The position controller uses as input data the desired OS4 position (x_d, y_d) generated by the OAC and the real position (x, y) estimated by the sensors and generates as outputs the desired OS4 pitch (θ_d) and roll (ϕ_d) angles. A novel solution for localization is presented in Scaramuzza et al., 2010. The authors have proposed a state of the art visual odometry system with millimeter precision, using a single downward looking monocular camera associated with the use of state-of-the-art visual SLAM algorithms. Likewise, the speed controller uses as input data the desired speeds ($\dot{x}_d$, $\dot{y}_d$) generated by the OAC and the real speeds ($\dot{x}$, $\dot{y}$) estimated by the sensors and generates as outputs the desired OS4 pitch (θ_d) and roll (ϕ_d) angles. Depending on the approach adopted, the OS4 yaw angle (ψ) is either kept constant or used to produce the evasive maneuver.

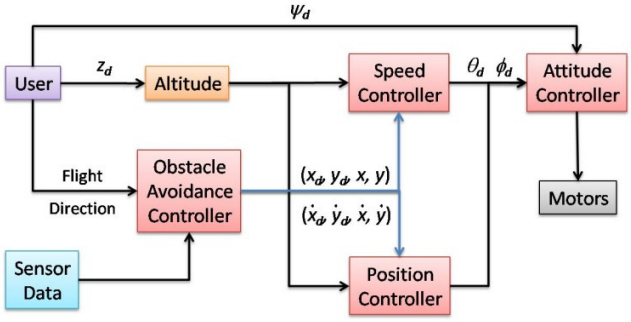


Fig. 17. OS4 Obstacle Avoidance Controller. The user selects which OAC is desired: position-based or speed-based approach.

Implementation

The general architecture of the OS4 onboard system is presented in Fig. 18. The data concerning the attitude of the helicopter is provided by the IMU and are directly accessible to the DSP. The wireless connection makes it possible to get real data during the experimental test. In order to keep a reliable and stable unit of time, the DSP cycle is synchronized with the IMU (13ms). The DSP programming was written in C.

It is also possible to control the mini-VTOL using a standard radio control (RC). Taking into account that anyone who wants to implement control software onboard a helicopter must consider safety issues relating to the helicopter and user, we implemented a security control layer which inhibits robot starting if the RC is not detected and/or in case it is not turned on manual mode with the throttle at minimum. Proper operation of each sensor is also verified before taking-off. If, for any reason, the contact in flight with the RC is lost, the safety layer automatically lands the helicopter. The implementation of a control loop on a helicopter must be done in a very careful way. In fact, the most critical part is the attitude loop. It must be strictly deterministic with the highest priority over the other processes (except for the safety control layer). On OS4, this loop is synchronized with the IMU and is able to deal with temporary loss of its data.

When flying, position control keeps the helicopter over the desired place, i.e.: the (x, y) horizontal position with regard to a starting point.

Horizontal motion is achieved by orienting the thrust vector towards the desired direction of motion. This is done by rotating the vehicle itself in the case of a quadrotor. In concrete terms, one performs position control by rolling or pitching the helicopter in response to a deviation of the y_d or x_d references respectively. Thus, the position controller outputs the attitude references ϕ_d and θ_d, which are tracked by the attitude controller.

Frontwards, backwards, and sidewards movements require complex maneuvers, since mini-quadrotors have a peculiar dynamics and they are hard to control when high speeds are reached in indoor environments. In order to move horizontally frontwards, for instance, pitch positive movement was made and was followed by a pitch negative one (see simulation detail in Fig. 19). This strategy allows better speed and position control. This intermittent movement avoids instabilities caused when high speeds are reached. Due to this, we forced the OS4 controllers to operate in a narrow range of pitch and roll angles (based on the simulations, we adopted ±0.18rad) when flying in indoor environments.

The OAC algorithm was implemented directly in the DSP. Before the OAC implementation on OS4 we carried out some experiments with the US sensors in order to get their real characteristics (visible cone shape, maximum range, set-up parameters, etc.).

