

In-Flight Collision Avoidance Controller Based Only on OS4 Embedded Sensors

Marcelo Becker

becker@sc.usp.br
 ABCM Senior Member
 LabRoM – SEM – EESC – USP

Rafael Coronel B. Sampaio

rafaelc@sc.usp.br
 LabRoM – SEM – EESC – USP

Samir Bouabdallah

samir.bouabdallah@mavt.ethz.ch
 ASL – IRIS – ETHZ

Vincent de Perrot

vincent.deperrot@gmail.com
 STI – I2S – EPFL

Roland Siegwart

rsiegwart@ethz.ch
 ASL – IRIS – ETHZ

The major goal of this research was the development and implementation of a control system able to avoid collisions during the flight for a mini-quadrotor helicopter, based only on its embedded sensors without changing the environment. However, it is important to highlight that the design aspects must be seriously considered in order to overcome hardware limitations and achieve control simplification. The controllers of a UAV (Unmanned Aerial Vehicle) robot deal with highly unstable dynamics and strong axes coupling. Furthermore, any additional embedded sensor increases the robot total weight and therefore, decreases its operating time. The best balance between embedded electronics and robot operating time is desired. This paper focuses not only on the development and implementation of a collision avoidance controller for a mini-robotic helicopter using only its embedded sensors, but also on the mathematical model that was essential for the controller developing phases. Based on this model we carried out the development of a simulation tool based on MatLab/Simulink that was fundamental for setting the controllers' parameters. 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 proved to be very robust and feasible. In addition to this, the controller has the advantage of being compatible with future path planners that we are developing.

Keywords: VTOL, obstacle avoidance procedures, mobile 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.

When it comes to 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 the 1980's and 1990's 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 later on they extended the path search techniques to 3D in order to obtain a low-altitude guidance system for military helicopters.

In the end of the 1990's decade, 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 a 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 could 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 embedded 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. Due to this, today it is possible to implement a set of embedded sensors for small-size helicopters at reasonable prices. The combined embedded 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 in order to allow their embedded use. Nevertheless, thanks to the 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

Paper received 8 July 2010. Paper accepted 1 March 2012
 Technical Editor: Glauco Carin

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 a 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 constraints were taken into account. Hrabar (2008) presented a novel combination of techniques that could allow an 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.

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 13.6 cm micro-helicopter that is able to hover for 3 minutes. It is remotely operated via Bluetooth link.

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 a 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.

Based on the works referenced above, we noticed that there is a lack of micro and mini-helicopters able to navigate autonomously based only on their embedded 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.

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 ETHZ, 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. In addition to this, the sFly project (sFly, 2010) 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, 2007 and Bouabdallah, 2007). Although numerous approaches have already been developed in the field of obstacle avoidance in mobile robotics, 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, we must highlight again that 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 obstacle avoidance problem. This work presents the development and implementation of an obstacle avoidance controller (OAC) for this flying robot using four embedded ultrasound sensors (US) for detecting obstacles. Initially the quadrotor helicopter configuration and the OS4 mini-helicopter are introduced. Then, the simulation tool developed in MatLab/Simulink and the embedded sensors are presented. Next, the OS4 embedded controllers are briefly described. The obstacle avoidance controller designed is then presented in detail. After that, the implementation phase is described and the experimental results obtained are addressed, and finally the conclusions and perspectives are presented.

Nomenclature

a	= lift slope, dimensionless
A	= propeller disk area, m^2
b	= thrust coefficient, $N.s^2$
c	= propeller chord, m
C	= propulsion group cost factor, dimensionless
\bar{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
C_x	= coefficient of friction along x axis
C_y	= coefficient of friction along y axis
C_z	= coefficient of friction along z axis
D_i	= drag force for each propeller, N
d	= drag coefficient, $N.m.s^2$
F	= forces, N
F_i	= thrust force for each propeller, N
H	= hub force, N
I_{xx}	= inertia on x axis, $kg.m^2$
I_{yy}	= inertia on y axis, $kg.m^2$
I_{zz}	= inertia on z axis, $kg.m^2$
J_r	= rotor inertia, $kg.m^2$
l	= arm length, 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
x	= x position, m
x_i	= distance measured by the ultrasound sensor, m
y	= y position, m
z	= z position (altitude), m
z_d	= desired z position (altitude), m

Greek Symbols

θ_0	= pitch angle of incidence, rad
θ_w	= twist pitch, rad
θ	= pitch angle, rad
θ_d	= desired pitch angle, rad
θ_{iOA}	= desired obstacle avoidance control pitch angle, rad
ψ	= yaw angle, rad
ϕ	= roll angle, rad
ϕ_d	= desired roll angle, rad
ϕ_{iOA}	= desired obstacle avoidance control roll angle, rad
τ	= torques, N.m
σ	= solidity ratio, dimensionless
λ	= inflow ratio, dimensionless
μ	= motor advance ratio, dimensionless
ρ	= air density, kg/m ³
ω	= body angular rate, rad/s
ω_{d_i}	= desired propeller angular speed, 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 PD² 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 CCD 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.

