2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Experimental Framework for a Ducted-Fan Miniature Aerial Vehicle

L. Gentili, L. Marconi, R. Naldi and A. Sala

CASY-DEIS-University of Bologna, ITALY.

Abstract— **This paper focuses on the design and validation of a control framework for a Ducted-Fan Miniature Aerial Vehicle (MAV) realized by the University of Bologna in order to pursuit simple high level operations like surveillance, video capture, etc. After a description of the aircraft prototype and the nonlinear control law, two different control scenarios are then addressed, the first one characterized by a continuous interaction with the human operator (denoted as *Remotely Operated Vehicle Mode*), and the second one characterized by preloaded desired references to be followed (*Trajectory Mode*). Detailed experiments are finally presented to show the effectiveness of the proposed design techniques.**

I. Introduction

Goal of this paper is to present a description of the *control framework* for a miniature ducted-fan Unmanned Aerial Vehicle (UAV) (see [8], [15], [12] for similar configurations) capable of fully autonomous or partially-supervised remote controlled flight, and to provide detailed experiments showing the effectiveness of the proposed design. The final purpose of the project is to design and build a Vertical Take-Off and Landing (VTOL) aircraft suitable for a large number of civil applications, such as mobile video-surveillance, forest fire detection, environmental pollution monitoring, detailed relief map survey in un-accessible sites, traffic monitoring etc., and to develop a control architecture suitable for a possibly large number of VTOL UAV systems, which include miniature helicopters (see for example [7], [3], [2]), as well as miniature quad-rotors (see for example [18], [4]).

The aircraft is designed as a ducted-fan which is composed by two main components: the first one consists of a single fixed-pitch propeller which generates the main thrust, while the second one consists of a set of actuated aerodynamic surfaces, whose effect, in conjunction with the main rotor downwash, is to provide the force and torque components needed to acquire full controllability of the attitude dynamics. The effectiveness of similar UAV configurations is testified by recent contributions (see for example [12], [8], [17], [15] and [6]) presenting preliminary results and tests along this design direction.

In this paper we briefly present a nonlinear robust regulator, based on the theoretical results in [13] , capable of asymptotically tracking arbitrary desired trajectories which are functionally controllable for the proposed configuration. The control law has been implemented on the experimental framework which is the main focus of the paper. The latter is composed by the avionics, whose hardware is described in Section IV-B, and the software algorithms which have been developed in order to be adaptable, with minor modifications, on different UAV architectures. The core of the software is represented by the nonlinear control law which is supported by a low-level state estimation layer and high-level application layer. In particular the application layer has been designed by figuring out two possible controlled scenarios which differ each other by the type of interactions with the human operator. The first is characterized by a continuous supervision of the human operator which remotely provides "high-level" desired commands which are properly elaborated by the onboard control unit to compute actual control inputs to the actuators. This control mode will be referred to as *ROV* (Remotely Operated Vehicle) *Mode*. The second type of controlled scenario is characterized by a partial supervision of the human operator which periodically transmits desired maneuvers (in terms, for instance, of way-points and desired trajectory profiles between them) and tasks (such as operations to be accomplished once reached certain positions) on an appropriate future time horizon. This mode will be referred to as *trajectory mode*. A detailed description of the communication protocol to implement the previous two operative scenarios will be also presented.

II. Dynamical Model of the Ducted-Fan MAV

According to [13], in order to derive a mathematical model for the system, the Newton-Euler equations of a rigid body can be used. In particular the dynamical model of the MAV with respect to an inertial frame is described by

$$\begin{array}{rcl} m\ddot{p} & = & Rf^b \\ J\dot{\omega} & = & -Skew(\omega)J\omega + w_{\rm P}G\omega + \tau^b \end{array} \tag{1}$$

where f^b and τ^b represent respectively the vector of forces and torques applied to the vehicle expressed in the body frame, m the vehicle total mass, $J = diag(j_x, j_y, j_z)$ the diagonal inertia matrix, $p = \mathrm{col}(x, y, z)$ the position of the center of mass, ω the angular velocity expressed in the body frame, R the rotation matrix relating the body frame and the inertial frame (parameterized by means of roll, ϕ, pitch, θ, and yaw, ψ) and $G = Skew\left(\mathrm{col}(0, 0, I_{rot})\right)$ with I_{rot} the inertia of the propeller with respect to the spin axis. The term $w_{\rm P}G\omega$ in (1) is introduced to model the gyroscopic precession torque effect due to the angular speed $w_{\rm p}$ of the propeller. [1] The external wrench vector $col(f^b, \tau^b)$ applied