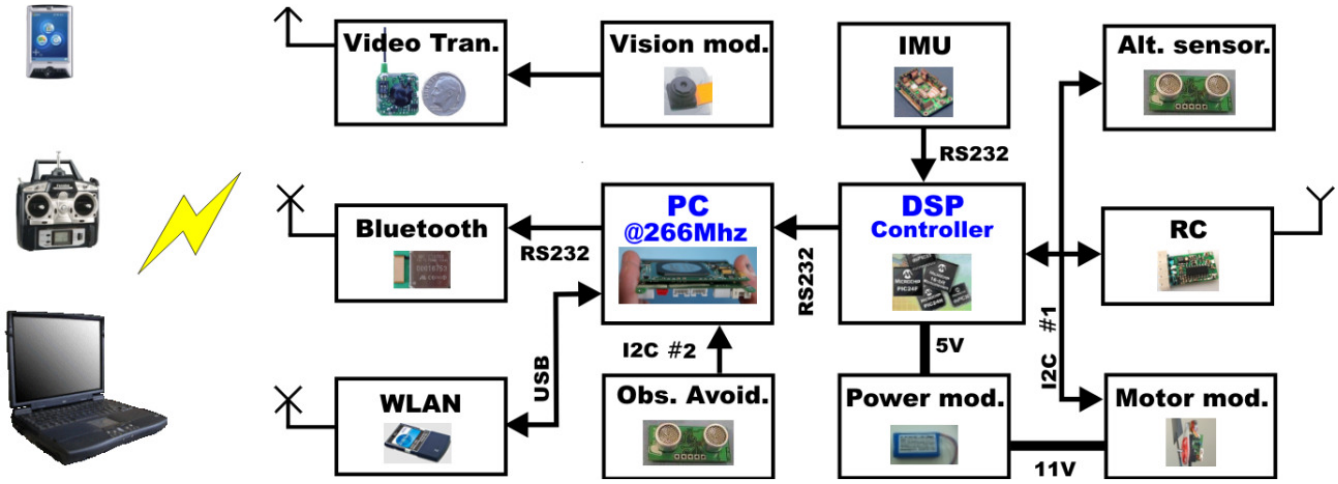


Fig. 18. OS4 block diagram. A DSP processor handles attitude and altitude control. Then, a miniature PC handles obstacles avoidance control and communication tasks. The robot communicates through a wifi interface and accepts standard remote control signals.

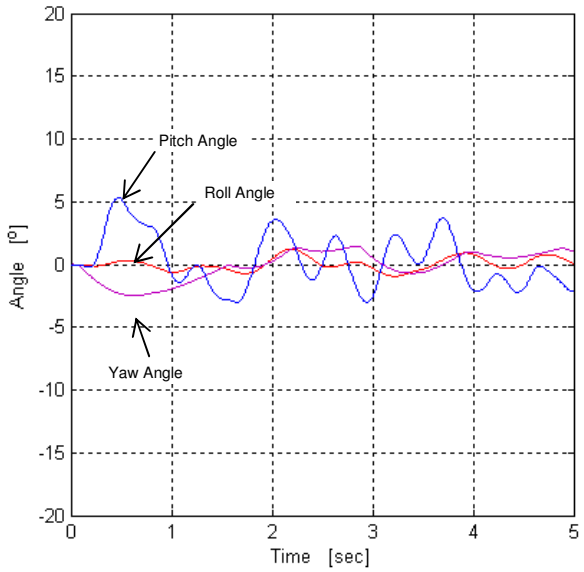


Fig. 19. OS4 Angles (degrees) during a frontwards movement simulation.

The US sensor data obtained during the tests were extremely noisy, mainly when no obstacle was in front of the sensor. On the contrary, when an obstacle was approaching the sensor, its data became stable. There are several reasons for this behavior, and the main factors are: the use of 5 US sensors simultaneously (at the same frequency) that can create disturbances; reflections due to the ground proximity; the effect of the wind caused by the propellers on the ultrasound wave; etc. Aiming to reduce the sensor noise level we added a filter. The basic premise for designing the filter was to consider the OS4 surrounds as a static environment.

Therefore, the US sensors data was taken into account only if the two last samples were sufficiently close to each other (we defined a threshold value of 15cm based on the experiments carried out before the OAC implementation). If they were not, the last value stored in memory replaced the sensor datum. This way, strong oscillations were eliminated and we could obtain a reliable signal to avoid obstacles (Fig. 20). Furthermore, safety loop was added in the algorithm to authorize the OAC only if the helicopter was at a minimal altitude. This would avoid that the helicopter crashing on the ground while flying at low altitudes and avoiding obstacles.

