Proceedings of the XII International Symposium on Dynamic Problems of Mechanics (DINAME 2007),
Varoto, P. S. and Trindade, M. A. (Editors), ABCM, Ilhabela, SP, Brazil, February 26 - March 2, 2007

Toward Obstacle Avoidance on Quadrotors

Samir Bouabdallah[1], Marcelo Becker[2], Vincent de Perrot[3], and Roland Siegwart[1]

[1] ETHZ, IRIS, ASL – Tannenstrasse, 3 – CH 8092 – Zürich, Switzerland

[2] PUC Minas, IPUC – Av. Dom José Gaspar, 500 – 30535-610 – Belo Horizonte, Brazil.

[3] EPFL – CH 1015 – Lausanne, Switzerland

Abstract: OS4 (Omnidirectional Stationary Flying OUtstretched Robot) is an electrically powered four-rotor miniature helicopter developed towards fully autonomous operation in indoor/outdoor environments. The major goal of this research is the development and implementation of an active control system for a quadrotor helicopter. However, one has to consider seriously the design aspect in order to get rid of the hardware limitations and achieve control simplification. Controlling a VTOL (Vertical Taking-Off and Landing) flying robot is basically dealing with highly unstable dynamics and strong axes coupling. In addition to this, any additional on-board sensor increases the robot total weight and therefore decreases its operation time. It is necessary to find a balance between on-board electronics and robot operation time. This paper focuses the development and implementation of an obstacle avoidance controller for this miniature flying robot using four ultra-sound (US) sensors for detecting obstacles and one US sensor for controlling its altitude. In order to facilitate the controller implementation, we developed a simulation tool in Matlab/Simulink using an accurate dynamic model of OS4. This tool allowed us to simulate and improve the OS4 controllers in different modeled environments and applying different approaches. The controller real implementation was carried out for hovering state and the results were encouraging.

Keywords: obstacle avoidance, unmanned aerial vehicle, mini-helicopter, VTOL, autonomous mobile robot.

INTRODUCTION

Last decades, due to their potential use on military and civil applications, 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, e.g.: office buildings and commercial centers. MFR-VTOL can also work on search-and-rescue missions after earthquakes, 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.

The potential use of these flying robots and the challenges behind their development are attracting the scientific and the industrial community. Recently, many works in the literature outlined the MFR-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. Another interesting development was 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, 2004) is a 13.6 cm micro-helicopter 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, 2006 and UAV-ETHZ, 2006). At the Autonomous Systems Lab (ASL) we are working on a quadrotor mini-helicopter named OS4. From 2003 to 2005 many goals concerning the mechanical design and control were achieved (Bouabdallah *et al*., 2004-a and 2004-b, Bouabdallah and Siegwart, 2005-a and 2005-b). Numerous approaches have already been developed in the field of obstacle avoidance in mobile robotics. However, most of these methods are not applicable to the OS4 quadrotor because of its 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-MFR-VTOLs.

Since 2005 we have been working on the OS4 obstacle avoidance problem. Some initial results concerning simulation of OS4 behavior in indoor environments with static and mobile obstacles were presented in (Becker *et al*., 2006). This work presents the development and implementation of an obstacle avoidance controller for this miniature-flying robot using four ultra-sound (US) sensors for detecting obstacles and one 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 a simulated environment using Matlab and Simulink using an accurate dynamic model of OS4. Many approaches were considered with single and multiple vertical obstacles and the most feasible one was implemented on OS4. Next, the implementation procedure and results obtained are related, and finally the conclusion and outlook are presented.

THE QUADROTOR CONFIGURATION

The helicopter quadrotor configuration is well known and has been studied since the beginning of 1900s. In 1907, the first known quadrotor helicopter, *Gyroplane No. I*, flied. It basically consists on four horizontal rotors positioned in cross configuration (Fig. 1). The rotor pair 1 and 3 rotates clockwise direction and the rotor pair 2 and 4, anticlockwise direction.