This work was supported by MIUR. Corresponding author: Roberto Naldi, CASY-DEIS University of Bologna, Via Risorgimento 2, 40136 Bologna, Italy. Tel: 0039 051 2093875, Fax: 0039 051 2093073, email: roberto.naldi@unibo.it

[1] We denote by $Skew(\mathrm{col}(x_1, x_2, x_3))$, the skew-symmetric matrix with the first, second and third row respectively given by $[0, -x_3, x_2]$, $[x_3, 0, -x_1]$ and $[-x_2, x_1, 0]$.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

to the rigid body can be seen as a nonlinear function of four *control inputs* $u = col(T, a, b, c)$ with T the propeller thrust, a, b and c angular deflections of the flaps subsystem which, deviating the air flow coming from the propeller, are used to govern the attitude dynamics of the system and indeed to counteract the motor torque and to project the propeller thrust in a desired direction. Accordingly, as an approximation, the forces and torques can be written as

$$\begin{aligned} f^b &= \left[\begin{array}{ccc} 0 & 0 - T \end{array}\right]^T + R^T \left[\begin{array}{ccc} 0 & 0 & mg \end{array}\right]^T \\ \tau^b &= A(T)v + B(T) \end{aligned} \qquad (2)$$

with $v = col(a,\, b,\, c)$ and with

$$A(T) = T \begin{bmatrix} 0 & -k_1 & 0 \\ k_1 & 0 & 0 \\ 0 & 0 & -k_2 \end{bmatrix}, B(T) = \begin{bmatrix} 0 \\ 0 \\ N(T) \end{bmatrix}$$

where $N(T) = (k_N/k_T)T$, with k_N, k_T, k_1 and k_2 constant parameters.

III. Nonlinear Control Law

Goal of the control law proposed in [13] (to which the reader is referred for a detailed description of the control design) is to generate the four control inputs in order to asymptotically track the four desired vertical, lateral, longitudinal and heading time references $x_{\rm r}(t)$, $y_{\rm r}(t)$, $z_{\rm r}(t)$ and $\psi_{\rm r}(t)$. The reference signals are supposed to be known arbitrary time profiles with the only restrictions dictated by the *functional controllability* of the system and by the fulfillment of physical constraints on the control inputs. For this purpose the overall control law has been divided into a vertical controller and a cascade structure for attitude and lateral/longitudinal control; in the latter, the attitude loop plays the role of *inner loop* and the lateral/longitudinal loop of the *outer loop* . In particular the nonlinear lateral/longitudinal control law has been designed by means of *nested saturations* (see [11]). The overall controller is a mixture of feedforward control terms (synthesized according to the references) and high gain control terms able to obtain asymptotic tracking of the four reference signals in case of perfect knowledge of the aircraft parameters, otherwise practical tracking with arbitrary small error.

A. Vertical Control Law

The vertical dynamics of the system are given by (see (1) and (2))

$$m\ddot{z} = -T\Psi(\Theta) + mg \qquad (3)$$

where $\Psi(\Theta) := C_\phi C_\theta$ (having denoted with $\Theta = \mathrm{col}(\phi, \theta)$). In order to decouple the vertical from the attitude dynamics, we consider the following preliminary choice for the input T

$$T = \frac{-T' + m(g - \ddot{z}_{\rm r})}{C_{\phi_{\rm s}} C_{\theta_{\rm s}}} \qquad (4)$$

in which $C_{\phi_{\rm s}} := max\{C_\phi, C_{\bar\phi}\}$ and $C_{\theta_{\rm s}} := max\{C_\theta, C_{\bar\theta}\}$, with $\bar\phi$ and $\bar\theta$ chosen s.t.

$$\bar\phi \in \left(\|\phi_{\rm r}(t)\|_\infty, \frac{\pi}{2}\right), \qquad \bar\theta \in \left(\|\theta_{\rm r}(t)\|_\infty, \frac{\pi}{2}\right) \qquad (5)$$

