2

The PVTOL Aircraft

2.1 Introduction

We introduce in this chapter the well-known Planar Vertical Take-Off and Landing (PVTOL) aircraft problem. The PVTOL represents a challenging nonlinear systems control problem that is a particular case of what is today known as "motion control". The PVTOL is clearly motivated by the need to stabilize aircraft that are able to take-off vertically such as helicopters and some special airplanes.

The PVTOL is a mathematical model of a flying object that evolves in a vertical plane. It has three degrees of freedom (x, y, ϕ) corresponding to its position and orientation in the plane. The PVTOL is composed of two independent thrusters that produce a force and a moment on the flying machine, see Figure 2.1. The PVTOL is an underactuated system since it has three degrees of freedom and only two inputs [144]. The PVTOL is a very interesting nonlinear control problem.

Numerous design methods for the flight control of the PVTOL aircraft model exist in the literature [24, 26, 45, 53, 92, 96]. Indeed, this particular system is a simplified aircraft model with a minimal number of states and inputs but retains the main features that must be considered when designing control laws for a real aircraft. Since, the system possesses special properties such as, for instance, unstable zero dynamics and signed (thrust) input [63], several methodologies for controlling such a system have been proposed.

An algorithm to control the PVTOL based on an approximate I-O linearization procedure was proposed in [63]. Their algorithm achieves bounded tracking and asymptotic stability. A nonlinear small gain theorem was proposed in [173] which can be used to stabilize a PVTOL. The author has proved the stability of a controller based on nested saturations [172]. The algorithm is very simple, has bounded control inputs and performs well even if the initial horizontal displacement error is large.

An extension of the algorithm proposed by [63] was presented in [100]. They were able to find a flat output of the system that was used for tracking control of the PVTOL in presence of unmodelled dynamics.

The forwarding technique developed in [102] was used in [44] to propose a control algorithm for the PVTOL. This approach leads to a Lyapunov function which ensures asymptotic stability. Other techniques based on linearization were also proposed in [43].

Recently [99] proposed a control algorithm for the PVTOL for landing on a ship whose deck oscillates. They designed an internal model-based error feedback dynamic regulator that is robust with respect to uncertainties.

In this chapter we present a global stabilizing strategy for the control of the PVTOL aircraft. The stability proof is simple. The control algorithm has been tested in numerical simulations and in real-time applications.

This chapter is organized as follows. Section 2.2 presents the PVTOL dynamical model. Section 2.3.1 gives the control of the vertical displacement while the roll and horizontal displacement control are presented in Section 2.3.2. Real-time experimental results are given in Section 2.4. The conclusions are finally given in Section 2.5.

2.2 System Description

The dynamical model of the PVTOL aircraft can be obtained using the Lagrangian approach or Newton's laws. Therefore, the PVTOL system equations are given by (see Figure 2.1)

$$\begin{aligned} \ddot{x} &= -\sin(\phi)u_1 + \varepsilon\cos(\phi)u_2 \\ \ddot{y} &= \cos(\phi)u_1 + \varepsilon\sin(\phi)u_2 - 1 \\ \ddot{\phi} &= u_2 \end{aligned} \tag{2.1}$$

where x is the horizontal displacement, y is the vertical displacement and ϕ is the angle the PVTOL makes with the horizontal line. u_1 is the total thrust and u_2 is the couple as shown in Figure 2.1. The parameter ε is a small coefficient which characterizes the coupling between the rolling moment and the lateral acceleration of the aircraft. The constant -1 is the normalized gravitational acceleration.

In general, ε is negligible and not always well known [63]. Therefore, it is possible to suppose that $\varepsilon = 0$, i.e.

$$\ddot{x} = -\sin(\phi)u_1 \tag{2.2}$$

$$\ddot{y} = \cos(\phi)u_1 - 1 \tag{2.3}$$

$$\ddot{\phi} = u_2 \tag{2.4}$$

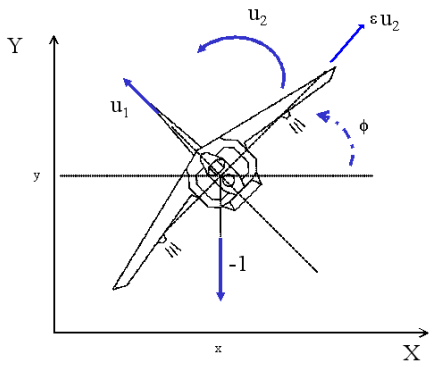

Fig. 2.1. The PVTOL aircraft (front view).

