Autonomous Indoor Hovering with a Quadrotor

G. Angeletti, J. R. Pereira Valente, L. Iocchi, D. Nardi

Sapienza University of Rome,
Dept. of Computer and System Science,
Via Ariosto 25, 00185 Rome, Italy
http://www.dis.uniroma1.it

Abstract. Mini and micro UAVs are very promising platforms for security and surveillance applications, because of their increased mobility in the environment. Moreover, they can effectively employed also in indoor environments. On the other hand, the limited payload for carrying sensors and the limited computational power on-board make the development of autonomous UAVs very challenging. In this paper we present hardware and software development of a quadrobot that can reliably navigate in indoor environments. In particular, we focus on the problem of indoor hovering by controlling the 6 DOF of the vehicle with different on-board and off-board sensors.

Key words: quadrotor, UAV, indoor, hovering

1 Introduction

Recently, the quadrotor has become a standard platform in the experiments and applications of *mini unmanned aerial vehicles* (mini-UAVs). The great manoeuvrability and small size of this platform make it suitable for indoor use, where the development of other kinds of UAVs is still limited. In this scenario, a quadrotor could be used in security and surveillance tasks, where the capacity of flying above ground obstacles give it a great advantage over ground robots. Indoor unsupervised flight of aerial vehicles is a hard challenge, because it is not possible to use global positioninng systems (such as GPS) as in outdoor applications. Moreover, quadrotors have usually limited payload that implies a selection of sensors that can be placed onboard. Finally, the dynamics of a flying vehicle is more complex than that of a ground robot, so that even the hovering becomes a difficult task.

In this paper, we propose a vision based and a laser based approach for the quadrotor indoor hovering. Our purpose is to propose two strategies that use different sensors in order to experiment the benefits and disadvantages of both the approaches.

2 Related Work

Among the several kinds of mini-UAVs, quadrotors are probably the most common. This platform has a “plus” shape with one rotor per corner and has been

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 472-481

widely developed by many Universities and research centers. “X3D-BL” [1], “X4-flyer” [2] and “OS4” [3] are examples of quadrotors which are entirely designed and created by a University. In these works, a customized modeling and design approach was employed, in order to create an effective platform.

Parallel to the hardware development, several efforts have been made in order to develop autmotic control systems, by considering both on-board and off-board sensors. Among others, several vision-based approaches for autonomous flight of UAVs have been studied. In these works the vision system is used to estimate the quadrotor position and orientation, in order to let it fly autonomously. Usually, a camera is placed on the quadrotor and the image processing is done on a ground station [4, 5] or directly on-board by a dedicated device [6]. The main issue in this approach concerns the (usually) bad quality images, resulting from the camera instability and signal interferences (when the image processing is done by a ground station). Altug and collaborators [7] used a two cameras architecture, with one camera placed onboard and the other one on the ground; a marker system allows the application to control the helicopter. However, a vision system presents some intrinsic problems, such as the high sensitivity to environment lighting conditions. Therefore, in some works an approach purely based on distance sensors was used. Roberts and collaborators [8] present a quadrotor that is autonomously able to take off, fly at a constant altitude and land, using only an ultrasonic sensor and four infrared sensors.

In this paper we analyze different combinations of sensors and software for controlling a quadrotor in order to maintain a stationary pose (hovering). We present both a vision-based approach, using an external camera placed on the ground and a a laser-based control using an on-board laser range finder.

3 System Architecture

3.1 Hardware

The hardware architecture is made of three distinct computational elements: the quadrotor, with its integrated control; the Gumstix boards, which provide an ARM processor, a set of input-output ports and a wi-fi connection; a ground station, used to execute the most complex computations (e.g.: run the vision algorithm). Furthermore, three different kinds of sensors were used to let the quadrotor sense the environment.