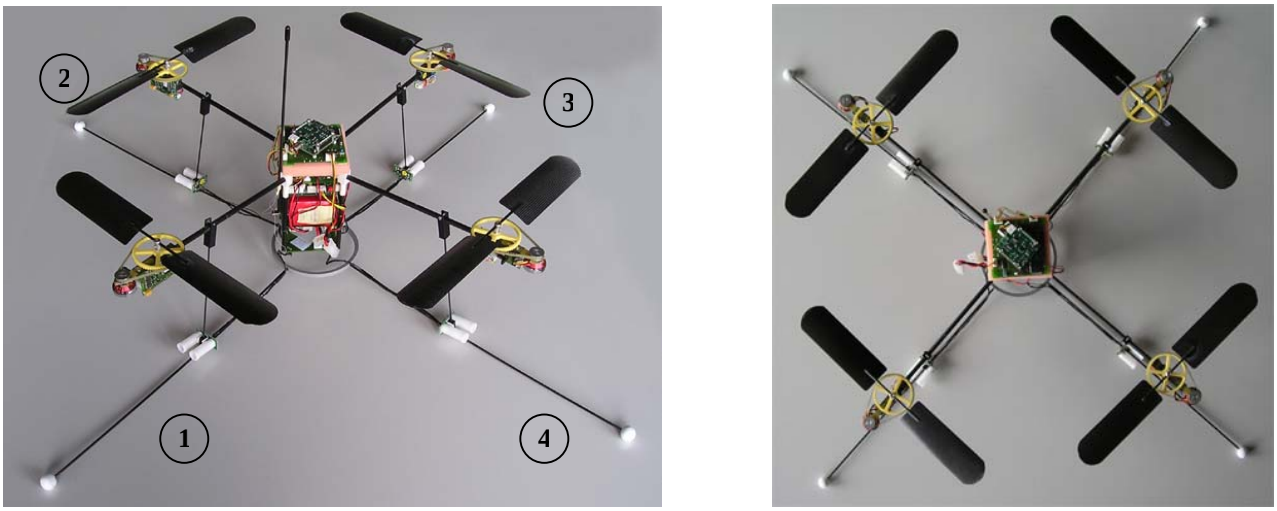
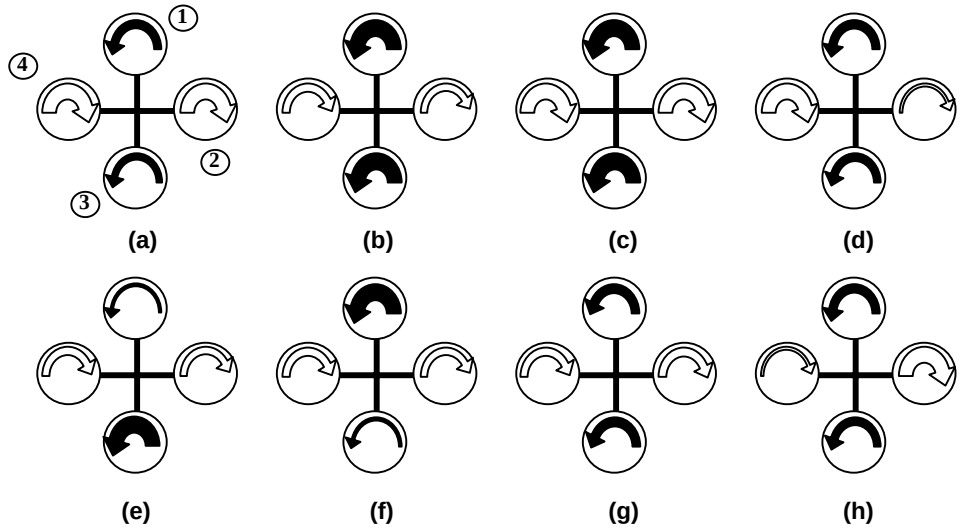


Figure 1 – Photos of the OS4 mini-helicopter quadrotor.

It is possible to maneuver the helicopter by varying its rotor angular velocities (Fig. 2). This way, by controlling the rotors it is possible to combine roll, pitch, and yaw motions in order to obtain the helicopter desired path. Like any mechanical system, the quadrotor configuration has advantages and drawbacks. However, its advantages are considerable: it allows one to increase the onboard load, simplify the rotor shape, and reduce the gyroscope effect.



(a)	Yaw (anticlockwise direction)	**(e)**	Pitch (anticlockwise direction)
(b)	Yaw (clockwise direction)	**(f)**	Pitch (clockwise direction)
(c)	Take-off or take-up	**(g)**	Land or take-down
(d)	Roll (clockwise direction)	**(h)**	Roll (anticlockwise direction)

Figure 2 – Illustration of the quadrotor helicopter motion obtained by varying its rotor angular velocities.

Today several research groups are working on MFR-VTOL using the quadrotor configuration. Mistler *et al.* (2001) proposed a non-linear dynamic model and a retro-feeding controller. Altuğ *et al*. related the use of visual retro-feeding using one (Altuğ *et al*., 2002) 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 using Lyapunov functions. Mokhtari and Benallegue (2004) developed a non-linear dynamic model based on Euler angles that associated to Lyapunov

functions controlled the helicopter roll, pitch, and yaw angles. Castillo *et al*. (2004) used the Lagrangean approach for modeling the quadrotor helicopter. The model was used together with Lyapunov functions and cyclic saturation algorithm to develop its controller. M^{c}Kerrow (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 base don data from a gyroscope and a vision system fixed on ground. Tayebi and M^{c}Gilvray (2004) proposed the use of retro-feeding controller based on quartenions 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.