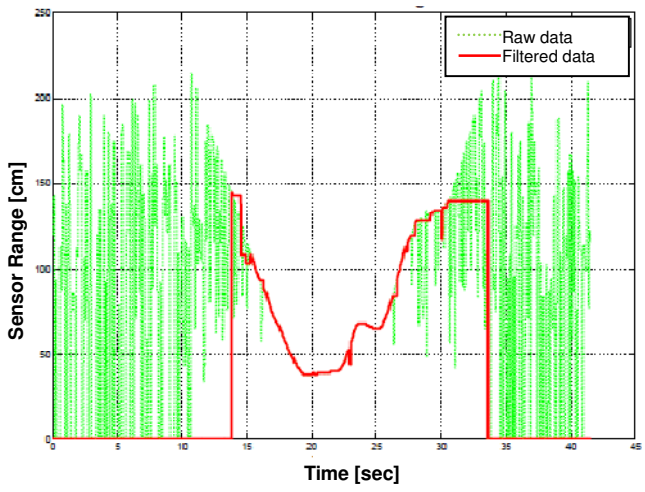


Fig. 20. OS4 US sensor data with and without filter.

Then we used the dynamic model developed in Bouabdallah and Siegwart (2005-b). Once more the speed estimation was carried out by integration, but this time we integrated the dynamic equations. We tested this procedure in the simulator and compared the exact speeds with the estimated ones. The results obtained while hovering were better. Unfortunately, they were not satisfying though. The average error was close to 0.5m/s, which was not acceptable (Bouabdallah, 2007). Concluding, we could not use the onboard sensors to estimate the OS4 speeds. Consequently, the speed-based approach proposed could not be implemented because of the lack of input data for the speed controller. This fact forced us to review the onboard sensor selection process and start to search new sensor technologies that could be used onboard OS4.

This is a complex task due to the sensor miniaturization and low power consumption needed (Bouabdallah et al., 2007). Meanwhile, we decided to implement and test the OAC only for hovering. This was carried out by adapting the safety loop previously described. In this case, the security zone radius was increased from 90cm to 1.5m and the pitch angles. Taking into account that the roll and pitch angle ranges previously tested in simulations, experimental tests confirmed that the ideal angle ranges for roll and pitch angles in indoor environments were between 0.15rad and 0.18rad (around 10°). Angle values less than 0.15rad would not produce a maneuver that was fast enough to avoid an obstacle approaching at 1m/s. Besides, angle values greater than 0.18rad would be dangerous for the helicopter in some cases when maneuvering in cluttered indoor environments. Rather than being considered a limitation on our OAC approach, these angle ranges should be regarded as a result of several simulation and experimental tests that took into account OS4 safety and typical indoor environment characteristics (cluttered areas with several corridors). If the environment consists of large areas, it is possible to increase the angle ranges and obtain faster maneuvers, without affecting the OS4 safety.

Experimental Results

Following a great number of flights and meticulous parameter settings, the experiment was finally carried out successfully: the hovering OAC and collision avoidance procedures were implemented onboard OS4. Figure 21 presents a photo-sequence of the experiment. Initially the helicopter took-off (Fig. 21-a) and assumed the hovering state (Fig. 21-b). Then, a person walks and approaches OS4 in front of US sensor #1 (Fig. 21-c). Immediately it started to avoid the person flying backwards (Fig. 21-d to Fig. 21-g). During this period it applied negative pitch angles to produce the evasive maneuvers. When the person distanced, OS4 reassumed the hovering state (Fig. 21-h). The OAC input data (US sensor nº 1 data), OS4 Altitude, Pitch, and OAC state during the experiment are shown in Fig. 22.

The observed pitch angle oscillations show that the system reacts to the OAC commands but it must also respect the stabilization constraints. As the OS4 is a high dynamic system, it was not possible to increase the OAC output angles and consequently, the evasive speed. It is however clear that the method works perfectly for people walking at moderate speeds (around 1m/s). In a near future we will implement new onboard sensors on OS4 that will allow us to test the approaches developed for cruiser speed.

Fig. 21. Photo-sequence of the experiment carried out at EPFL.