The Quadrotor The aerial platform we have used in our experiments is a quadrotor (Fig. 1) developed by *Ascending Technologies* [1]. The main features of this platform consist of a great robustness – due to the quality materials used – and an on-board 1 KHz controller, which provides a good flight stability. Moreover, the opportunity to fly the quadrotor and receive its status via a serial port available on-board, makes this platform an excellent base to start developing any autonomous behavior.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 472-481

As we said, the quadrotor offers a very-fast embedded controller, which – using data derived from a triaxial accelerometer – makes the quadrotor able to keep the roll and pitch angles commanded via the radiocontroller or the serial interface. As a consequence, if the roll and pitch stick in the radiocontroller is left centered (or, in the same way, if a roll and pitch zero angle is commanded through the serial port), the quadrotor will try to keep an horizontal orientation[1]. Since the quadrotor has not a compass, the yaw command will be an angular velocity rather than an angle; using the on-board gyroscope, the control system will try to keep the current yaw (if the yaw command is neutral), being unable to keep directly a yaw angle. However, as the datafusion is done with an update rate of 1 KHz, in the practice the quadrotor needs only occasional yaw corrections.

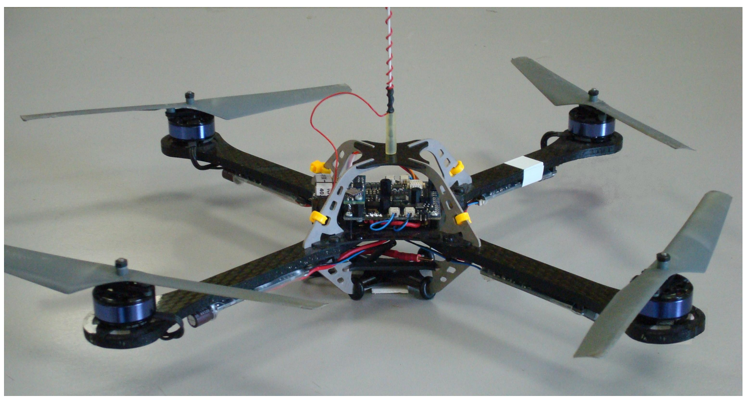

Fig. 1: The quadrotor X-3D-BL, developed by *Ascending Tecnologies*.

The Gumstix Boards In order to interact with the quadrotor electronics, three small size boards were added, named respectively: Gumstix, Robostix and Wifistix. All these boards are manufactured by Gumstix Inc., specialized in mini-boards 8 x 2 cm.

The Gumstix board provides a 400 MHz ARM processor, without the floating point unit, which can execute C and C++ code. Moreover, an optimized Linux distribution is installed on this board, so that it is possible to use all the features (e.g. multithreading) of an advanced operating system. This board can be considered as a motherboard, to which one can link many expansion cards, in order to have a wide range of extra functionalities.

In this work, we have used two expansion boards, the Robostix and the Wifistix. The first one provides a microcontroller which can drive serial (UART) ports, PWM ports, analog to digital converters and GPIO (General Purpose

[1] Obviously, due to the sensors measurement errors, the quadrotor will not keep the position.

Input Output) ports. The connection with the Gumstix is achieved via an I2C bus. The second expansion board allows, as the name “Wifistix” suggests, the use of the 802.11b/g protocol.

Since the Gumstix boards need a 5V voltage whereas the quadrotor battery provides a 11.1V voltage, a Tracopower® DC-DC voltage converter was used.

The Ground Station Due to the Gumstix hardware limitations, it is possible to execute only the simplest computations on-board. For the more demanding computations, we have used a ground station, that is a high-performance PC which keeps the communication with the quadrotor via the Wi-Fi interface.

Sensors. In order to achieve any autonomous behavior, it is necessary to have a feedback of the environment. In our work we have used three kinds of sensors: a sonar, a monocular camera and a laser. The context in which each sensor was used will be specified in the ”Implementation” section.

Onboard sensors. The ***sonar*** we decided to use is a model manufactured by MaxBotix® (Fig. 2a). This is a widely used sensor, characterized by a good resolution (2.54cm) and a very good distance range (0.16m–6.45m), with a small size package (19.9mm x 22.1mm x 16.4mm). This sensor has an onboard microcontroller that calculates the distance and converts it to an analogue voltage, PWM signal and USART. Readings can occur up to every 50ms (20Hz rate).