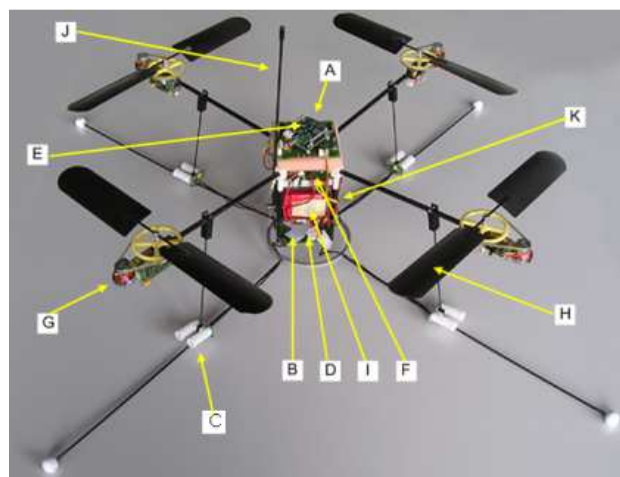


Figure 1. Photo of OS4 quadrotor mini-helicopter.

The OS4 total span is 800 mm (300 mm diameter propeller), about 200 mm in height, the four arms are tilted by 5° with respect to the horizontal plane, and its total mass is about 650 g. 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 60 W of 66 W average power consumption. However, the last one depends on flight conditions

and represents a weighted average between the equilibrium (40 W) and the worst possible inclination state (120 W) 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. Due to this, OS4 has several processing units: one processor Geode 1200 (AMD), one DSP 30F6014A (Microchip), and five microcontrollers 16F876A (Microchip). The first processing unit is a miniature computer module (CM), based on Geode 1200 processor running at 266 MHz with 128 MB of RAM and flash memory. It runs a Debian-based minimalist Linux distribution and is mainly used for high level tasks which are often not time critical. It is also used for Wi-Fi communication with the ground control station. The computer module is x86 compatible and does offer all standard PC interfaces. The whole computer is 44 g in mass, 56 mm by 71 mm in size. The DSP (30 MIPS) runs the obstacle avoidance, attitude, altitude, and position control algorithms. All the processing is carried out in less than 4 ms. One microcontroller is interfaced with the RC receiver. It was used to decode the Pulse Position Modulation (PPM) signal picked-up from a 1.6 g, 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. Finally, each of the four motor-control units has one microcontroller. All OS4 processing units are programmed in C, using standard GCC compiler for the processor and proprietary compilers for the DSP (C30) and the microcontrollers (CCS).

When it comes to monitor OS4 surrounds, we developed 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.

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

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 unstable systems, since we do not need to perform closed loop identification in flight.

Table 2. 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	$J_r \dot{\alpha} \Omega_r$
	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_x A_c \rho \dot{x} \dot{x} $
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 y axis	$-\sum_{i=1}^4 H_{y_i}$
	Friction	$\frac{1}{2} C_y A_c \rho \dot{y} \dot{y} $

* where s represents sine function and c cosine.

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. The complete list of the effects acting on our helicopter is described in Table 2 (Mullhaupt, 1999), 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. We spent an effort on developing this realistic model in order to have as reliable as possible simulation results before implementing the controllers onboard the OS4.

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:

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

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 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:

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

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:

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

where \bar{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.

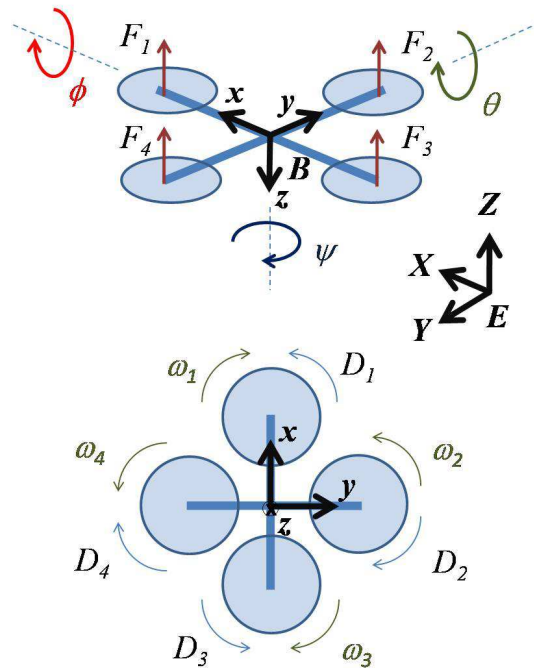


Figure 2. OS4 local coordinate system.

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

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.

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

where C_{Rm} is the rolling moment coefficient.

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

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

Another simple way to proceed is to consider that the inflow ratio in ground effect (IGE) is $\lambda_{IGE} = (v_{i,OGE} - \delta v_i - z)/\Omega R_{rad}$, 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:

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

Then we compared the variation of inflow velocity in and out of ground effect using OS4 simulator. The influence is perceptible at a ratio of z/R_{rad} approximately equal to 2, but it becomes really important around a ratio from $z/R_{rad} < 1$. 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 (Table 2). More detailed information concerning this model can be found in Appendix B at Bouabdallah (2007). The equations of motion (Eqs. (8) to (13)) are derived from Eq. (1) and all the forces and moments listed in Table 2.

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

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

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

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

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

$$m \ddot{x} = (\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi) \sum_{i=1}^4 T_i - \sum_{i=1}^4 H_{x_i} - \frac{1}{2} C_x A_c \rho v_i |x|, \quad (12)$$

$$m \ddot{y} = (-\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi) \sum_{i=1}^4 T_i - \sum_{i=1}^4 H_{y_i} - \frac{1}{2} C_y A_c \rho v_i |y|. \quad (13)$$

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 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.