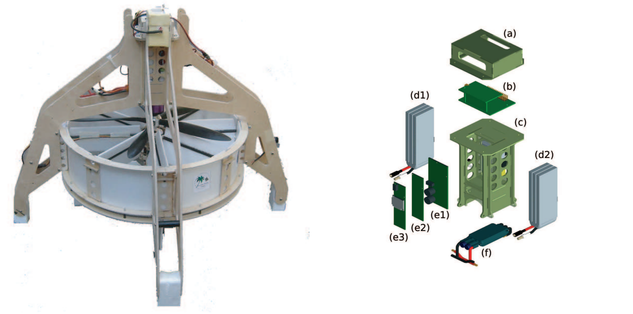

Fig. 1. The prototype of the ducted-fan Miniature Aerial Vehicle used for the experiments with avionics and control hardware: the fuselage ((a) and (c)) containing the MNAV sensor (b), the embedded ARM computer (e), the brushless motor regulator (f) and the LiPo batteries (d).

where T' is an auxiliary control input chosen as the PID control law

$$\begin{aligned} T' &= \xi - k_{p2}(\dot{e}_z + k_{p1}e_z) \\ \dot\xi &= -k_{p2}(\dot{e}_z + k_{p1}e_z) + m\dot{e}_z \end{aligned} \qquad (6)$$

in which $e_z := z - z_{\rm r}$ is the vertical tracking error.

B. Attitude and Lateral/Longitudinal Control Law

The control of the attitude dynamics is achieved by means of a law of the form $v = A^{-1}(T)\left(\tilde{v} - B(T)\right)$ with

$$\begin{aligned} \tilde{v} \;=\; & -K_P\left(K_D\omega + \begin{bmatrix} \tan\Theta - A(\Theta_\psi)\Theta_{\rm out} \\ \psi + K_\psi\eta_\psi \end{bmatrix}\right) + \\ & + \; K_P K_D \omega_{\rm r} + K_P \begin{bmatrix} \tan\Theta_{\rm r} \\ \psi_{\rm r} \end{bmatrix} + \\ & + \; J\dot\omega_{\rm r} + \mathrm{Skew}(\omega_{\rm r})J\omega_{\rm r} - w_{P_{\rm r}}G\omega_{\rm r} \end{aligned}$$

in which K_P, K_D and K_ψ are design parameters, $\Theta_{\rm r}$ and $\omega_{\rm r}$ are functions of the lateral, longitudinal, vertical and heading references,

$$A(\Theta_\psi) := \begin{bmatrix} -C_\psi & S_\psi C_\theta / C_\phi \\ S_\psi / C_\theta & C_\psi / C_\phi \end{bmatrix},$$

η_ψ is an integrator variable governed by $\dot\eta_\psi = \psi - \psi_{\rm r}$ and $\Theta_{\rm out}$ is a residual control input which will be chosen as the following nested-saturation control law

$$\begin{aligned} \Theta_{\rm out} &= \lambda_3\sigma\left(\frac{K_3}{\lambda_3}\xi_3\right) \\ \xi_3 &:= \left[\begin{array}{cc} \dot{e}_y & \dot{e}_x \end{array}\right]^{\rm T} + \lambda_2\sigma\left(\frac{K_2}{\lambda_2}\xi_2\right) \\ \xi_2 &:= \left[\begin{array}{cc} e_y & e_x \end{array}\right]^{\rm T} + \lambda_1\sigma\left(\frac{K_1}{\lambda_1}\xi_1\right) \\ \xi_1 &:= \left[\begin{array}{cc} \eta_y & \eta_x \end{array}\right]^{\rm T} \end{aligned}$$

where $e_x := x - x_{\rm r}$, $e_y := y - y_{\rm r}$ and η_y, η_x represent integrator variables governed by $\dot\eta_y = e_y$ and $\dot\eta_x = e_x$. In the definition of the outer controller, (λ_i, K_i), $i = 1, 2, 3$, represent design parameters while $\sigma(\cdot)$ is a *saturation function* rigourously defined in [11].

IV. The Experimental Framework

The experimental framework can be thought as divided into two different subsystems, the *aircraft prototype* and the *control framework*. The aircraft prototype (see figure 1) has been designed according to the general ideas proposed in Section II and its details are proposed in Section IV-A. Since one of the design goal was to realize a low cost miniature autonomous system, a miniature integrated *IMU-GPS* (Inertial Measurement Unit - Global Positioning System) avionics sensor suite and a miniature *cpu-board* have been adopted. More details about the selected Hardware are presented in Section IV-B. The control framework (onboard autopilot) is designed according to a three layer structure:
layer 1: state estimation architecture. This layer is realized in order to estimate the state of the system from the sensor outputs both in presence of the GPS signal or not.
layer 2: the nonlinear control law. This is the core of the framework providing the four control inputs to follow desired time reference signals.
layer 3: communication and interaction algorithms. This high level layer is designed to handle all communication and to interact with the human operator; it is hence devoted to suitably create the time reference signals used in the control layer in order to assure that the UAV accomplishes desired task.