Furthermore, several authors have shown that by an appropriate change of coordinates, we can obtain a representation of the system without the term due to ε [125]. For instance, R. Olfati-Saber [125] applied the following change of coordinates

$$\bar{x} = x - \varepsilon \sin(\phi) \tag{2.5}$$
$$\bar{y} = y + \varepsilon(\cos(\phi) - 1) \tag{2.6}$$

The system dynamics considering these new coordinates become

$$\begin{aligned} \ddot{\bar{x}} &= -\sin(\phi)\bar{u}_1 \\ \ddot{\bar{y}} &= \cos(\phi)\bar{u}_1 - 1 \\ \ddot{\phi} &= u_2 \end{aligned} \tag{2.7}$$

where $\bar{u}_1 = u_1 - \varepsilon\dot{\phi}^2$. Note that this structure (2.7) has the same form as (2.1) with $\varepsilon = 0$.

2.3 Control Strategy

In this section we present a simple control algorithm for the PVTOL whose convergence analysis is also relatively simple as compared with other controllers proposed in the literature. We present a new approach based on nested saturations to control the PVTOL which can lead to further developments in nonlinear systems. The simplicity of both the algorithm and the analysis allows for a better understanding of the problem. The proposed algorithm can be considered as an extension of the control of multiple integrators of [172] to the case when there are nonlinear functions between the integrators.

This section is divided in two parts. In the first part, we are interested in stabilizing the altitude y, while in the second part we propose u_2 in order to control the roll angle and the horizontal displacement in the x-axis.

2.3.1 Control of the Vertical Displacement

The vertical displacement y will be controlled by forcing the altitude to behave as a linear system. This is done by using the following control strategy

$$u_1 = \frac{r_1 + 1}{\cos \sigma_p(\phi)} \tag{2.8}$$

where $0 < p < \frac{\pi}{2}$ and σ_η, for some $\eta > 0$, is a saturation function

$$\sigma_\eta(s) = \begin{cases} \eta & \text{for } s > \eta \\ s & \text{for } -\eta \leq s \leq \eta \\ -\eta & \text{for } s < -\eta \end{cases} \tag{2.9}$$

and

$$r_1 = -a_1 \dot{y} - a_2(y - y_d) \tag{2.10}$$

where y_d is the desired altitude and a_1 and a_2 are positive constants such that the polynomial $s^2 + a_1 s + a_2$ is stable. Let us assume that after a finite time T_2, $\phi(t)$ belongs to the interval

$$I_{\frac{\pi}{2}} = (-\frac{\pi}{2} + \epsilon, \frac{\pi}{2} - \epsilon) \tag{2.11}$$

for some $\epsilon > 0$ so that $\cos \phi(t) \neq 0$. Introducing (2.8) and (2.10) into (2.2)–(2.4) we obtain for $t > T_2$

$$\begin{aligned} \ddot{x} &= -\tan \phi (r_1 + 1) \\ \ddot{y} &= -a_1 \dot{y} - a_2(y - y_d) \\ \ddot{\phi} &= u_2 \end{aligned} \tag{2.12}$$

Note that in view of the above, $y \to y_d$ and $r_1 \to 0$ as $t \to \infty$.

2.3.2 Control of the Roll Angle and the Horizontal Displacement

We will now propose u_2 to control $\dot{\phi}, \phi, \dot{x}$ and x. The control algorithm will be obtained step by step. The final expression for u_2 will be given at the end of this section (see (2.63)). Roughly speaking, for ϕ close to zero, the (x, ϕ) subsystem is represented by four integrators in cascade. A. Teel [172] proposed a control strategy based on nested saturations that can be used to control a set of integrators connected in cascade. We will show that the strategy proposed in [172] for linear systems can be modified to control x and ϕ in the nonlinear system (2.12).