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

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.

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).

OS4 Embedded Sensors

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 tested some sensors trying to find the best balance between their range, noise, power consumption, size, and weight.

When it comes to distance sensors, we have chosen US SRF10 sensors (SRF10 Sensor, 2008). It is very small and light and only detects the first echo (nearest object), taking into account a threshold value set by the user. After some tests, 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 15 Hz (acquisition every 66 ms). 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 in order to prevent the cross-talk interference. 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 do not know the exact position and the speed of the quadrotor, the obstacle itself is the reference point for evasive closed-loop maneuvers. From this perspective, it is more advantageous that the ultrasound sensors are able to accurately detect obstacles in a long distance range rather than 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 3 cm plastic tube which allowed us to increase the sensor's range of vision. Still, the length of the tube must be observed for the sensor's appropriate behavior. The volume of detection of the US sensor with and without the tubes can be evaluated from Fig. 3, whereas the dashed line represents the sensing volume without the tubes and the series of solid lines 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.

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 angle of vision, as it can be seen from Fig. 4. By its simplicity, the sensor was positioned at the intersection point of the vertical stem and the skids form (position 3 in Fig. 4). Thus, the angle of vision is around 41° (which is still broad), and free of the blades interruption. 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 to improve the quadrotor vision.

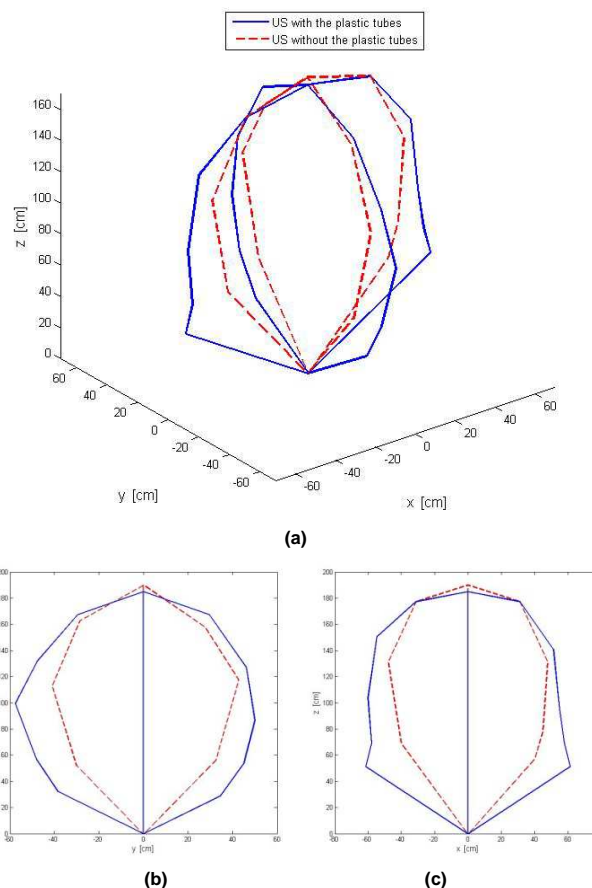


Figure 3. In (a) the volume captured by the US with the plastic tubes (dashed line) and without the plastic tubes (solid line) and in (b) and (c) details of the shapes in YZ and XZ planes, respectively.

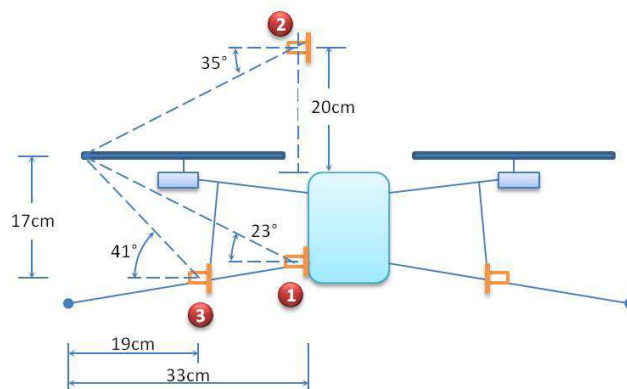


Figure 4. OS4 sketch showing all the three possible positions for the ultrasound sensor.

The main disadvantage of this set is that, at 41° , there are four dead zones in which the sensors cannot accurately detect the obstacle (Fig. 5-a). One possible solution for such problem would be to bring the sensors to positions 1 and 2 (Fig. 4). 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° for both positions. Figure 5-b shows the position of the US (already endowed with the plastic tube) in the quadrotor skids.

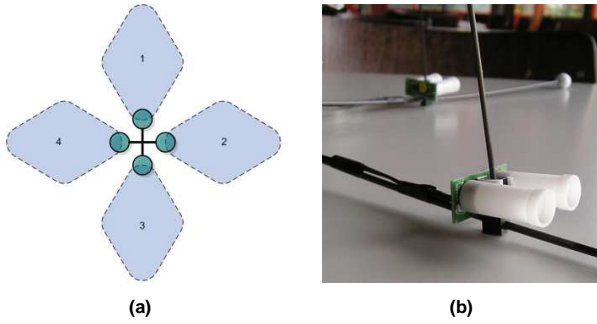


Figure 5. (a) Illustration of the OS4 US sensor fields of vision (dashed-contour areas) and dead zones and in (b), detail of the US sensor with the plastic tube assembled at the base of the quadrotor skids.