The OS4 Mini-helicopter

The OS4 as a whole represents the result of the design methodology developed at ASL (Bouabdallah *et al*., 2006) and fits the desired requirements. Its total span is 800mm (300mm diameter propeller) and total mass is about 520g. Its battery (Lithium-Polymer) takes almost one-half of the total mass. The actuators, only one-third, thanks to brushless DC (BLDC) technology. They consume 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 loosing 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 128Mo of RAM and flash memory was developed. The computer module is x86 compatible and offers 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 developed. 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 the altitude sonar.

SIMULATION AND CONTROL

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's 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 in the libraries the desired combination between MFR-VTOL model, sensors, controllers and environments. 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 software is that the libraries accept the inclusion of new models for sensors, MFR-VTOLs, controllers and environments. The visualization of the results can be carried out using graphical interfaces. The Simulink model considers hub forces and rolling moments based on the literature (Done and Balmford, 2001 and Fay, 2001). 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 (Bouabdallah and Siegwart, 2005-a). 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*., 2006). Since the beginning of the OS4 project at ASL in 2003, we have developed and tested several approaches for controlling it. 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-b). 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*., 2006).

Obstacle Avoidance

The implementation of obstacle avoidance procedures on OS4 started in the middle of 2005. First of all we decided to improve the OS4 Simulator in order to verify its controller behavior while avoiding obstacles. We introduced the obstacle avoidance controller (OAC) into the Simulink model and inserted the environment and sensor libraries. 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 a 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 horizontal plane with a fixed altitude). This would reduce the path planning complexity to a 2D problem. We also restricted its direction of flight: OS4 can move only on the four directions where the US sensors were placed (Fig. 3-a). To increase the flight safety, a 90cm-radius security zone is constantly maintained between the helicopter and the environment (Fig. 3-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).

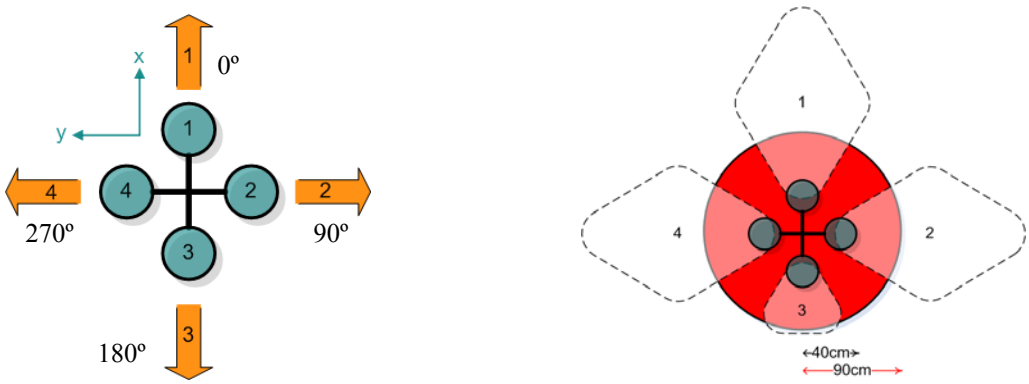


Figure 3 – Visualization of the 4 flight directions (a) and the security zone around the helicopter (b).

During the OAC simulation phase using the MatLab/Simulink tool, we developed and simulated several obstacle avoidance approaches. 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 ($\dot{x}_d$, $\dot{y}_d$). Due to this, we also developed two additional controllers: the position and the speed controllers (respectively, Fig. 4-a and 4-b). 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. In the same manner, 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.

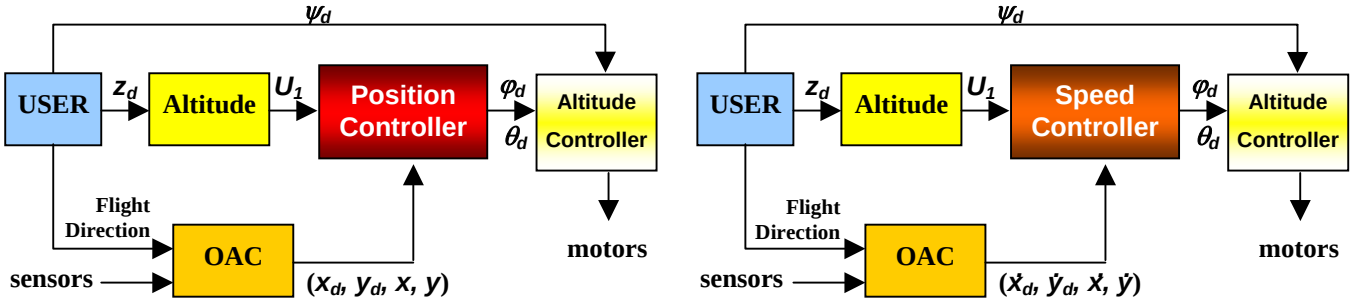


Figure 4 – Position (a) and Speed (b) Controllers developed OAC during the simulation phase.