Considering the analog interface (the one we have used), the output consists of a voltage proportional to the measured distance, equal to $\frac{VCC}{512}$V per inch. This analog output is then converted to digital by a 10-bit Analog to Digital Converter (ADC) available on the Robostix.

The Hokuyo® URG (Fig. 2b) ***laser*** range finder has a 240° measuring angle and can range objects between 60mm and 4095mm. The accuracy is ±10mm for objects between 60mm and 1000mm, 1% of measurement between 1000mm and 4095mm. The scanning time is about 100ms (10Hz rate). There are three interfaces available: USB, RS-232C and NPN open-collector (synchronous output).

External sensors. The ***monocular camera*** used in the vision system (Fig. 2c) has in the *fisheye lens* its main feature. This lens has a very large field angle (180°), which lets the camera capture a large area; on the other hand, this lens introduces a (easy-to-correct) distortion, which is quadratic with the distance from the center of the image. The camera can be connected with a PC via a FireWire interface.

3.2 Software

In this section, an overview of the software architecture (with the interaction among components) is presented. As shown in Fig. 3, there is a single approach to control the quadrotor height, whereas two distinct strategies for roll, pitch

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 472-481

(a) MaxBotix® EZ1 sonar

(b) Hokuyo® URG laser

(c) The monocular fish-eye camera

Fig. 2: The three kinds of sensors used.

and yaw control are presented; these two strategies are discussed in depth in Section 4.2 and 4.3. A fusion of these solutions into a single one is currently under investigation.

As it is possible to see in Fig. 3, the throttle command generated by the sonar-based control goes directly to the actuators, whereas the commands coming from the laser-based control and the vision-based one pass through the quadrotor control jurisdiction (see 3.1). In summary, we have a two-layer control architecture: an inner control, with a fast cycle time, which provides a low level stabilization and an outer control, much slower than the former, which aims to keep the quadrotor flying autonomously above a desired area.

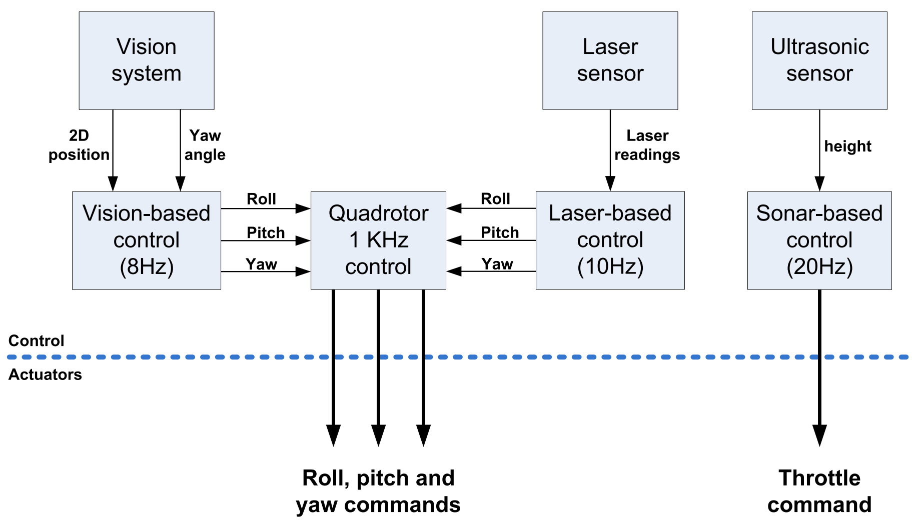

Fig. 3: Main software elements.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 472-481

4 Implementation

4.1 Altitude control through sonar

In order to control the quadrotor height, we have used an ultrasonic sensor linked to the Robostix ADC (see 3.1). This sensor was pointed at the ceiling, which present less obstacles (e.g.: chairs, tables etc.) than the ground. Data coming from the sonar are processed by a simple PI (Proportional-Integrative) control (see equation 1), which – proportionately to the gap between the current height and the desired one – produces a quadrotor throttle command. The integrative term – being a purely additive term – was added to compensate the battery dischargement, which makes the throttle command less effective.