All US sensors were tested with and without the propellers rotating. OS4 was hanged in a room at 1.5 m height and we adjusted the US sensor gains in order to obtain the range previously presented in Fig. 3. With the OS4 hanged and the propellers rotating we obtained stable signs, even in presence of vibrations (the propeller speeds were set below the minimum lift force needed to take-off the helicopter only to verify the propeller wind influence on the US sensor data).

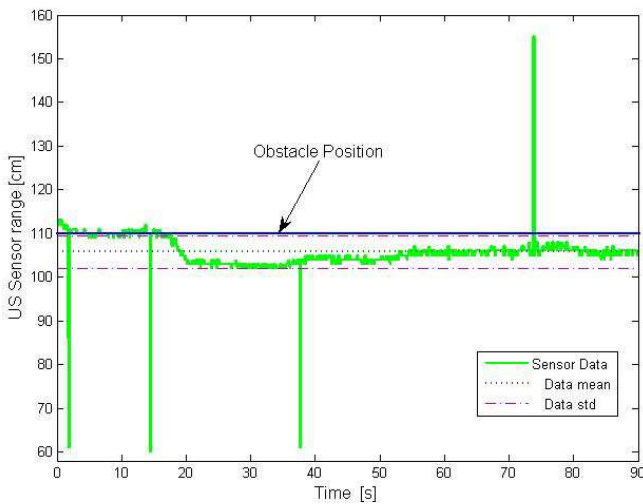


Figure 6. US sensor data obtained when OS4 was placed at 1.1 m far from an obstacle with its propellers rotating.

Figure 6 presents the results obtained when OS4 was placed at 1.1 m far from a wall that was in front of one of its US sensors. One may observe in this figure that the data were stable (solid line) and very close to the real distance (110 cm). The mean value obtained for a 90 s experiment was 105.8 cm (dotted line) with a standard deviation of 3.674 cm (dash dot lines). In this experiment we acquired data of all 5 US sensors embedded on OS4 in order to verify the cross-talk interference. Based on this experiment we concluded that there was no presence of cross-talk interference and the wind influence on US sensor data could be neglected.

Then we moved to the next experimental test phase: in-flight tests. We decided to run tests at 50 cm altitude in a large room where we could later run obstacle avoidance tests. This altitude was chosen for safety reasons because, in case of an accident, the OS4 would quickly crash against the ground and the probability of hitting someone would be low. In the beginning no obstacle was present and we expected the same US sign behavior obtained during the previous test phase. Unfortunately, the US sensor data obtained during the tests were

extremely noisy, mainly when no obstacle was in front of the sensors. On the contrary, when an obstacle was approaching the sensor, its data became stable. As we previously verified by the US sensor field of view, the effect of the wind caused by the propellers on the ultrasound wave, and the cross-talk interference, we concluded that at low altitudes the use of several US sensors simultaneously (at the same frequency) can create disturbances and reflections due to the ground proximity. Figure 7 presents the results of a 23 s test carried out without the presence of obstacles.