A. The Aircraft Prototype

The aircraft prototype has been designed by means of a Computer Aided Design (CAD) software suite. The avionics is composed by the following components: a IMU/GPS sensor board, an embedded computer board, a wireless communication board, a serial interface to connect the sensor board with the processor board and eventually other peripheries, a power system with a LiPo (Lithium Polimery) battery and voltage regulator with an overall weight of approximately 150 g.
As far as the layout is concerned, a system with two levels of flaps and with the avionics box positioned over the propeller disk has been selected. The power subsystem has then been designed to achieve a desired flight endurance in term of hovering time. By comparing different solutions available on market place, a brusshless motor capable of a static thrust of about 1900 g, when used with a 3 blade 13 inches propeller and a couple of LiPo batteries (3 cells, 2.1 Ah each), has been selected. The weight of the power subsystem is about 0.7 Kg, including a brushless motor regulator. This system is nominally able to guarantee at least 15 minutes of hovering when the take-off weight of the aircraft is less or equal to 1.5 Kg. The total mass, inertia, and the other aerodynamical parameters which characterize system dynamics have been accurately estimated by the CAD model, and are shown in Table I. The lift coefficient and the drag coefficient of the flaps can be estimated by choosing a known NACA airfoil profile (see [1], [16]) for which the above numbers are known. Once the prototype has been implemented, we measure a real hovering endurance of about 14 minutes, even though in the last 2 minutes the overall vertical authority drastically decreases.

$J = diag(1.8\ 1.8\ 0.9) \cdot 10^{-2}\ kg\ m^2$	$m = 1.576\ kg$
$k_1 = 31.13$	$k_2 = 23.67\ n\ m/rad$
$k_N = 9.07 \cdot 10^{-7}\ N/m$	$k_T = 7.6 \cdot 10^{-5}\ N$
$I_{rot} = 0.3 \cdot 10^{-3}\ kg\ m^2$	$k_{p1} = 4\ k_{p2} = 160$
$K_P = 0.4\ K_D = 0.39\ K_\psi = 0.3$	$\lambda_1 = 0.3\ \lambda_2 = 0.8\ \lambda_3 = 0.5$
$K_1 = 0.06\ K_2 = 1\ K_3 = 0.08$	

TABLE I

The parameters of the prototype of ducted-fan MAV used in our experiments.

B. The Onboard Hardware

The avionics used in the experiments proposed in this work is composed by two main hardware components, the IMU/GPS sensor and the embedded ARM computer. The IMU/GPS sensor provides the inertial and position measures necessary to estimate the whole system state and also a micro-controller to drive the actuators (miniature R/C servos). The embedded computer is a modular unit which can be equipped with different I/O and special purpose boards. It is based on a general purpose CPU in order to run an open source operative system; this allows to exploit an established software framework to implement the autopilot software.

1) IMU/GPS: The measurement hardware is based on a Crossbow MNAV100CA (see [9] for the detailed specifications), which is is an integrated low cost miniature sensor suite weighting only 33 g. The MNAV includes all the sensors necessary to accurately estimate the attitude and the position of a rigid body in the configuration $SO(3) \times \mathbb{R}^3$. These sensors include the following Micro Electro-Mechanical Systems (MEMS) components: a 3-axis accelerometers to provide the accelerations in the body frame; a 3-axis gyros to provide the angular velocities, a 3-axis magnetometers to provide the magnetic field measure along the body axis, a static pressure sensor to measure the local air pressure which depends on the altitude of the system and a dynamic pressure sensor (Pitot tube) to measure the relative wind velocity.
The MNAV integrates also a GPS receiver which is nominally capable of a lateral/longitudinal position accuracy of about 5 m and velocity accuracy of 0.1 m/s. The sensor board is equipped with an embedded microprocessor which provides serial ports and the PWM outputs necessary to drive the R/C servos and the motor regulator of the ducted-fan. The overall firmware is available as an opensource code and is capable of processing sensor outputs up to 100 Hz. For a detailed comparison of this platform with respect to other miniature autopilots see also [5].