Approach #1

This technique was based on the use of a speed controller. Initially the user selects the desired flight parameters (direction, altitude, heading, and cruiser speed). Then, the OS4 that was hovering starts to move. The OAC evaluates the environment that surrounds it and acts on its x and y speeds while keeping its heading (ψ) and altitude (z). When an obstacle is detected the distance between it and the helicopter is classified based on given threshold values as "far", "close" or "too close" (Tab. 1). If the obstacle distance is "far", no avoidance action is needed and the OAC does not interfere with the helicopter normal flight. On the other hand, if the obstacle distance is "close" the OAC informs the OS4 flight control, reduces its speed, and generates evasive maneuvers using predefined flight directions (Tab. 2). The selection of the desired flight direction to avoid the obstacle is based on the sensor that detected the obstacle and the

desired flight direction previously selected by the user. If the obstacle is "too close", the safety loop assumes the flight control. Finally, if the quadrotor is surrounded by obstacles that are "too close", it reduces the speed and assumes a hovering behavior.

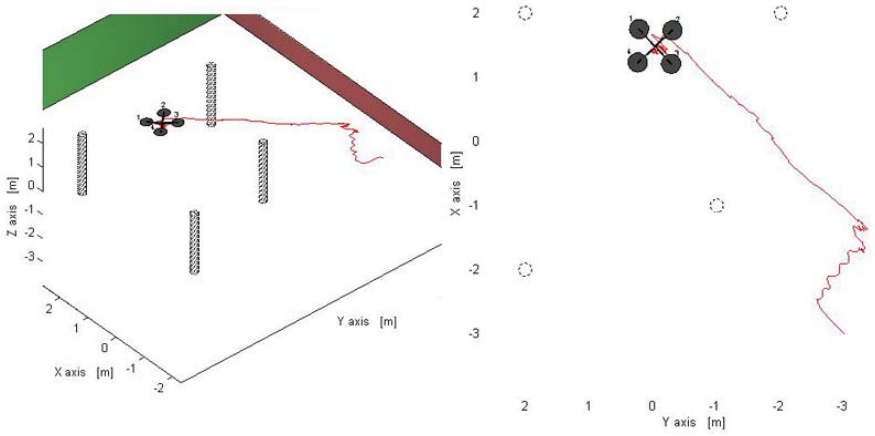
Table 1 – Threshold values that defines the linguistic variables.

Linguistic Variable	Distance (m)
"Far"	]∞; 1.8]
"Close"	]1.8; 0.9]
"Too close"	]0.9; 0]

Table 2 – Forbidden flight directions based on the sensor headings.

Sensor – "Close"	Forbidden Flight Direction	Sensor – "Too close"	Forbidden Flight Direction
#1	0º	#1	0º, 90º, and 270º
#2	90º	#2	0º, 90º, and 180º
#3	180º	#3	90º, 180º, and 270º
#4	270º	#4	0º, 180º, and 270º

Figure 5 shows a simulation of OS4 flying path in a 25m² environment with static obstacles represented as columns of 0.2m in diameter and 3m in height. Based on the simulated sensor data, the OAC was able to reduce the speed and generate the evasive maneuvers while keeping the altitude and the yaw angle stable.



(a) **(b)**

Figure 5 – OAC first technique simulation in MatLab: 3D view (a) and top view (b).