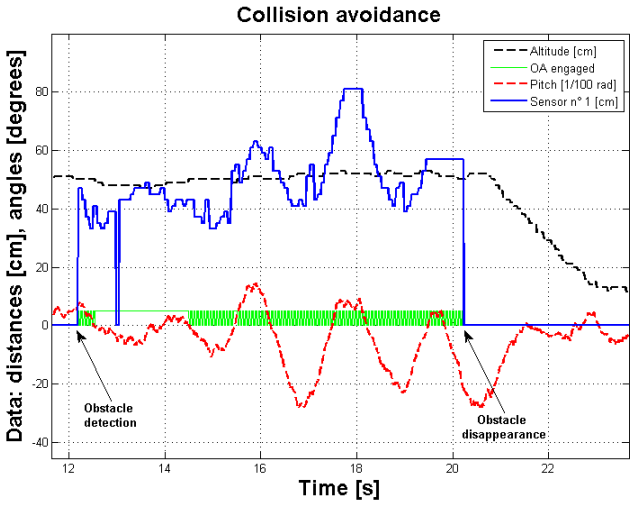


Fig. 22. Experimental results: OS4 OAC system variables. For time = 21s the helicopter started to land.

Conclusions

First of all we presented in this paper a short review of the state of art on mini-VTOL, specially the quadrotor configuration. As we highlighted previously, we could not find in the literature papers that focused on the obstacle avoidance control for mini-quadrotors. Due to this, we decided to develop a simulation tool based on MatLab/Simulink that would allow us to design and test different approaches before implementing them onboard the real helicopter. Our goal was to obtain an obstacle avoidance behavior without the use of grounded sensors and without changing the environment features. Next we briefly described the OS4 dynamical modeling and the simulator. In order to model the OS4 mini-helicopter, we used Newton-Euler formalism, model identification, and blade element and momentum theories.

The whole dynamic model was built based on physics and aerodynamics equations, and a faithful CAD model that allowed easy extraction of the physical parameters. In addition, rotor dynamics was identified in order to accurately grasp the dynamics of the brushless motor, its power electronics, the gearbox and the propeller. The implementation of an aerodynamics block allowed the consideration of variable aerodynamic coefficients that were validated in hover.

The result is a set of equations describing the vehicle dynamics not only in hover, but also in motion. A simulator was developed based on this model and is presently used in other quadrotor projects. This simulator was designed to allow the user to change easily environment features, OAC techniques, flight conditions, etc. and to visualize the results. It takes into account a complex dynamic model for the helicopter, aerodynamic effects, sensor characteristics and delays, etc. The final experiments were all performed using strictly the same parameters found by simulation.

Then, we presented navigation approaches found in literature applied to helicopters and UAVs and highlighted the lack of researches focused on real implementation of onboard navigators for mini-helicopters. Next, we introduced 2 different approaches based on position and speed controllers. Finally the implementation phase onboard the OS4 mini-quadrotor was described and the results were presented.

Due to onboard sensor limitations, we could not implement the designed cruiser speed OACs. Instead, we implemented a hovering OAC that was based on a safety loop described previously in the Obstacle Avoidance Section. The controller algorithm was simpler when compared to the other cruiser speed OACs developed. Nevertheless, it proved to be very robust and has the advantage of being compatible with a future path planner. In spite of the difficulties during the final implementation phase, the hovering OAC algorithm feasibility was indeed proven and we are planning to add new onboard sensors that will allow us to implement the cruiser speed OACs. As far as we know this was the first successful collision avoidance experiment on such systems based only on board sensors.

Extending the capabilities of OS4 requires a further improvement of the dynamics of its actuators, its sensory capability and a more integrated design. The improvement in the bandwidth of the actuators will release the power of backstepping controllers. This will allow OS4 to be more stable, to fly in more difficult environments and to enlarge its flight envelope to more complex maneuvers.

A first step in enhancing its sensory capability is to investigate optical flow or feature tracking algorithms to estimate the heading, speed and/or position onboard. The second step is to apply appropriate sensor fusion algorithms between inertial and vision sensors to have a better estimation of its state. This will allow the application of more sophisticated obstacle avoidance algorithms (e.g.: the ones described in Caldeira et al, 2007 and Vassallo et al, 2007) and navigation systems (as described in Waldmann, 2003, 2004, and 2007).

At this moment, the available stabilization and navigation algorithms require too much processing power and high resolution sensors. A potential solution is to couple insect-inspired navigation strategies with low resolution panoramic vision sensors. In addition, the demand for higher capacity batteries will never cease, and miniaturized fuel-cells represent now a promising technology. However, the real problem lies rather in energy management and conversion and not only in the source itself. Finally, the system level integration is the key to bring together all these technologies and create an optimal solution. All these challenges and possible solutions are now being investigated at ETHZ, through an european project called "muFly" (muFly, 2009) lead by the authors. This project proposes the development and implementation of the first fully autonomous micro helicopter comparable in size and weight to a small bird. Still at ETHZ, a novel vision based approach (Scaramuzza et al., 2010) using a single camera and a visual SLAM (Davison et al., 2007) for localization at unknown and unstructured environments, making possible to fly GPS-denied and unexplored scenarios.