$$u_{PI}^{(n)} = \overbrace{M_0}^{\text{0-error output}} + \overbrace{\underbrace{(K_p \cdot (h_{des}^{(n)} - h_{mis}^{(n)}))}_{\text{proportional term}} + \underbrace{(K_i \cdot \sum_{i=0}^{n} E_i)}_{\text{integrative term}}}^{\text{correction at the step n}} . \quad (1)$$

The control cycle time is 20Hz, the same of the sonar readings updating.

As shown in Fig 4, the PI control runs on the Gumstix while the Robostix manages the communication with the sonar and the quadrotor.

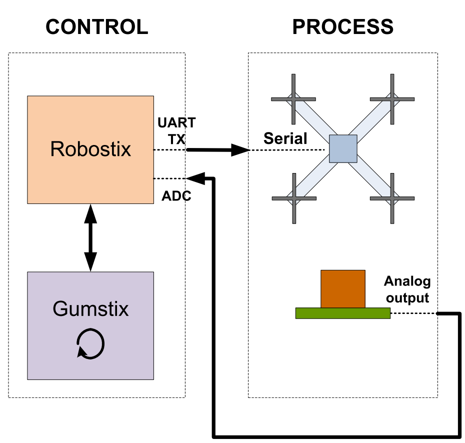

Fig. 4: Height control architecture.

4.2 Position control through off-board vision

The position and yaw control within indoor environments presents more difficulties than the outdoor one, because it is not possible to use the GPS signal to track the quadrotor position.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 472-481

As mentioned in Section 2, there are different approaches that can be used in a vision-based setting. Our approach is based on a single camera architecture. This camera (see 3.1) – placed on the ground – is connected with a PC where a vision algorithm is running. Moreover, two markers (Fig. 5) are placed just below the base and the front rotor of the quadrotor. The vision algorithm corrects the distortion introduced by the fisheye lens and extracts the position and the yaw angle of the quadrotor, relatively to the image coordinates. These information are sent via Wi-Fi to a two threads application running on the Gumstix: one thread is a server listening for data from the PC; the other one implements three (roll, pitch, yaw) PI controls (see equation 1) and communicates with the first one via shared memory. In order to let the user decide the position that the quadrotor has to keep, a graphic interface (Fig. 6) was implemented on the ground station. This graphic interface is split into two sub-windows: the left window shows the image as the camera has captured it; the right one shows the vision algorithm elaboration. In this way, the user can – pushing the “L” (as *lock!*) key on the keyboard – decide the point of the image which will be considered as a reference-point by the control application. When the “L” key is pressed, the system makes a freeze of the current position and height (considering in the same architecture the height control too) and send them – as a reference 3D point – to the controls implemented on the Gumstix.

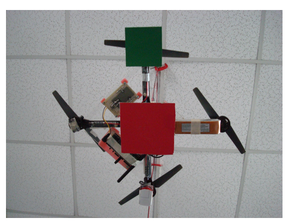

Fig. 5: Markers.

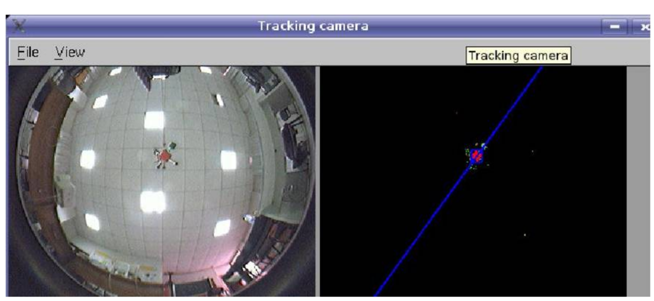

Fig. 6: The graphic interface implemented on the ground station.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 472-481