We will also show that $\phi(t) \in I_{\frac{\pi}{2}}$ in (2.11) after $t = T_2$ independently of the input u_1 in (2.8).

Boundedness of $\dot{\phi}$

In order to establish a bound for $\dot{\phi}$ define u_2 as

$$u_2 = -\sigma_a(\dot{\phi} + \sigma_b(z_1)) \tag{2.13}$$

where $a > 0$ is the desired upper bound for $|u_2|$ and z_1 will be defined later. Let

$$V_1 = \tfrac{1}{2}\dot{\phi}^2 \tag{2.14}$$

Then it follows that

$$\dot{V}_1 = -\dot{\phi}\sigma_a(\dot{\phi} + \sigma_b(z_1)) \tag{2.15}$$

Note that if $| \dot{\phi} | > b + \delta$ for some $b > 0$ and some $\delta > 0$ arbitrarily small, then $\dot{V}_1 < 0$. Therefore, after some finite time T_1, we will have

$$| \dot{\phi}(t) | \leq b + \delta \tag{2.16}$$

Let us assume that b satisfies

$$a \geq 2b + \delta \tag{2.17}$$

Then, from (2.12) and (2.13) we obtain for $t \geq T_1$

$$\ddot{\phi} = -\dot{\phi} - \sigma_b(z_1) \tag{2.18}$$

Boundedness of ϕ

To establish a bound for ϕ, define z_1 as

$$z_1 = z_2 + \sigma_c(z_3) \tag{2.19}$$

for some z_3 to be defined later and

$$z_2 = \phi + \dot{\phi} \tag{2.20}$$

From (2.18)–(2.20) we have

$$\dot{z}_2 = -\sigma_b(z_2 + \sigma_c(z_3)) \tag{2.21}$$

Let

$$V_2 = \tfrac{1}{2}z_2^2 \tag{2.22}$$

then

$$\dot{V}_2 = -z_2\sigma_b(z_2 + \sigma_c(z_3)) \tag{2.23}$$

Note that if $|z_2| > c + \delta$ for some δ arbitrarily small and some $c > 0$, then $\dot{V}_2 < 0$. Therefore, it follows that after some finite time $T_2 \geq T_1$, we have

$$|z_2(t)| \leq c + \delta \tag{2.24}$$

From (2.20) we obtain for $t \geq T_2$

$$\phi(t) = \phi(T_2)e^{-(t-T_2)} + \int_{T_2}^{t} e^{-(t-\tau)} z_2(\tau) d\tau \tag{2.25}$$

Therefore, it follows that there exists a finite time T_3 such that for $t \geq T_3 > T_2$ we have

$$|\phi(t)| \leq \bar{\phi} \triangleq c + 2\delta \tag{2.26}$$

If

$$c + 2\delta \leq \frac{\pi}{2} - \epsilon \tag{2.27}$$

then $\phi(t) \in I_{\frac{\pi}{2}}$, see (2.11), for $t \geq T_2$.

Assume that b and c also satisfy

$$b \geq 2c + \delta \tag{2.28}$$

Then, in view of (2.24), (2.21) reduces to

$$\dot{z}_2 = -z_2 - \sigma_c(z_3) \tag{2.29}$$

for $t \geq T_3$.

Note that the following inequality holds for $|\phi| < 1$

$$|\tan\phi - \phi| \leq \phi^2 \tag{2.30}$$

We will use the above inequality in the following development.