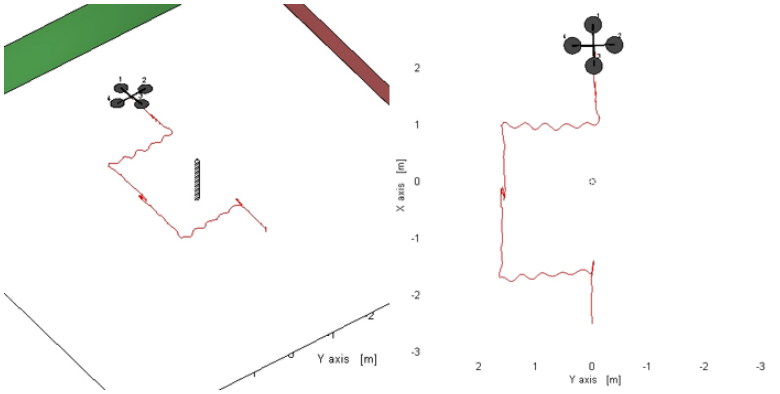
Approach #2

This approach aimed to obtain a square-shaped trajectory of the quadrotor while avoiding a single obstacle. To do so, the position controller uses the US sensor data to keep a desired relative position between the helicopter and the obstacle, while the square-shaped path is generated. If an obstacle is detected laterally, an evasive angle is generated to avoid it while trying to keep a flight direction close to the one previously selected by the user and the speed controller is active. On the other hand, a square-shaped path is generated if an obstacle is detected in front of the OS4 desired flight direction (US sensor #1). Initially the control is switched to the position controller and it is used to stabilize the helicopter in front of the obstacle. The distance read by the frontal sensor indicates the OS4 current relative position (x, y) and the desired position values (x_d, y_d) corresponds to the distance that the helicopter must keep from the obstacle. The obstacle is used to move the helicopter in closed-loop on the flight direction axis and laterally, keeping its heading, until it reaches a desired relative position from the obstacle (y_d), completing the first part of the square-shaped path. Then the helicopter moves forward (x_d) and stabilizes. Then the OS4 is detects the obstacle with the US sensor #2 and completes half of the second part of the desired path. After that it moves again forwards (x_d), stabilizes and moves laterally (y_d) to complete the evasive maneuver. Finally, the helicopter recovers its initial trajectory and cruiser speed. The resulting OS4 trajectory using the second approach is presented in Fig. 6. One disadvantage of this method lies on the US sensor limitations, e.g.: dead zone (see implementation section for details). The helicopter can lose the obstacle position after the first lateral movement and become an open-loop system.

Approach #3

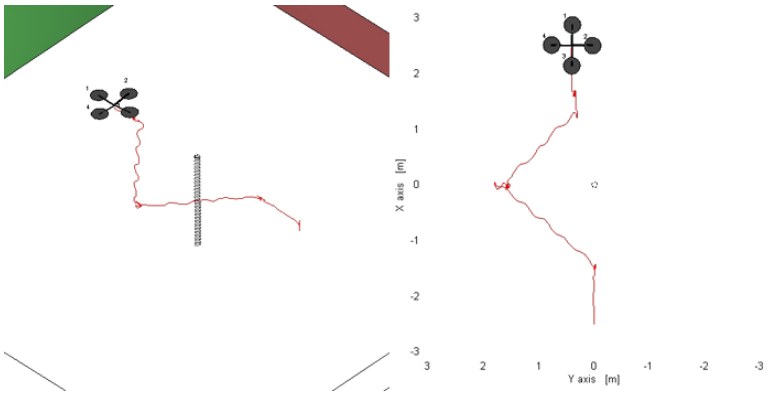
In this case we desired a triangle-shaped evasive maneuver. This version improves the preceding approach by generating a diagonal trajectory between the two first stabilization points. The generated trajectory takes the shape of a triangle. The same strategy is applied after the second movement, when the obstacle is detected by the US sensor #2 and

#3. This strategy reduces the open-loop phase probability but does not guarantee that it will never happen. Figure 7 shows the triangular path obtained.



(a) **(b)**

Figure 6 – OAC second technique simulation in MatLab: 3D view (a) and top view (b).

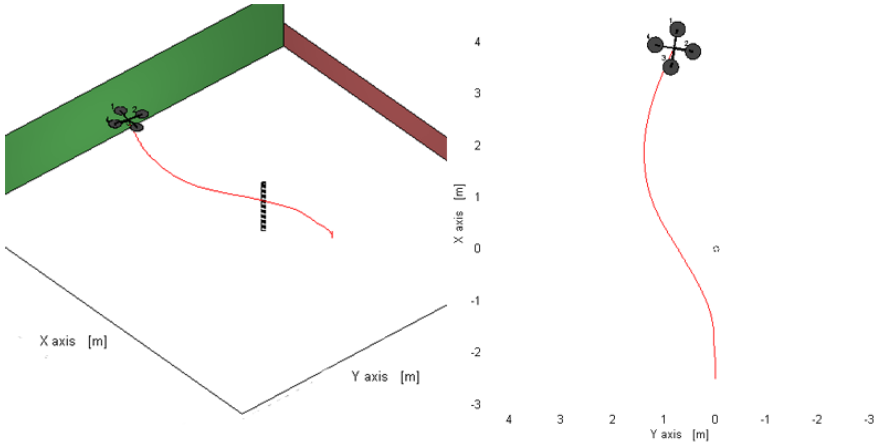


(a) **(b)**

Figure 7 – OAC third technique simulation in MatLab: 3D view (a) and top view (b).

Approach #4

In this approach, the helicopter does not stabilize when an obstacle in the flight direction is detected (US sensor #1). The OAC immediately generates a predefined negative roll angle, independently the approach speed. When the obstacle is inside the US sensor #2 visible Cone, a positive roll angle is used to finish the maneuver. The maneuver is completely carried out in open-loop (position and speed controllers are not used). This technique has the advantage of generating a continuous and smooth path but it is dangerous and random. The ideal pitch and roll angles to generate the path were obtained after successive try-and-error tests using the simulator.