2) Embedded ARM computer: The embedded computer is composed by a miniature computer board based on an ARM (Advanced Risc Machine) processor at 400 Mhz (for detailed specification see [10]), weighting approximately 8 g. The board is connected to an interface board providing two serial RS232 ports and to a wireless board. The main computer is capable of executing an optimized Linux kernel 2.6 operative system, in which, using standard POSIX API (see [14]), the algorithms are implemented. The processor receives sensor outputs by means of RS232 communication and then computes the control outputs which are sent back to the MNAV through a second RS232 port. The MNAV firmware is able to decode these control values and to generate the corresponding PWM signals necessary to drive the servos. The maximum sample rate of the closed loop system is limited by the MNAV communication rate frequency which is 50 Hz.

C. Communication and Interaction Algorithms

Almost all civil applications, such as surveillance and image acquisition, require interaction with a human operator. In our framework, the operator decides which task should be executed by means of a *ground station* and the information is sent to the remote aircraft through the *wireless network* connection. The role of the ground-station is to support, from the ground side, the functionalities of the layer 3 of the onboard autopilot. To cover a possibly large and heterogenous set of operations we allow two different interaction scenarios. The first one is based on *asynchronous* communication in which a single task is divided into a certain number of way-points to be followed through a desired trajectory and then is communicated to the aircraft. The exchange of information between the ground station and the aircraft could be not frequent as it could even end with the initial programming. In the second scenario, the system is continuously governed by high level commands (e.g. gain/lose altitude, hover, move forward/backward etc.) which are converted into instantaneous desired trajectories; the communication is frequent and the network should be sufficiently reliable (in particular in term of delay) in order to let the desired task to be accomplished. Both solutions require an on-board software interpreter (parser), which is basically implemented as a *trajectory smoother* able to obtain the four time reference signals $x_{\mathrm{r}}(t)$, $y_{\mathrm{r}}(t)$, $z_{\mathrm{r}}(t)$ and $\psi_{\mathrm{r}}(t)$ to be tracked by the nonlinear control law.

1) Trajectory Mode Scenario: In this scenario, all the available information relative to a certain high level mission is translated in a predefined *language* and collected in a data packet. Each data packet contains a set of micro-instructions in order to identify both a certain desired *waypoint* (desired position and heading orientation that has to be reached) and also a set of symbols which are interpreted by the parser to compute the desired trajectory according to a certain predefined routine. A micro instruction, as visible in figure 2, has been defined by the following fields:
T_F - denotes the overall amount of time to reach the desired final position;
P - denotes the desired lateral, longitudinal and vertical position x_{r}, y_{r} and z_{r} with respect to the standard inertial NED (North - East - Down) frame at time $t_i + T_F$, with t_i denoting the time when a micro-instruction starts to be executed;
V - denotes the desired velocities $\dot{x}_{\mathrm{r}}$, $\dot{y}_{\mathrm{r}}$ and $\dot{z}_{\mathrm{r}}$, at time $t_i + T_F$;
H - denotes the desired heading ψ_{r} of the system at time $t_i + T_F$. This along with P define the desired final position and orientation (way-point) that have to be reached at the end of the micro-instruction;
H_V - denotes the desired heading velocity $\dot{\psi}_{\mathrm{r}}$ at time t_i+T_F;
C - denotes a command for the parser. It can be used to choose between different ways to generate a desired trajectory according to the previous values of the micro-instruction (e.g. *spline*);
OP - denotes a certain high level operation that could be performed over the desired waypoint. Such operations could be photos, video, etc.
This way of interaction has revealed sufficiently general in order to accomplish a considerable amount of high level operations. Once one packet or a sequence of packets are received, the on-board interpreter, according to a predefined policy C, calculates the trajectory fulfilling the desired requirements by merging sequentially the micro instructions of each packet. This allows to define trajectories and operations also run-time, once more information of the task may be available, as it has been shown in the example in figure 2: four way-points are given by two different data packets, in which S and E denotes the *start* and *end* delimiters; in particular the OP packet defines the operation to be performed over the relative way-point (photo-shot at way-point 1 and 4, video shot at way-point 2). Observe that this approach of generating feasible trajectories is a generalization of the standard way-point navigation implemented in *commercial off-the-shelf* (COTS) autopilots. As a matter of fact, not only the desired point to be reached is specified, but also more information relative to the desired trajectory between two different points and possibly the task to be performed over a certain target are transmitted.