Boundedness of $\dot{x}$

To establish a bound for $\dot{x}$, let us define z_3 as

$$z_3 = z_4 + \sigma_d(z_5) \tag{2.31}$$

where z_4 is defined as

$$z_4 = z_2 + \phi - \dot{x} \tag{2.32}$$

and z_5 will be defined later. From (2.12), (2.20) and (2.29) and the above it follows that

$$\dot{z}_4 = (1 + r_1)\tan\phi - \phi - \sigma_c(z_4 + \sigma_d(z_5)) \tag{2.33}$$

Define

$$V_3 = \tfrac{1}{2} z_4^2 \tag{2.34}$$

then

$$\dot{V}_3 = z_4 \left[(1 + r_1)\tan\phi - \phi - \sigma_c(z_4 + \sigma_d(z_5))\right] \tag{2.35}$$

Since $r_1 \tan\phi \to 0$ (see (2.10) and (2.12)), there exists a finite time $T_5 > T_4$, large enough such that if

$$|z_4| > d + \bar{\phi}^2 + \delta \tag{2.36}$$

and

$$c \geq \bar{\phi}^2 + \delta \tag{2.37}$$

for some δ arbitrarily small and $d > 0$, then $\dot{V}_3 < 0$. Therefore, after some finite time $T_6 > T_5$, we have

$$|z_4(t)| \leq d + \delta + \bar{\phi}^2 \tag{2.38}$$

Let us assume that d and c satisfy

$$c \geq 2d + \delta + \bar{\phi}^2 \tag{2.39}$$

Thus, after a finite time T_6, (2.33) reduces to

$$\dot{z}_4 = (1 + r_1)\tan\phi - \phi - z_4 - \sigma_d(z_5) \tag{2.40}$$

Note that in view of (2.20), (2.32) and (2.38) it follows that $\dot{x}$ is bounded.

Boundedness of x

To establish a bound for x, let us define z_5 as

$$z_5 = z_4 + \phi - 2\dot{x} - x \tag{2.41}$$

From (2.12), (2.20), (2.32) and (2.40) we get

$$\begin{aligned}\dot{z}_5 &= (1 + r_1)\tan\phi - \phi - z_4 - \sigma_d(z_5) + \dot{\phi} + 2\tan\phi(r_1 + 1) - \dot{x} \\ &= -\sigma_d(z_5) + 3r_1\tan\phi + 3(\tan\phi - \phi)\end{aligned} \tag{2.42}$$

Define

$$V_4 = \tfrac{1}{2}z_5^2 \tag{2.43}$$

then

$$\dot{V}_4 = z_5\left[-\sigma_d(z_5) + 3r_1\tan\phi + 3(\tan\phi - \phi)\right] \tag{2.44}$$

Since $r_1\tan\phi \to 0$, there exists a finite time $T_7 > T_6$, large enough such that if $|z_5| > 3\bar{\phi}^2 + \delta$ for some δ arbitrarily small and

$$d \geq 3\bar{\phi}^2 + \delta \tag{2.45}$$

then $\dot{V}_4 < 0$. Therefore, after some finite time $T_8 > T_7$, we have

$$|z_5(t)| \leq 3\bar{\phi}^2 + \delta \tag{2.46}$$

After time T_8 (2.42) reduces to

$$\dot{z}_5 = -z_5 + 3r_1\tan\phi + 3(\tan\phi - \phi) \tag{2.47}$$

Boundedness of x follows from (2.38), (2.41) and (2.46).
Let us rewrite all the constraints on the parameters a, b, c, d and $\overline{\phi}$

$$a \geq 2b + \delta \tag{2.48}$$
$$\bar{\phi} \triangleq c + 2\delta \leq 1 \tag{2.49}$$
$$b \geq 2c + \delta \tag{2.50}$$
$$c \geq (c + 2\delta)^2 + 2d + \delta \tag{2.51}$$
$$d \geq 3(c + 2\delta)^2 + \delta \tag{2.52}$$

From the above we obtain

$$a \geq 4c + 3\delta \tag{2.53}$$
$$b \geq 2c + \delta \tag{2.54}$$
$$c + 2\delta \leq 1 \tag{2.55}$$
$$\begin{aligned} c &\geq (c + 2\delta)^2 + 2d + \delta \\ &\geq (c + 2\delta)^2 + 2(3(c + 2\delta)^2 + \delta) + \delta \\ &\geq 7(c + 2\delta)^2 + 3\delta \end{aligned} \tag{2.56}$$
$$d \geq 3(c + 2\delta)^2 + \delta \tag{2.57}$$

Convergence of ϕ, $\dot{\phi}$, x and $\dot{x}$ to zero

Therefore, c and δ should be chosen small enough to satisfy (2.53) and (2.57). The parameters a, b and d can then be computed as a function of c as above.