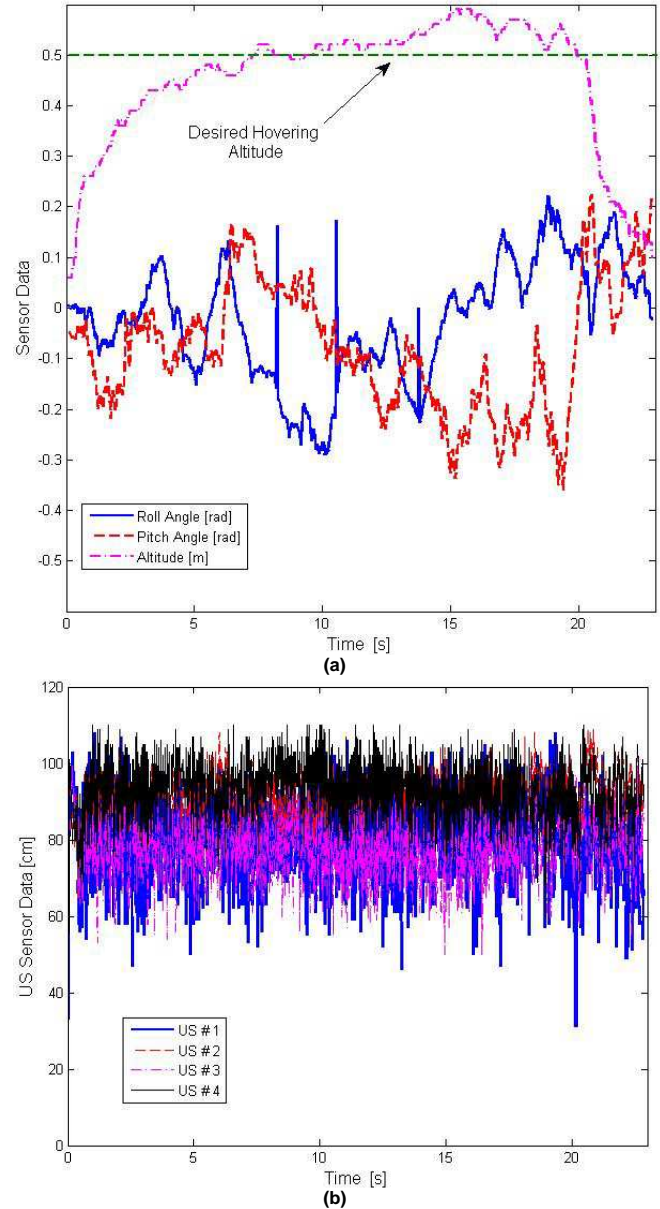


Figure 7. OS4 embedded sensor in-flight data during the experiment: take-off and hovering (at 0.5 m) maneuvers in a large room without obstacles. In (a) roll (solid line) and pitch (dashed line) angles and OS4 altitude (dash-dot line). In (b) US sensors data.

In Fig. 7-a, the desired hovering altitude (0.5 m) is represented by a dashed green line. The OS4 altitude measured by US sensor #5 was very stable (dash-dot line). In addition to this we plotted the OS4 roll (solid line) and pitch (dashed line) angles. These angles were measured by an IMU and their minimum and maximum values were

-0.29 rad and 0.22 rad (-16.6° and 12.6°) and -0.36 rad and 0.22 rad (-20.6° and 12.6°), respectively for roll and pitch angles. In addition to this, Fig. 7-b shows the noisy signs acquired by the US sensors while performing the hovering maneuver at 0.5 m in a wide room without the presence of obstacles. Then, we repeated the experiment for a 0.3 m hovering altitude. At this altitude, taking into account the US sensor volume of detection experimentally measured and presented in Fig. 3, we expected to detect the ground obtaining US sensor signs close to 0.4 m. As the results were again very noisy we calculated the mean value and the standard deviation of both experiments in order to compare them. One may observe in Fig. 7 and Table 3 that independently of the hovering altitude, roll and pitch angles, and the position of the sensors used for obstacle avoidance, the US signs were noisy and the statistics values obtained were similar. Due to this, and taking into consideration that US sensor #5 (the altitude sensor that was placed below OS4 pointing down) presented a stable behavior we concluded that the main sources of noise are the ground proximity and the US wave incidence angle that creates disturbances and reflections.

Table 3. Statistics of the in-flight tests (take-off and hovering at 0.5 and 0.3 m) without the presence of obstacles.

Test	US Sensor	Mean Value [cm]	Standard Deviation [cm]
Take-off and Hovering at 0.5 m	#1	80.70	11.36
	#2	92.13	6.45
	#3	77.42	8.39
	#4	92.89	8.65
Take-off and Hovering at 0.3 m	#1	80.31	8.66
	#2	91.62	8.32
	#3	80.54	8.96
	#4	73.33	6.80

Thus, 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 were taken into account only if the two last samples were sufficiently close to each other (we defined a threshold value of 15 cm based on the experiments carried out and the data statistics presented in Table 3). 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. One may observe an example of the filtered signs in Fig. 8. In this experiment a person first approaches up to 150 cm far from the helicopter. Then, he walks in its direction for some seconds, and finally, he turns away up to 150 cm.

In order to estimate OS4 speeds, we decided to use the Inertial Measurement Unity (IMU) embedded in the OS4. The Microstrain 3DM-GX1 (MicroStrain IMU, 2008) was assembled on the top of the OS4 embedded hardware case. It acts as an accelerometer, a gyroscope and also as a magnetometer, simultaneously. We firstly tried to use the IMU accelerations that were acquired at 75 Hz in order to estimate the OS4 speeds. The idea was to store the IMU data onboard and then numerically integrate it. However, the IMU data was extremely noisy.

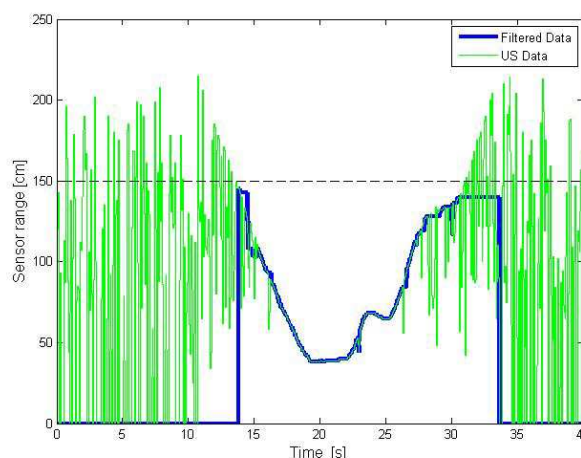


Figure 8. Distances from obstacles detected by the OS4 US sensor with and without filtering.