Acknowledgments

The Authors thank the EPFL – ASL staff members for their help and kindness during this research. Marcelo Becker thanks CAPES (Grant # 0269-05-0) for the financial support during his stay at EPFL.

References

Aero-epfl (2008). [Online], accessed in Jan. 2008. Available: http://aero.epfl.ch/

Altuğ, E., Ostrowski, J.P., and Mahony, R. (2002). Control of a Quadrotor Helicopter using Visual Feedback. In: Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation.

Altuğ, E., Ostrowski, J.P., and Taylor, C.J. (2003). Quadrotor Control using Dual Camera Visual Feedback. In: Proc. of the 2003 IEEE Int. Conf. on Robotics and Automation.

Andert, F. and Goormann, L. (2007). Combined grid and feature-based occupancy map building in large outdoor environments. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS 2007, pp. 2065 – 2070.

Becker, M., Bouabdallah, S., and Siegwart, R. (2006). Development of an Obstacle Avoidance Controller for a Mini-UAV VTOL – 1st phase: Simulation (in Portuguese). In: XVI Congresso Brasileiro de Automática - CBA 2006, October 2006, Salvador - BA, Brazil.

Bae, Y. and Kim, J. (2004). Obstacle avoidance methods in the chaotic UAV. In: Proc. of The 23rd Digital Avionics Systems Conference, Vol.2.

Benallegue, A., Mokhtari, A., and Fridman, L. (2006). Feedback linearization and high order sliding mode observer for a quadrotor UAV. In: Proc. of VSS'06. International Workshop on Variable Structure Systems. pp. 365 – 372.

Besnard, L., Shtessel, Y. B., and Landrum, B. (2007). Control of a Quadrotor Vehicle Using Sliding Mode Disturbance Observer. In: Proc. of ACC '07 American Control Conference. pp. 5230 – 5235.

Bluteau, B., Briand, R., and Patrouix, O. (2006). Design and Control of an Outdoor Autonomous Quadrotor powered by a four strokes RC engine. In: Proc. of IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics. pp. 4136 – 4240.

Boivin, E., Desbiens, A., and Gagnon, E. (2008). UAV collision avoidance using cooperative predictive control. In: Proc. of 16th Mediterranean Conference on Control and Automation, pp. 682 – 688.

Bouabdallah, S., Murrieri, P., and Siegwart, R. (2004-a). Design and Control of an Indoor Micro Quadrotor. In Proc. of Int. Conf. on Robotics and Automation – ICRA 2004, New Orleans, USA.

Bouabdallah, S., Noth, A., and Siegwart, R. (2004-b). PID vs. LQ Control Techniques Applied to an Indoor Micro Quadrotor. In Proceedings of the IEEE Int. Conf. on Intelligent Robots and Systems – IROS 2004, Sendai, Japan.

Bouabdallah, S. and Siegwart, R. (2005-a). Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. In Proceedings of IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain.

Bouabdallah, S. and Siegwart, R. (2005-b). Towards Intelligent Miniature Flying Robots. In Proc. of Field and Service Robotics, Port Douglas, Australia.

Bouabdallah, S., Becker, M., and Siegwart, R. (2007). Autonomous miniature flying robots: coming soon! - Research, Development, and Results. *IEEE Robotics and Automation Society Magazine*, Vol. 14, No. 3, pp. 88-98.

Bouabdallah, S. and Siegwart, R. (2007). Full control of a quadrotor. In: Proc. of IROS 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 153 – 158.

Bouabdallah, S. (2007). Design and Control of Quadrotors with Application to Autonomous Flying. PhD thesis, Swiss Federal Institute of Technology - EPFL, Lausanne.

Bouabdallah, S., Becker, M., Perrot, V., Siegwart, R. (2007) Toward Obstacle Avoidance on Quadrotors. In: XII DINAME - International Symposium on Dynamic Problems of Mechanics, 2007, Ilhabela - SP. Proceedings of DINAME 2007, 2007. v. 1. p. 1-10.

Bouktir, Y., Haddad, M., and Chettibi, T. (2008). Trajectory planning for a quadrotor helicopter. In: Proc. of 16th Mediterranean Conference on Control and Automation, pp. 1258 – 1263.