From (2.47) it follows that for a time large enough,

$$\mid z_5(t) \mid \leq 3\phi^2 + \delta \tag{2.58}$$

for some δ arbitrarily small. From (2.40) and (2.58) we have that for a time large enough,

$$\mid z_4(t) \mid \leq 4\phi^2 + 2\delta \tag{2.59}$$

for some δ arbitrarily small. From (2.31) and the above we have

$$\mid z_3(t) \mid \leq 7\phi^2 + 3\delta \tag{2.60}$$

Similarly, from (2.29)

$$\mid z_2(t) \mid \leq 7\phi^2 + 4\delta \tag{2.61}$$

and finally for a time large enough and an arbitrarily small δ, from (2.25) and the above we get

$$\mid \phi \mid \leq 7\phi^2 + 5\delta \tag{2.62}$$

Since δ is arbitrarily small, the above inequality implies that

$$\phi = 0$$
$$| \phi | \geq \frac{1}{7}$$

If c is chosen small enough such that $\bar{\phi} < \frac{1}{7}$ (see (2.26)), then the only possible solution is $\phi = 0$. Therefore $\phi \to 0$ as $t \to \infty$. From (2.58)–(2.61) and (2.19) we have that $z_i(t) \to 0$ for $i = 1, 2, ..., 5$. From (2.20) we get $\dot{\phi} \to 0$. From (2.32) and (2.41) it follows respectively that $\dot{x} \to 0$ and $x \to 0$. The control input u_2 is given by (2.13), (2.19), (2.20), (2.31), (2.32) and (2.41), i.e.

$$u_2 = -\sigma_a(\dot{\phi} + \sigma_b(\phi + \dot{\phi} + \sigma_c(2\phi + \dot{\phi} - \dot{x} + \sigma_d(3\phi + \dot{\phi} - 3\dot{x} - x)))) \quad (2.63)$$

The amplitudes of the saturation functions should satisfy the constraints in (2.53)–(2.57).

2.4 Real-time Experimental Results

In this section we present a real-time application of the control algorithm for the PVTOL presented in the previous sections. We have tested our control algorithm in two experiments, the first one was made using a quad-rotor rotorcraft and the second one with a prototype of the PVTOL built at the Heudiasyc laboratory.

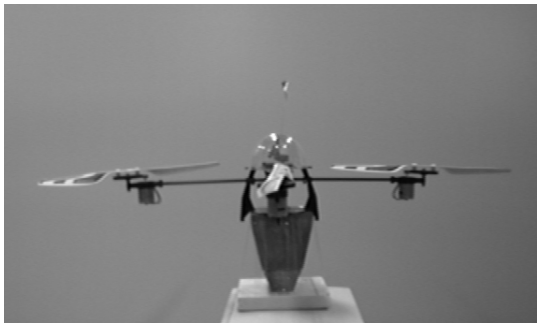

Fig. 2.2. Photograph of the quad-rotor rotorcraft (front view).

2.4.1 Experiment using a Quad-Rotor Rotorcraft

We use a four-rotor electric mini-helicopter as shown in Figure 2.2. Note that when the yaw and pitch angles are set to zero, the quad-rotor rotorcraft reduces to a PVTOL.

In our experiment the pitch and yaw angles are controlled manually by an experienced pilot. The remaining controls, i.e. the main thrust and the roll control, are controlled using the control strategy presented in the previous sections. In the four-rotor helicopter, the main thrust is the sum of the thrusts of each motor. Pitch movement is obtained by increasing (reducing) the speed of the rear motor while reducing (increasing) the speed of the front motor. The roll movement is obtained similarly using the lateral motors.

The radio is a Futaba Skysport 4. The radio and the PC (INTEL Pentium III) are connected using data acquisition cards (ADVANTECH PCL-818HG and PCL-726). The connection in the radio is directly made to the joystick potentiometers for the main thrust and pitch control.