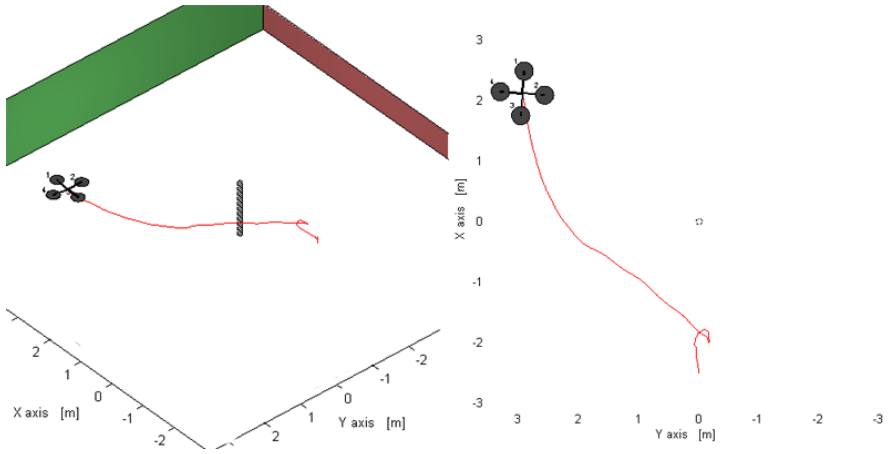


(a) **(b)**

Figure 8 – OAC fourth technique simulation in MatLab: 3D view (a) and top view (b).

Approach #5

This last approach was based on the third one. However, the helicopter does not recover its initial trajectory (Fig. 9), only the initial flight direction. After detecting the obstacle (US sensor #1), the helicopter stabilizes in its front side using the position controller. The desired pitch angle (θ_d) is controlled in closed-loop to maintain the desired distance between helicopter-obstacle, and the roll angle (φ_d) is kept at 0º. A movement in open-loop succeeds this stabilization: the pitch and the roll OS4 angles are changed to predefined values (respectively 0.06rad and -0.06rad for 2 seconds – values obtained after try-and-error tests using the simulator). The open-loop movement allows the OAC to generate the oblique trajectory to escape the obstacle. Unfortunately, due to the US sensor dead zone, the obstacle is lost for a short time. After 2s the helicopter assumes a horizontal attitude and keeps this way until the obstacle is detected by US sensor #3. The obstacle is then avoided and the OS4 continues flying in the desired flight direction.



(a) (b)

Figure 9 – OAC fifth technique simulation in MatLab: 3D view (a) and top view (b).

IMPLEMENTATION

The general architecture of the OS4 onboard system is presented in Fig. 10. The data concerning the attitude of the helicopter are 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 cycle of the DSP is synchronized with the IMU (13ms). The DSP programming was made in C. The OAC algorithm was implemented directly in the DSP.

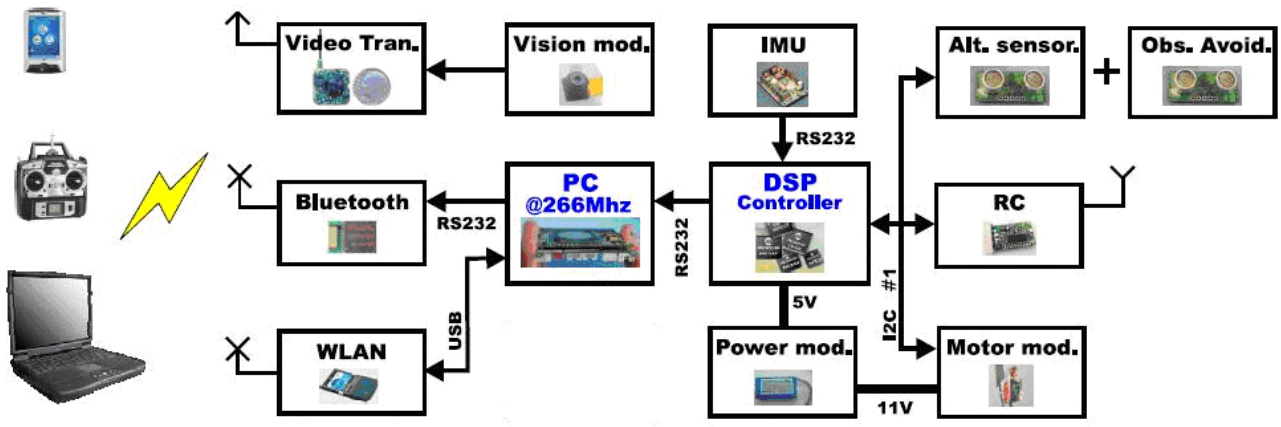
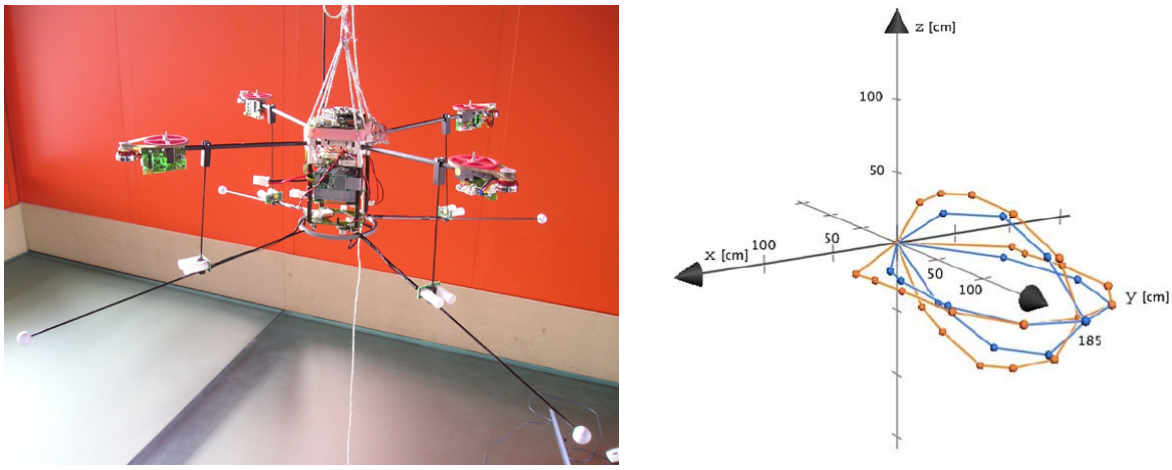