Caldeira, E.M.O., Schneebeli, H.J.A., and Sarcinelli-Filho, M. (2007). An optical flow-based sensing system for reactive mobile robot navigation, *SBA Controle & Automação*, v.18, n.3, Natal, Jul./Sept., 2007.

Castillo, P., Dzul, A., and Lozano, R. (2004). Real-time Stabilization and Tracking of a Four-Rotor Mini Rotorcraft. *IEEE Trans. on Control Systems Technology*, Vol. 12, No. 4, pp. 510-516.

Castillo, P., Albertos, P., Garcia, P., and Lozano, R. (2006). Simple Real-time Attitude Stabilization of a Quad-rotor Aircraft with Bounded Signals. In: Proc. of 45th IEEE Conference on Decision and Control. pp. 1533 – 1538.

Cheeseman, I. and Bennett, W. (1957). The Effect of the Ground on a Helicopter Rotor in Forward Flight. Aeronautical Research Council, No. 3021.

Cheng, V.H.L. (1990). Concept development of automatic guidance for rotorcraft obstacle avoidance. *IEEE Transactions on Robotics and Automation*, Vol. 6, Issue 2, pp. 252 – 257.

Cheng, V.H.L. and Sridhar, B. (1990). Integration of Active and Passive Sensors for Obstacle Avoidance. *IEEE Control Systems Magazine*, Vol. 10, Issue 4, pp. 43 – 50.

Cheng, V.H.L. and Lam, T. (1992). Automatic guidance and control laws for helicopter obstacle avoidance. In: Proc. of IEEE International Conference on Robotics and Automation, pp. 252 – 260, vol.1.

Courbon, J., Mezouar, Y., Guenard, N., Martinet, P. (2009). "Visual navigation of a quadrotor aerial vehicle". In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, 2009, 5315 - 5320

Coza, C. and Macnab, C.J.B. (2006). A New Robust Adaptive-Fuzzy Control Method Applied to Quadrotor Helicopter Stabilization. In: Proc. of NAFIPS 2006. Annual meeting of the North American Fuzzy Information Processing Society. pp. 454 – 458.

Davison, A. J., Reid, I. D., Molton, N. D., Stasse, O., "Monoslam: Real-time single camera slam," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 6, pp. 1052–1067, 2007

Deng, X., Schenato, L., and Sastry, S.S. (2003). Attitude control for a micromechanical flying insect including thorax and sensor models. In Proc. (IEEE) International Conference on Robotics and Automation (ICRA'03), Taipei, Taiwan.

Done, G. and Balmford, D. (2001). Bramwell's Helicopter Dynamics. Oxford Butterworth-Heinemann.

Dunfied, J., Tarbouchi, M., and Labonte, G. (2004). Neural Network Based control of a Four Rotor Helicopter. In: Proc. of the 2004 IEEE Int. Conf. on Industrial Technology (ICIT).

Earl, M.G. and D'Andrea, R. (2004). Real-time Attitude Estimation Techniques applied to a Four Rotor Helicopter. In: Proc. of the 43rd IEEE Conf. on Decision and Control.

Erginer, B. and Altuğ, E. (2007). Modeling and PD Control of a Quadrotor VTOL Vehicle. In Proc. of 2007 IEEE Intelligent Vehicles Symposium. pp. 894 – 899.

Fay, G. (2001). Derivation of the aerodynamic forces for the mesicopter simulation. PhD thesis, Stanford University, USA.

Griffiths, D. and Leishman, J. G. (2002). A Study of Dual-rotor Interference and Ground Effect using a Free-vortex Wake Model. In: Proc. of the 58th Annual Forum of the American Helicopter Society.

Guenard, N., Hamel, T., and Moreau, V. (2005) Dynamic Modeling and Intuitive Control Strategy for an X4-flyer. In: Proc. of the 2005 Int. Conf. on Control and Automation.

Guenard, N., Hamel, T., and Eck, L. (2006). Control Laws for the tele operation of an unmanned aerial vehicle known as X4-flyer. In: Proc. of IROS 2006.

Hamel, T., Mahony, R., and Chriette, A. (2002). "Visual servo trajectory tracking for a four rotor VTOL aerial vehicle". In: Proc. of the IEEE Int. Workshop on Robot and Human Interactive Communication.

He, Z., Iyer, R.V., and Chandler, P.R. (2006). "Vision-based UAV flight control and obstacle avoidance". In: Proc. of American Control Conference, 5pp.