We use the 3D tracker system (POLHEMUS) [46] for measuring the position and the orientation of the rotorcraft. The Polhemus is connected via RS232 to the PC.

In Chapter 3, we will give a detailed description of this platform.

We wish to use our control law with a quad-rotor rotorcraft; this helicopter evolves in 3D and its movements are defined by the variables $(x, y, z, \psi, \theta, \phi)$. We are going to assimilate the altitude of the quad-rotor rotorcraft to the altitude of the PVTOL. This means that we will see y of the PVTOL as z of the quad-rotor rotorcraft.

The control objective for this experience is to make the quad-rotor rotorcraft hover at an altitude of 20 cm i.e. we wish to reach the position $(x, z) = (0, 20)$ cm while $\phi = 0°$.

With respect to the time derivatives, we have simplified the mathematical computation by setting $\dot{q}_t \cong \frac{q_t - q_{t-T}}{T}$, where q is a given variable and T is the sampling period. In the experiment $T = 71$ ms.

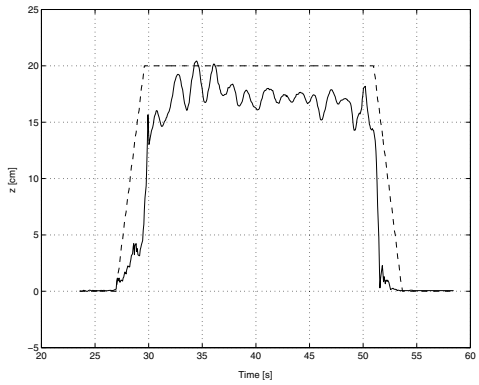

Fig. 2.3. Altitude of the quad-rotor rotorcraft.

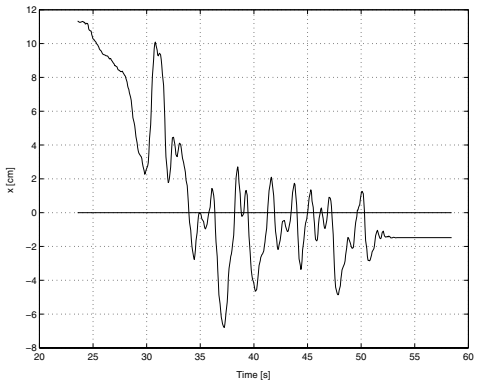

Fig. 2.4. x position of the quad-rotor rotorcraft.

Figures 2.3–2.4 show the performance of the controller when applied to the helicopter. Take-off and landing were performed autonomously. The choice of the values for a, b, c, d were carried out satisfying inequalities (2.53)−(2.57). However these parameters have been tuned experimentally in the sequence as they appear in the control input u_2.

Figure 2.3 describes the gap between the real altitude of the rotorcraft and the desired altitude. One can notice that it follows satisfactorily the desired reference. Following the vertical axis, we notice that the position error in the variable z is of about 3 to 4 cm. In Figure 2.4, one can see the gap between the real horizontal position of the quad-rotor rotorcraft (according to x) and the desired position. We note that the rotorcraft position converges towards the desired position. The final position error is about 2 to 4 cm.

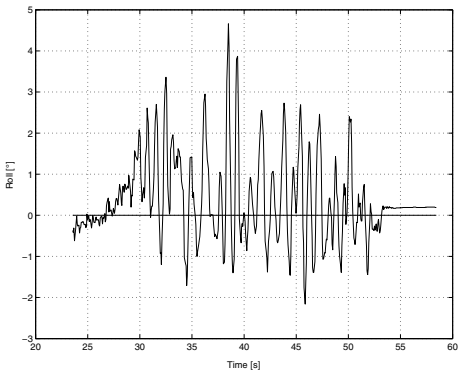

Fig. 2.5. Roll angle (ϕ) of the quad-rotor rotorcraft.

Figure 2.5 shows the evolution of the ϕ angle. It can be seen that the control law achieves convergence of the roll angle to zero, as the helicopter goes up to the altitude of 20 cm. It also shows the rotorcraft horizontal displacement in the x-axis.