We tested the IMU onboard the OS4 during a 360 s hovering maneuver and we got some acceleration peaks that exceeded 15 m/s². During this experiment, the mean values for acceleration were -0.2057 m/s² and 0.2242 m/s², respectively for x and y axis. But, the standard deviation values were 4.0064 m/s² and 3.9733 m/s², respectively for x and y axis. In Fig. 7-a, we present a 6 s sample of the signal acquired during the hovering experiment. Due to this, we decided to add a filter that calculates the average value of the last 5 measurements in order to minimize the high frequency noise which was present while data was acquired. As it can be seen from Fig. 7-b, the acceleration peaks values were less than 6 m/s². In addition to this, mean values for acceleration were -0.2061 m/s² and 0.2242 m/s², respectively for x and y axis, and the standard deviation values were 1.7976 m/s² and 1.6910 m/s², respectively for x and y axis. Unfortunately, the results were not satisfying though. Even using another IMU (model MT9-B from Xsens) the signs acquired were noisy enough to make impracticable the full implementation of an Inertial Navigation Technique onboard the OS4 (it is necessary to highlight here our embedded processing power limitations, as described before; the DSP runs all obstacle avoidance, attitude, altitude, and position control algorithms). Concluding, we could not use the onboard IMU sensor to estimate the OS4 speeds. This fact forced us to review the onboard sensor selection process and start to search for new sensor technologies that could be embedded on OS4 to estimate its speeds. It is important to highlight that this is a complex task due to the sensor miniaturization and low power consumption needed (Bouabdallah et al., 2007).

The OS4 position sensor is based on an on-board pointing down CCD camera (type OV7648FB by Omnivision) and a simple pattern on the ground that acts as an artificial beacon. The camera provides a motion-blur free image of 320×240 at up to 25 fps. Most of the small cameras available on the market require high lightening condition in order to deliver motion-blur free image. The CCD camera used delivers an almost motion-blur free image. The camera is used with a red A4 paper with a white spot shifted from the pattern center. The algorithm detects the pattern, estimates the pose and provides the camera position (x , y) and heading angle (yaw angle, ψ). We used the Canny edge detector and Douglas-Peucker algorithms already implemented in OpenCV (2008). In addition, we ran a least-square based linear regression to refine the detection. Pose estimation is then performed using PnP algorithm (DeMenthon and Davis, 1995). Afterwards, the algorithm was improved in order to manage different situations where the pattern is not detected or it is only partially detected. Before embedding the camera on OS4 we

carried out some experiments fixing it on a linear motorized slider. This equipment allowed us to compare the position and yaw angle estimations obtained with the algorithms and the ones provided by its encoders. At 1 m/s the mean error was approximately 2 cm. When it comes to yaw angle (ψ), the mean error was about 3° at $180^\circ/s$.

Overall 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. When it comes to the motor speed control of the 4 propellers, a PI controller was used. Figure 9 shows an illustration of the system adopted for the attitude, altitude, position, obstacle avoidance, take-off / landing controllers, the embedded sensors, and the controllers input and output data.

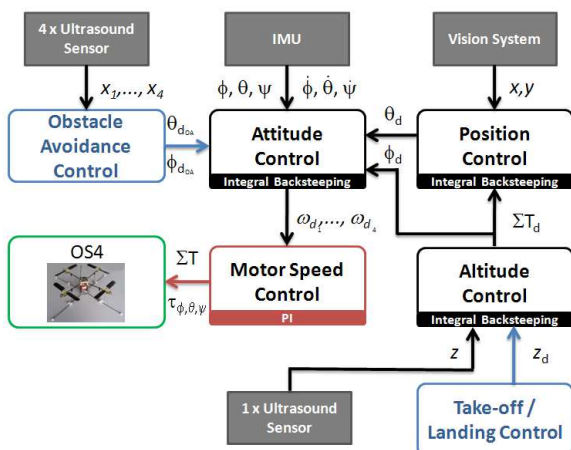


Figure 9. The proposed control architecture for OS4.

It is important to highlight that the difficulties found in OS4 control included sensor quality, yaw drift, and robustness against large disturbances and model uncertainties. As described in the previous sections, 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 our quadrotor. Thanks to this technique, OS4 has been able to perform autonomous hovering with

altitude control, obstacle avoidance maneuvers, and autonomous take-off and landing. More detailed information concerning the OS4 controllers may be found in Bouabdallah and Siegwart (2007) and Bouabdallah (2007). As the obstacle avoidance procedure is the focus of this work, this controller is detailed in the next section.

Obstacle Avoidance Controller

First of all, we decided to use our OS4 Simulator to test different approaches for obstacle avoidance before implementing them onboard the mini-helicopter. We modeled the sensors and included their delays and noises. Next, 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, for instance: hovering or keeping cruiser speed while not avoiding obstacles, landing, taking-off, etc. In this simulation phase of the development of our OAC we tested several possible solutions considering the use of the position control and a speed control that could be later on added to our control architecture (Becker et al., 2006 and Bouabdallah, 2007). Depending on the approach adopted, the OS4 yaw angle (ψ) was either kept constant or used to produce the evasive maneuver. This would enable us to perform more complex maneuvers while grazing the obstacles detected.

When we started to implement the OAC routines onboard OS4, we needed to face its sensors limitations. As the real experiments carried out with the IMU embedded on OS4 produced very noisy data, the speed estimation could not be implemented. Due to this, for the moment, we had a lack of input data for the speed controller. So, we could rely only on the OS4 control architecture and sensor data, as they were presented in Fig. 9, during the implementation of the embedded OAC. Consequently, we decided to keep the OAC as simple as possible taking into account only the reliable data provided by onboard sensors without the need of environment changes (it means that we would not use the position control because it is based on recognizing an artificial beacon placed on the ground of the environment).