Hoffmann, G., Rajnarayan, D. G., Waslander, S. L., Dostal, D., Soon Jang, J., and Tomlin, C. J. (2004). "The Stanford testbed of autonomous rotorcraft for multi agent control (starmac)". In Proc. 23rd Digital Avionics Systems Conference (DASC'04), Salt Lake City, USA.

Hrabar, S. (2008). "3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs". In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS 2008, pp. 807 – 814.

Hrabar, S., Sukhatme, G.S., Corke, P., Usher, K., and Roberts, J. (2005). "Combined optic-flow and stereo-based navigation of urban canyons for a UAV". In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems – IROS 2005, pp. 3309 – 3316

Huang, H., Hoffmann, G. M., Waslander, S. L., Tomlin, C. J. (2009). "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering". In: Proceedings - IEEE International Conference on Robotics and Automation, 2009, 3277 - 3282

Kanade, T., Amidi, O., and Ke, Q. (2004). "Real-time and 3D vision for autonomous small and micro air vehicles". In: Proc. of 43rd IEEE Conference on Decision and Control, pp. 1655 – 1662, Vol.2.

Kim, J., Kang, M.-S., Park, S., (2010). "Accurate modeling and robust hovering control for a quad-rotor VTOL aircraft". Journal of Intelligent and Robotic Systems: Theory and Applications, 57, 9 - 26

Kroo, I., Prinz, F., Shantz, M., Kunz, P., Fay, G., Cheng, S., Fabian, T., and Partridge, C., 2000, "The mesicopter: A miniature rotorcraft concept - phase II interim report", Stanford University, USA.

Lee, D., Burg, T. C., Dawson, D. M., Shu, D., Xian, B., Tatlicioglu, E., (2009). "Robust tracking control of an underactuated quadrotor aerial-robot based on a parametric uncertain model". In: Proceedings - IEEE International Conference on Systems, Man and Cybernetics, 2009, 3187 - 3192

Leishman, J. G. (2006). *Principles of Helicopter Aerodynamics*. Cambridge University 2nd Ed.

Madani, T. and Benallegue, A. (2006). "Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique". In: Proc. of 45th IEEE Conference on Decision and Control. pp. 1515 – 1520.

McKerrow, P. (2004). "Modeling the Dragonflyer four-rotor helicopte"r. In: Proc. of the 2004 IEEE Int. Conf. on Robotics and Automation.

Microstrain IMU (2008), [Online], accessed in Jan. 2008, Available: http://www.microstrain.com/3dm-gx1.aspx

Mistler, V., Benallegue, A., and M'Sirdi, N. K. (2001). "Exact linearization and non-interacting control of a 4 rotors helicopter via dynamic feedback". In: Proc. of the IEEE Int. Workshop on Robot and Human Interactive Communication.

Mokhtari, A. and Benallegue, A. (2004). "Dynamic Feedback Controller of Euler Angles and Wind parameters estimation for a Quadrotor Unmanned Aerial Vehicle". In: Proceedings of the 2004 IEEE International Conference on Robotics and Automation.

muFly (2009). [Online], accessed in Jun. 2009. Available: http://www.mufly.ethz.ch/

Mullhaupt, P. (1999). "*Analysis and Control of Underactuated Mechanical Nonminimum-phase System*". Ph.D. Thesis, EPFL.

Murray, R. M., Li, Z., Sastry, S. S. (1994). "*A Mathematical introduction to Robotic Manipulation*". CRC.

Nicol, C., Macnab, C., Ramirez-Serrano, (2008). "A. Robust neural network control of a quadrotor helicopter". Canadian Conference on Electrical and Computer Engineering, 2008, 1233 - 1237

Paul, T., Krogstad, T.R., and Gravdahl, J.T. (2008). "UAV formation flight using 3D potential field". In: Proc. of 16th Mediterranean Conference on Control and Automation, pp. 1240 – 1245.