Since in this application we focus on the quadrotor position keeping, the yaw angle does not matter when the quadrotor achieves the goal. Consequently, as shown in Fig. 7, the camera image was divided in four reference angles: 0°, 90°, 180° and 270°. The quadrotor performs a starting manoeuvre to reach the closer angle, with a 45° rotation in the worst case. This manoeuvre produces a great control simplification (Fig. 8), because – with the quadrotor directed to one of this four angles – we can correct the position error separately for the coordinates x and y, as the roll corrects directly the x-axis error and the pitch the y-axis error or viceversa (it depends on the reference angle). After reaching the reference yaw angle, this one is kept by the embedded quadrotor control (see 3.1); since the quadrotor has not a compass, it can happen that the reference yaw angle is lost. Therefore, when the yaw angle measured by the vision algorithm is out of a fixed interval centered on the reference angle, the outer control is activated and the quadrotor returns within the interval.

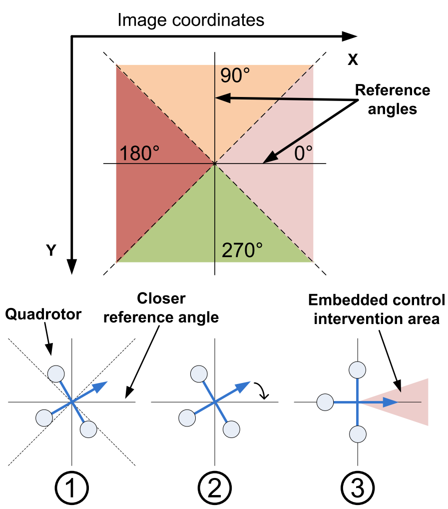

Fig. 7: An example of the quadrotor starting manoeuvre.

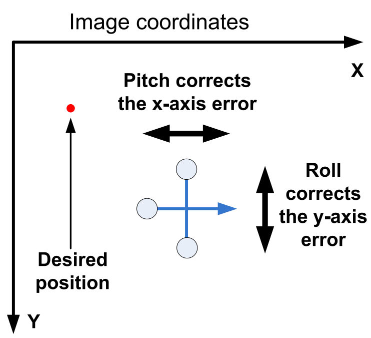

Fig. 8: An example of the reference angle strategy benefits.

The vision algorithm uses a blob extraction function to identify the markers and the Hough transform to compute lines corresponding to the axis of the quadrotor and to determine the current yaw angle. In the right window of the graphic interface (Fig. 6), the centers of the two markers and the line indicating the quadrotor direction are highlighted.

4.3 Position control through on-board laser

In this approach, the quadrotor position and heading (i.e. the yaw angle) are retrieved by using data provided by an horizontal laser range finder and a vertical

ISBN 978-88-95872-01-8
pp. 472-481

ultrasonic sonar (see Section 3.1 for details on the hardawre). In this setting, we implemented a control system for autonomous hovering, combining the sonar based height control described in Section 4.1 with a position based control based on on-board laser.

Fig. 9 shows an example of the environment. When the hovering phase starts (indicated by the human operator), the quadrotor memorizes the current laser reading and use this as a reference scan. During the hovering task, a 2D scan matching procedure (i.e., the current scan is compared to reference scan) is performed and X, Y, θ displacement with respect to the target pose is thus computed. This procedure ignores role and pitch angles of the vehicle, relying on the low-level stabilization implemented on the quadrotor.

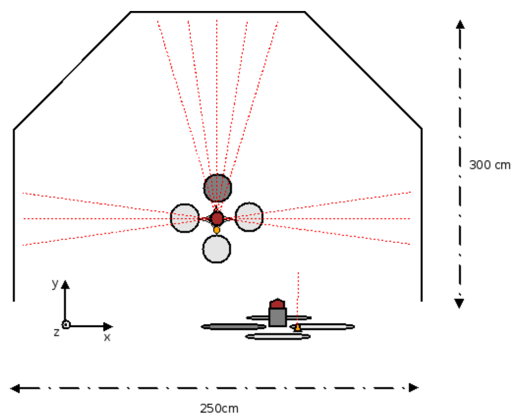

Fig. 9: Example of laser-based hovering.