Figure 10 – OS4 onboard system architecture.

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.). US sensors used were SRF10. During these experiments, we tested different sensor assembly positions on OS4 structure and set-up parameters. 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 (Fig. 11-b). Their maximum reading frequency was 15Hz (acquisition every 65ms). We used 5 US sensors: four on OS4 structure (US sensor #1 to #4) for detecting obstacles and one vertically downwards assembled to measure its altitude (US sensor #5). Only one US sensor was fired a time.

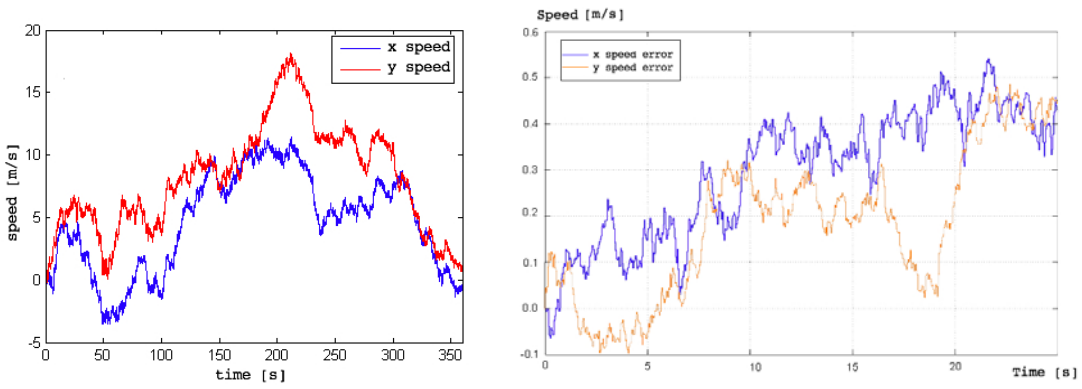


(a) **(b)**

Figure 11 – US sensor experimental test set-up – without the propellers (a) and the visible volume obtained (b): with (yellow line) and without plastic tubes (blue line).

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 extremely 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 provoke disturbs; 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 were taken into account only if the two last samples were sufficiently close to each other (we defined a threshold value of 15cm). 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. In addition to this, 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 crashed to the ground while flying at low altitudes and avoiding obstacles.

To estimate the OS4 speeds we firstly tried to use the IMU accelerations that were acquired at 76Hz (model 3DM-GX1 – MicroStrain IMU, 2006). The idea was to integrate the IMU data, but these data were extremely noisy 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) – Fig. 12-a.



(a) **(b)**

Figure 12 – OS4 speed estimation: (a) integrating the IMU acceleration and (b) using the OS4 dynamic model.

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 (Fig. 12-b). We tested this procedure in the simulator and compared the exact speeds with the estimated ones. The results obtained while hovering were better, unfortunately, not better enough. The average error was close to 0.5m/s, which was not tolerable.

Concluding, we could not use the onboard sensors to estimate the OS4 speeds. Consequently, the first 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*., 2006).

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 helicopter reacted to mobile obstacles using predefined roll and pitch angles. Simulations and experimental tests shown that the ideal angle ranges for roll and pitch angles were between 0.15rad and 0.18rad (around 10°). Angle values less than 0.15rad would not produce a maneuver fast enough to avoid an obstacle approaching at 1 m/s and angle values greater than 0.18rad would overturn the helicopter in some cases.

RESULTS

Following a great number of flights and meticulous parameter settings, the experiment was finally carried out successfully: the hovering OAC was implemented onboard OS4. Initially the helicopter took-off and assumed the hovering state. Then, a person walking approached to OS4 in front of US sensor #1. Immediately it started to avoid the person flying backwards (it used negative pitch angles to produce the evasive maneuver). When the person distanced, OS4 reassumed the hovering state. The results are shown in Fig. 13. 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.