Roberts. J. F., Stirling, T. S., Zufferey, J. C., Floreano, D., "Quadrotor Using Minimal Sensing For Autonomous Indoor Flight", 3rd US-European Competition and Workshop on Micro Air Vehicle Systems (MAV07) & European Micro Air Vehicle Conference and Flight Competition (EMAV2007), 17-21 September 2007, Toulouse, France

Scaramuzza, D., Blosh, M., Weiss, S., Siegwart, R., "Vision Based MAV Navigation in Unknown and Unstructured Environments", 2010 IEEE Intl. Conference on Robotics and Automation, Anchorage, USA.

sFly, (2010). [Online], accessed in Jul. 2010. Available: http://projects.asl.ethz.ch/sfly/doku.php

SRF10 Sensor (2008), [On line], accessed in Jan. 2008, Available: http://www.robot-electronics.co.uk/htm/ srf10tech.htm

Sridhar, B. and Cheng, V.H.L. (1988). "Computer vision techniques for rotorcraft low-altitude flight". *IEEE Control Systems Magazine*, Vol.8, Issue 3, pp.59 – 61.

Stepaniak, M. J., Van Graas, F., De Haag, M. U., (2009). "Design of an electric propulsion system for a quadrotor unmanned aerial vehicle". Journal of Aircraft, 2009, 46, 1050 - 1058

Tarek, M. and Benallegue, A. (2007-a). "Backstepping control with exact 2-sliding mode estimation for a quadrotor unmanned aerial vehicle". In: Proc. of IROS 2007 - IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 141 – 146.

Tarek, M. and Benallegue, A. (2007-b). "Sliding Mode Observer and Backstepping Control for a Quadrotor Unmanned Aerial Vehicles". In: Proc. of ACC '07 American Control Conference. pp. 5887 – 5892.

Tayebi, A. and McGilvray, S. (2004). "Attitude Stabilization of a four-rotor aerial robot". In: Proc. of the 43rd IEEE Conf. on Decision and Control.

Tayebi, A. and McGilvray, S. (2006). "Attitude stabilization of a VTOL quadrotor aircraft". *IEEE Transactions on Control Systems Technology*. Vol. 14, Issue 3, pp. 562 – 571.

The EPSON website (2008). [Online], accessed in Jan. 2008. Available: http://www.epson.co.jp/

UAV-ETHZ (2008). [Online], accessed in Jan. 2008. Available: http://www.uav.ethz.ch/

Vassallo, R.F., Santos-Victor, J.A., and Schneebeli, H.J.A. (2007). "Aprendizagem por imitação através de mapeamento Visiomotor baseado em imagens omnidirecionais". *SBA Controle & Automação*, Mar 2007, vol.18, no.1, p.1-12.

Voos, H. (2006). "Nonlinear State-Dependent Riccati Equation Control of a Quadrotor UAV". IEEE International Conference on Control Applications. pp. 2547 – 2552.

Voos, H. (2007). "Nonlinear and neural network-based control of a small four-rotor aerial robot". In: Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics. pp. 1 – 6.

Xu, R. and Ozguner, U. (2006). "Sliding Mode Control of a Quadrotor Helicopter". In: Proc. of 45th IEEE Conference on Decision and Control. pp. 4957 – 4962.

Zapata, R. and Lepinay, P., (1999). "Flying among obstacles". In: Proc. of Third European Workshop on Advanced Mobile Robots, pp. 81-88.

Zelenka, R.E., Swenson, H.N., Dearing, M.G., and Hardy, G.H. (1993). "Simulation development of a forward sensor-enhanced low-altitude guidance system". In: Proc. of AIAA/IEEE Digital Avionics Systems Conference, pp. 267 – 275.

Zengin, U. and Dogan, A. (2007). "Real-Time Target Tracking for Autonomous UAVs in Adversarial Environments: A Gradient Search Algorithm". *IEEE Transactions on Robotics*, Vol.23, Issue 2, pp. 294 – 307.

Waldmann, J. (2003). "Sculling and Scrolling Effects on the Performance of Multirate Terrestrial Strapdown Navigation Algorithms". Proceedings of the 17th International Congress of Mechanical Engineering COBEM2003, São Paulo, SP, Brazil.

Waldmann, J. (2004). "A Derivation of the Computer-Frame Velocity Error Model for INS Aiding". Proceedings of the 15th Brazilian Automation Conference (Congresso Brasileiro de Automática), Gramado, RS, Brazil.

Waldmann, J. (2007). "Feedforward ins aiding: an investigation of maneuvers for in-flight alignment". *SBA Controle & Automação*, Dec 2007, vol.18, no.4, p.459-470.

Wang, X., Yadav, V., and Balakrishnan, S.N. (2007). "Cooperative UAV Formation Flying With Obstacle/Collision Avoidance". *IEEE Transactions on Control Systems Technology*, Vol.15, Issue 4, pp. 672 – 679.