The control of the vehicle is based on three controllers, one for each movement: pitch, roll and yaw.

The pitch control is responsible for the aircraft displacement on the y-axis and it is achieved through a PD control law. The roll control influences the displacement on the x-axis and it is implemented using a cascade control architecture (Fig. 10), where a simple P control was nested inside a PD control loop. Also the yaw control strategy is implemented through a standard PD control law based on the orientation estimation provided by the scan matching procedure.

5 Discussion

As already mentioned, the functionality for a quadrotor of autonomously hovering over a target is very important in many security and surveillance applications. Moreover, it is a first step towards a complete autonomous control.

The analysis and the implementation described in this paper is the first step towards a detailed experimental analysis that has the goal of evaluating the

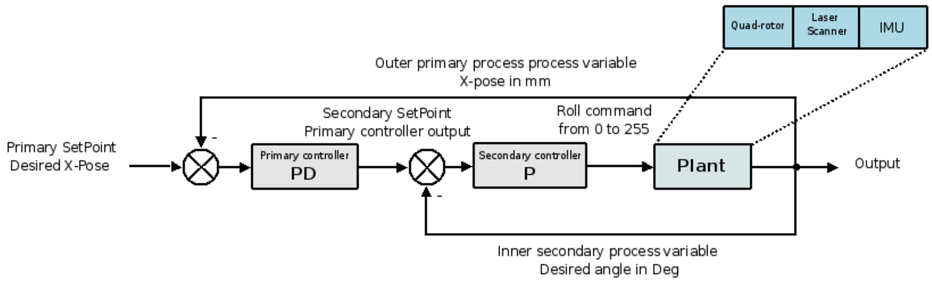

Fig. 10: Roll cascade control architecture.

different performances of the two proposed control methods, and in general the effectiveness of the proposed approaches.

Besides such an experimental analysis, future work include the study and the development of other more complex autonomous behaviors in indoor environments, like wall following, point-to-point navigation, etc.

References

1. Gurdan, D., Stumpf, J., Achtelik, M., Doth, K.M., Hirzinger, G., Rus, D.: Energy-Efficient Autonomous Four-Rotor Flying Robot Controlled at 1 KHz. In: IEEE International Conference on Robotics and Automation. Rome (2007)
2. Pound, P., Mahony, R., Hynes, P., Roberts, J.: Design of a four-rotor aerial robot. Australasian Conference on Robotics and Automation, November 2002
3. Bouabdallah, S.: Design and Control of Quadrotors with Application to Autonomous Flying. PhD thesis, Ècole Polytechnique Fédérale De Lausanne, 2007
4. Kemp, C.: Visual Control of a Miniature Quad-Rotor Helicopter. PhD thesis, University of Cambridge, 2006
5. Tournier, G.P., Valenti, M., How, J.P.: Estimation and Control of a Quadrotor Vehicle Using Monocular Vision and Moiré Patterns. Massachusetts Institute of Technology, Cambridge. AIAA Guidance, Navigation and Control Conference and Exhibit, 2006.
6. Lillywhite, K., Lee, D.J., Tippetts, B., Fowers, S., Dennis, A., Nelson, B., Archibald, J.: An embedded vision system for an unmanned four-rotor helicopter. Robotic Vision Laboratory, Department of Electrical and Computer Engineering, Brigham Young University
7. Altug, E., Taylor, C.: Vision-based pose estimation and control of a model helicopter. Istanbul Technical University and University of Pennsylvania
8. Roberts, J.F., Stirling, T.S., Zufferey, J.C., Floreano, D.: Quadrotor Using Minimal Sensing For Autonomous Indoor Flight. 3^{rd} US-European Competition and Workshop on Micro Air Vehicle Systems (MAV07) & European Micro Air Vehicle Conference and Flight Competition (EMAV2007), September 2007

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 472-481