2.4.2 Experimental Platform Using Vision

In this subsection we have tested the control algorithm using vision [131].

The PVTOL prototype built at Heudiasyc laboratory is shown in Figure 2.6. The rotors are driven separately by two motors. One motor rotates clockwise while the second one rotates counter-clockwise. The main thrust is the sum of the thrusts of each motor. The rolling moment is obtained by increasing (decreasing) the speed of one motor while decreasing (increasing) the speed of the second motor.

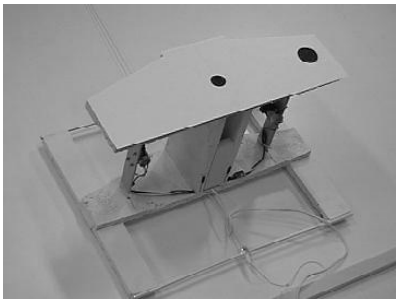

Fig. 2.6. Photograph of the PVTOL aircraft (front view).

The PVTOL moves on an inclined plane, which defines our 2D workspace. The PVTOL platform is an experimental setup designed to study the problems currently found in navigation at low altitude of a small flying object. At low altitude, GPS and even intertial navigation systems are not enough to stabilize mini-flying objects. Vision using cameras should provide additional information to make autonomous flights near the ground possible. For simplicity, at a first stage, we have placed the camera outside the aircraft. In the future, the camera will be located at the base of the mini-helicopter, pointing downwards. Note that even when the camera is located outside the flying object, we still have to deal with the problems of object localization computation using cameras and delays in the closed-loop system.

In the platform, a CCD camera Pulnix is located perpendicular to the plane at a fixed altitude and provides an image of the whole workspace. We have used an acquisition card PCI-1409 of National Instruments Company. The camera is linked to the PC dedicated to the vision part (which will be

referred as Vision PC). From the image provided by the camera, the program calculates the position (x, y) and the orientation ϕ of the PVTOL with respect to a given origin. Then, the Vision PC sends this information to another PC dedicated to the control part (which we call Control PC), via a RS232 connection, transmitting at 115 200 bps (bauds per second).

The control inputs are therefore calculated according to the proposed strategy based on saturation functions and sent to the PVTOL via the radio. In order to simplify the implementation of the control law we have designed the platform in such a way that each of the two control inputs can independently work either in automatic or in manual mode. The minimum sampling period we are able to obtain in the experimental platform is 40 ms. This includes the computation of the control law, image processing, localization computation and A/D and D/A conversion in the radio–PC interface.

Experiment

The control objective is to make the PVTOL hover at an altitude of 60 pixels i.e. we wish to reach the position $(x, y) = (0, 60)$ in pixels while $\phi = 0°$.

The measurements of x, y are expressed in pixels in the image frame and ϕ is expressed in degrees, which means that the servoing is done on the basis of image features directly.

With respect to the time derivatives, we have simplified the mathematical computation by setting $\dot{q}_t = \frac{q_t - q_{t-T}}{T}$, where q represent either x, y or ϕ and T is the sampling period.

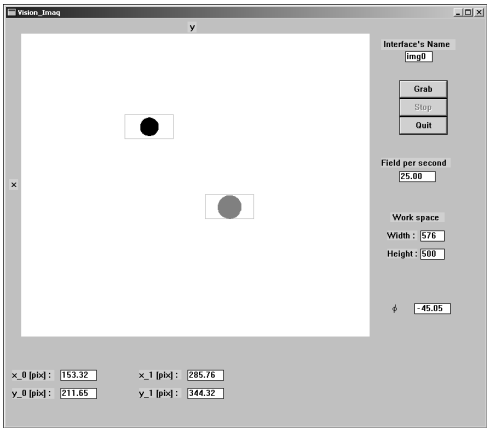

Fig. 2.7. Vision interface.

In Figure 2.7, the results of the image acquisition program are shown. We clearly see the detection of two points located on the PVTOL prototype. From the measurement of these two points, we compute the position x, y and the angle ϕ of the system.