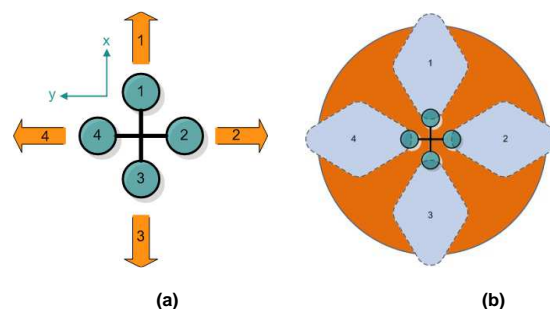


Figure 10. (a) Visualization of the 4 flight directions and (b) illustration of the 150 cm security zone around the OS4.

Aiming to simplify the procedure, we decided to keep the OS4 altitude (z) constant during the OAC maneuvers (so 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. 10-a). This would minimize the

Table 4. Rules used for the OAC.

Obstacle Quantity	Direction(s) of the Detected Obstacle(s)	Evasive Maneuver	Output Angle
1	In front	Escape Back	$\theta_d = \theta_{max}$
	Back	Escape Front	$\theta_d = -\theta_{max}$
	Left	Escape Right	$\phi_d = \phi_{max}$
	Right	Escape Left	$\phi_d = -\phi_{max}$
2	In front and Back	Escape Right	$\phi_d = \phi_{max}$
	In front and Left	Escape Back	$\theta_d = \theta_{max}$
	In front and Right	Escape Back	$\theta_d = \theta_{max}$
	Back and Left	Escape Front	$\theta_d = -\theta_{max}$
	Back and Right	Escape Front	$\theta_d = -\theta_{max}$
	Left and Right	Escape Front	$\theta_d = -\theta_{max}$
3	In front, Back, and Left	Escape Right	$\phi_d = \phi_{max}$
	In front, Back, and Right	Escape Left	$\phi_d = -\phi_{max}$
	In front, Right, and Left	Escape Back	$\theta_d = \theta_{max}$
	Back, Right, and Left	Escape Front	$\theta_d = -\theta_{max}$
4	In front, Back, Right, and Left	None - Keep Hovering	-

chance of hitting an obstacle that was not detected by the onboard US sensors while avoiding a detected obstacle. In order to increase the flight safety, a 150 cm-radius security zone is constantly maintained between the helicopter and the environment (Fig. 10-b). This security zone assures a 110 cm-distance between the helicopter rotors and any obstacle. If an obstacle is detected inside the security zone, the OAC interferes in the OS4 flight control and generates an evasive maneuver. As the US filtered sensor data provide a good estimation only of the distances between the nearest obstacles and the OS4, but no data concerning their angular position inside the US sensor field of view, the evasive maneuver is obtained by selecting a predefined pitch (ϕ) or roll (θ) angle that would avoid a collision between the helicopter and the obstacle(s). The selection of the angles depends on the direction of the detected obstacles. In the beginning we tried to implement a distance-based fuzzy controller to define the OAC output pitch (ϕ) or roll (θ) angle. But during the simulation phase we noticed that as OS4 can fly safely at 1 m/s and a person walking also moves proximately at 1 m/s, the controller would be quickly saturated. So, instead of using a fuzzy set to describe the distances and output angles, we decided to use a threshold value for the distance to define when the OAC would act. Thus, obstacle distances larger than 150 cm do not provoke any OAC reaction. On the other hand, obstacle distances less than 150 cm are considered "too close" and the OAC reacts setting the desired pitch (ϕ) or roll (θ) angle as their maximum allowed values (ϕ_{max} or θ_{max}). The values can be positive or negative depending on the desired maneuver, as shown in Table 4. As previously described, we did not allowed combinations between pitch and roll output angles, because

they could cause movements inside the US sensors blind areas and consequently a collision with an undetected obstacle could occur. So, Table 4 presents the desired OAC behavior in the presence of 1, 2, 3, and 4 obstacles. In the last case, the obstacles are blocking the helicopter. Due to this, we preferred to keep OS4 hovering, instead doing any evasive maneuver.

Furthermore, a safety loop was added in the algorithm to allow the action of the OAC only if the helicopter was at a minimal altitude. This would keep OS4 from crashing on the ground while flying at low altitudes and avoiding obstacles.

Implementation

For safety reasons, it is also possible to control OS4 using a standard radio control (RC). Taking into account that anyone who wants to implement control software embedded on 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 to 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.

Frontward, backward, and sideward 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. 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 of frontward, backward, and lateral maneuvers carried out using the OS4 Simulator previously presented, we adopted 0.18 rad (approximately 10.3°) as the maximum allowed values for pitch and roll angles (respectively, ϕ_{max} and θ_{max}) when flying in indoor environments.

The OAC algorithm was implemented directly in the DSP. After that, we carried out some experiments in order to verify the angle ranges previously obtained through the simulation tests. So, we did a series of tests in a wide room, varying the pitch and roll angles and verifying OS4 behavior. The experimental tests confirmed that the ideal angle ranges for roll and pitch angles in indoor environments were between 0.15 rad and 0.18 rad (between 8.6° and 10.3°). Angle values less than 0.15 rad would not produce a maneuver that was fast enough to avoid an obstacle approaching at 1 m/s (a person walking). Besides, angle values greater than 0.18 rad 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 obstacle avoidance experiment was finally carried out successfully: the OAC procedure was implemented onboard OS4. Figure 11 presents a photo-sequence of an experiment carried out in a wide room at EPFL. Initially the helicopter took-off (Fig. 11-a) and assumed the hovering state at 50 cm (Fig. 11-b). Then, a person walks and approaches OS4 in front of US sensor #1 (Fig. 11-c). Immediately it started to avoid the person flying backwards (Fig. 11-d to Fig. 11-g). During this period it applied negative pitch angles to produce the evasive maneuvers. When the person distanced, OS4 reassumed the hovering state (Fig. 11-h).