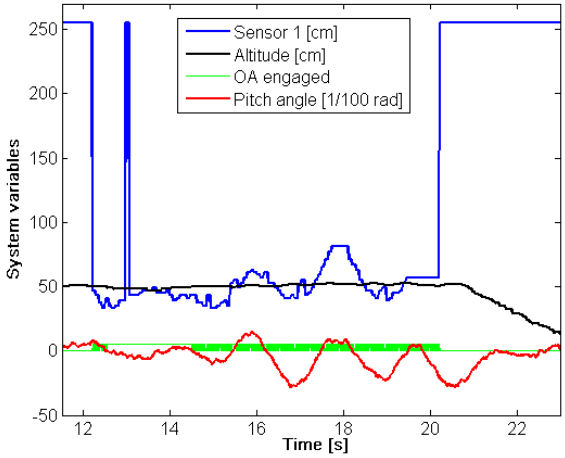


Figure 13 – OS4 OAC system variables. For Time = 21 s the helicopter started to land.

CONCLUSION

First of all we presented in this paper a short review of the state of art on VTOL-UAVs, specially the quadrotor configuration. As we highlighted on the text, 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 simulator. The 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. Then, we introduced 5 different approaches based on position and speed controllers. Simulation results for each approach were presented, as well as the comments on their advantages and drawbacks. 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, but it proved to be very robust and has the advantage of being compatible with a future path planner. In spite of the many difficulties encountered during the final implementation phase, the hovering OAC algorithm feasibility was indeed proven and we are planning to add in near future new onboard sensors that will allow us to implement the cruiser speed OACs. As far as we know this is the first successful collision avoidance experiment on such systems.

ACKNOWLEDGMENTS

Marcelo Becker wants to thank CAPES for the financial support received during his stay at EPFL, Switzerland.

REFERENCES

Aero-epfl. [Online], accessed in Jan. 2006. Available: http://aero.epfl.ch/

Altuğ, E., Ostrowski, J.P., and Mahony, R., 2002, "Control of a Quadrotor Helicopter using Visual Feedback", 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", Proc. of the 2003 IEEE Int. Conf. on Robotics and Automation.

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.

Bouabdallah, S., Becker, M., and Siegwart, R., 2006, "Autonomous Miniature Flying Robots: Coming Soon!", In: IEEE Robotics and Automation Society Magazine, *in press*.

Bouabdallah, S. and Siegwart, R., 2005-a, "Towards Intelligent Miniature Flying Robots"**,** In Proc. of Field and Service Robotics, Port Douglas, Australia.

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

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

Bouabdallah, S., Murrieri, P.,and Siegwart, R., 2004-b, "Design and Control of an Indoor Micro Quadrotor"**,** In Proc. of Int. Conf. on Robotics and Automation, New Orleans, USA.

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

Deng, X. *et al*., 2003, "Attitude control for a micromechanical flying insect including thorax and sensor models," in Proc. (IEEE) International Conference on Robotics and Automation (ICRA'03), Teipei, 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", 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", Proc. of the 43rd IEEE Conf. on Decision and Control.

Fay, G., 2001, "Derivation of the aerodynamic forces for the mesicopter simulation," Stanford University, USA.

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

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

Hoffmann, G. *et al.*, 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, Oct. 2004.

Leishman, J.G., 2000, "The Breguet-Richet Quad-Rotor Helicopter of 1907", [Online], accessed in Apr. 2006. Available: http://www.enae.umd.edu/AGRC/Aero/Breguet.pdf

Kroo, I. *et al*., 2000, "The mesicopter: A miniature rotorcraft concept - phase ii interim report," Stanford University, USA.

Marv. [Online], accessed in Jan. 2006. Available: http://www.enae.umd.edu/AGRC/Design00/MARV.html

Microstrain IMU, [On line], accessed in Jan. 2006, 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", Proc. of the IEEE Int. Workshop on Robot and Human Interactive Communication.

M^{c}Kerrow, P., 2004, "Modeling the Draganflyer four-rotor helicopter", Proc. of the 2004 IEEE Int. Conf. on Robotics and Automation.

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

SRF10 Sensor, [On line], accessed in Jan. 2006, Available: http://www.robot-electronics.co.uk/htm/srf10tech.htm

Tayebi, A. and M^{c}Gilvray, S., 2004, "Attitude Stabilization of a four-rotor aerial robot", Proc. of the 43rd IEEE Conf. on Decision and Control.

The EPSON website. [Online], accessed in Jan. 2006. Available: http://www.epson.co.jp/

UAV-ETHZ. [Online], accessed in Jan. 2006. Available: http://www.uav.ethz.ch/

RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.