Figures 2.8–2.10 show the performance of the controller when applied to the PVTOL aircraft. Hovering at 60 pixels as well as following a horizontal trajectory were performed satisfactorily. The choice of the values for a, b, c, d were carried out satisfying inequalities (2.53)–(2.57). However these parameters have been tuned experimentally in the sequence as they appear in the control input u_2.

In Figure 2.8, one can see the gap between the real horizontal position of the PVTOL in the x-axis and the desired position. In the x-axis, 1 cm corresponds to 5 pixels. This means that the error position is about 2 to 3 cm.

Figure 2.9 presents the gap between the real PVTOL altitude and the desired altitude. One can notice that the PVTOL follows the desired reference. In the vertical y-axis, 1 cm corresponds to 2.5 pixels. This means that the vertical position error is about 2 to 3 cm.

Figure 2.10 shows the evolution of the ϕ-angle In this figure, one can see that the control law achieves convergence of the ϕ-angle to zero. Therefore, the difference between the real trajectory and desired one is due to friction. The results are nevertheless very satisfactory.

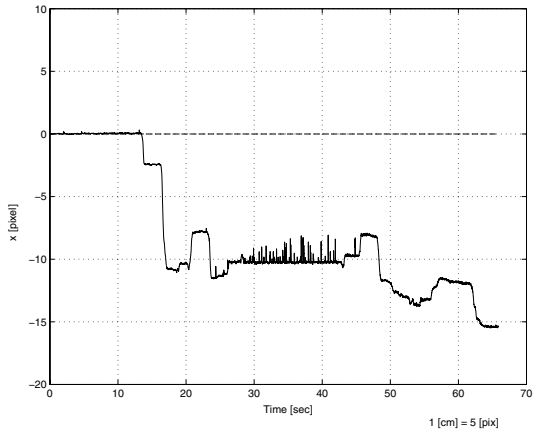

Fig. 2.8. Position $[x]$ of the PVTOL.

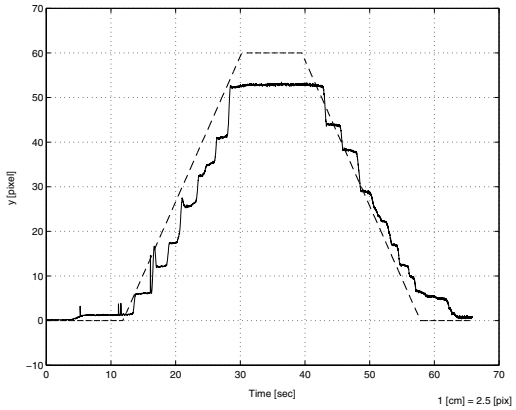

Fig. 2.9. Position $[y]$ of the PVTOL.

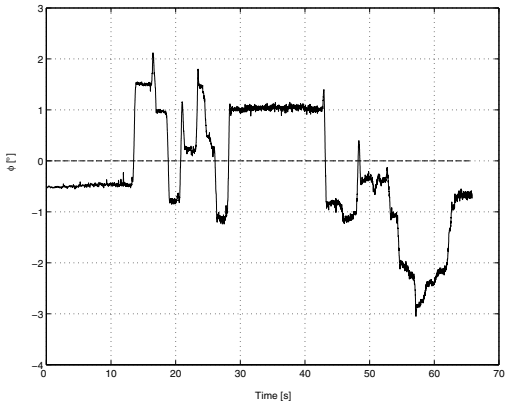

Fig. 2.10. Orientation $[\phi]$ of the PVTOL.

2.5 Conclusion

We have presented a simple control algorithm for stabilizing the PVTOL. The controller is an extension of the nested saturations technique introduced in [172]. The convergence Lyapunov analysis has shown that the proposed algorithm is asymptotically stable.

We have been able to test the algorithm in real-time applications. The simplicity of the algorithm was very useful in the implementation of the control algorithm. The results showed that the algorithm performs well. We were able to perform autonomously the tasks of take-off, hover and landing.

This platform exhibits the same difficulties found in autonomous flight close to the ground and can be used as a benchmark for developing controllers for unmanned flying vehicles.