Figure 11. Photo-sequence of the experiment carried out at EPFL.

The OAC input data are shown in Fig. 12: OS4 altitude, pitch angle, US sensor #1 data, and the OAC state during the experiment. We presented only a detail of the experiment that shows the period when the OAC was acting. The observed pitch

angle oscillations show that the system reacts to the OAC commands, but it must also respect the stabilization constraints. Because of this, the OAC state oscillates between engaged (1) and non-engaged (0) even in presence of obstacles. If there is no stabilization issue, the OAC state holds engaged (in Fig. 12 for the time between 12.5 s and 14.5 s). As the OS4 is a high dynamic system, it was not possible to increase the OAC output angles and consequently, the evasive speed, without the risk of losing its altitude and crashing to the ground. It is, however, clear that the method works perfectly for obstacles moving at moderate speeds (around 1 m/s).

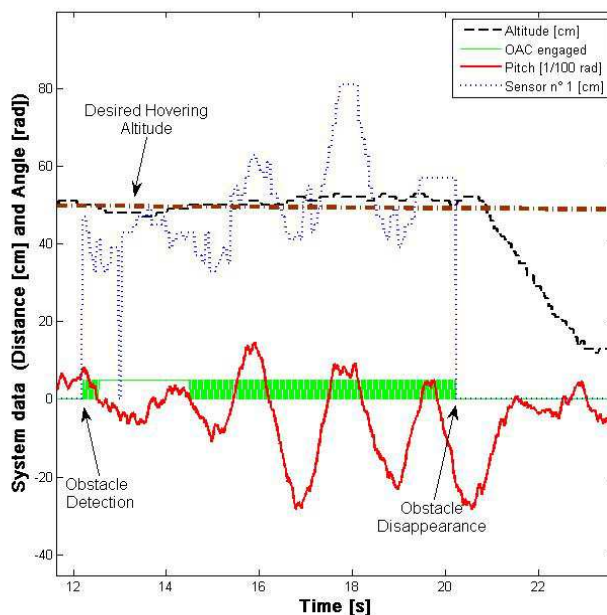


Figure 12. Detail of an in-flight test of the OS4 OAC system. OS4 was hovering at an altitude of 50 cm. For time equal to 12.2 s, a person approaches to OS4. For time equal to 20.3 s the person distanced and the US sensor stops to detect it. Finally, for time equal to 20.8 s the helicopter started to land.

Conclusions

Our goal in this research was to obtain an obstacle avoidance behavior without the use of grounded sensors and without changing the environment features. 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. 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 briefly described the OS4 dynamical modeling and the simulator developed in MatLab/Simulink. 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 was 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 easily change 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, delays, etc. The final experiments were all performed using strictly the same parameters found by simulation.

Next, we presented the OS4 embedded sensors (US, IMU, and CCD mini-camera) and described the experimental tests carried out in order to get the sensors characteristics and limitations. Due to onboard sensor limitations, we could not use a speed controller (extremely noisy IMU data) and needed to design a filter for the US sensors (because of the reflections caused by the ground proximity). Then, we described the OAC developed, and the tests done to set the controller parameters (maximum output angles). Finally, the implementation phase onboard the OS4 mini-quadrotor was described and the results were presented.

The controller algorithm was simpler when compared to the other speed-based OACs that we 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 OAC algorithm feasibility was indeed proven and we are planning to add new onboard sensors that will allow us to implement the speed-based OACs. As far as we know this was the first successful collision avoidance experiment on such systems (mini-quadrotors) based only on board sensors. The simulation tools based on MatLab/Simulink was extremely important in order to save time and reduce the quantity of experiments needed to set the control parameters.

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.

Acknowledgements

The Authors thank the EPFL – ASL staff members for their help and kindness during this research, and the Brazilian Agencies: CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and CNPq (National Council for Scientific and Technological Development).

References

- Aero-epfl, 2008, [Online], accessed in Jan. 2008. Available: <http://aero.epfl.ch/>
- Altug, 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.
- Altug, 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, Vol. 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.
- 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, Vol. 1, pp. 252-260.
- 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, pp. 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.
- DeMenthon, D. and Davis, L., 1995, "Model-based object pose in 25 lines of code". In: *International Journal of Computer Vision*, Vol. 15, No. 1-2, pp. 123-141.
- 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.

- Dunfield, 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". Ph.D 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 Chriet, 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, 5 p.
- 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, pp. 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, Vol. 2, pp. 1655-1662.
- 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, pp. 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, pp. 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 helicopter". 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.muflly.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, pp. 1233-1237.
- OpenCV, 2008, [On line], accessed in Jun 2008. Available: <http://www.intel.com/research/mrl/research/opencv/>.
- 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., 2007, "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., Blosch, M., Weiss, S., Siegwart, R., 2010, "Vision Based MAV Navigation in Unknown and Unstructured Environments", 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/html/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*, 46, pp. 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/>
- 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.
- 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.