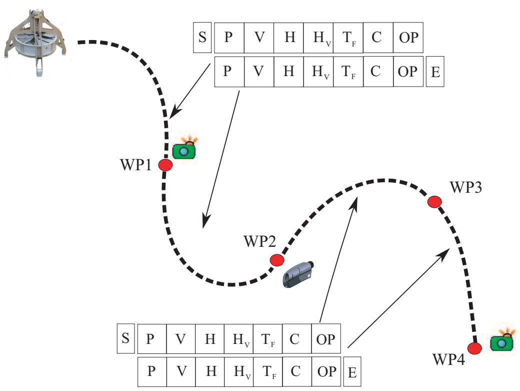

Fig. 2. Trajectory mode scenario. The four way-points are given by two different data packets, in which S and E denotes the *start* and *end* delimiters.

2) ROV Mode Scenario: In this scenario the human operator remotely provides “high level” desired behaviors (i.e. gain/loss altitude, hover, move toward/backward, land, etc...) which are properly elaborated by the onboard autopilot to compute instantaneous reference and actual control inputs to the actuators. In order to increase immediateness of remote piloting, we make use of a new body-fixed reference frame F_a (auxiliary frame), with the origin in the center of mass of the system, and with the axis x^a aligned with the projection of the body x axis over the inertial lateral/longitudinal plane. The use of the auxiliary frame allows the human pilot to control the lateral / longitudinal displacement by looking at the images coming from the wireless camera or at the virtual representation of the ducted-fan given by the ground station. The operator governs the system through the information contained in data packets similar to the ones defined in Section IV-C.1 for *trajectory mode* scenario; in *ROV mode* data packets are characterized by:

V - denotes the instantaneous desired velocity $\dot{x}^a_{\mathrm{r}}$, $\dot{y}^a_{\mathrm{r}}$ and $\dot{z}^a_{\mathrm{r}}$ along the axis of the auxiliary frame F_a;

H_V - denotes the desired instantaneous heading velocity $\dot{\psi}_{\mathrm{r}}$;

C - define a specific command for the control implemented on the remote host; some of them are already implemented in the autopilot:

−*motor start-up*, *motor power-off*: control the propeller on/off;

−*take off*: this command forces the autopilot to execute an open loop control signal for the thrust T in order to rapidly take-off from the ground. A timer switch from the open loop control to the a closed loop stabilization at a safe altitude from the ground;

−*emergency landing*: this commands impose a vertical fast landing trajectory to the system. It is used in situation where the system is flying at risk;

−*switch*: to change the current mode from ROV to trajectory and viceversa;

−*trim commands*: these are used in order to modify the *trim* position of the flaps and the nominal hovering value of the propeller velocity. These commands are clearly adopted to compensate manually for non ideality in the construction of the prototype;

−*video-on*: start video shot;

−*video-off*: end video shot;

−*photo*: execute photo shot.

In our experimental framework the human operator could govern the aircraft providing high level commands (take off, gain altitude, hover and move towards a direction etc.) using a standard joy-pad whose inputs are sampled periodically and translated in the *ROV mode* packet previously defined and then sent to the remote host continuously (at the moment packets are transmitted at 4 Hz frequency). As a default policy, each packet is labeled with a progressive number by the ground station, and non ordered packets are discharged by the onboard communication layer.

3) Ground-Station software: To allow the interaction with the human operator a *ground-station* software interface has been developed. The role of the ground-station is to support, from the ground side, the functionalities of the layer 3 of the onboard autopilot. For this reason two different control panels for each type of communication have been developed. In Trajectory Mode, with the support of a detailed satellite map, it is possible to choose a desired ordered set of waypoints and a desired interpolation trajectory. The second panel allows to control the functionalities of a keyboard or a joy-pad input interface, in order to control the UAV in *ROV mode*. The human pilot, in this last scenario, is helped by a realtime representation of the UAV on the map in which it is possible to monitor all state and inputs of the system and by a camera window, in which the images coming from an onboard wireless camera are shown. The ground station continuously samples the output of the joy-pad, at a predefined frequency, and generate the ROV mode packets to be sent continuosly through the wireless connection.

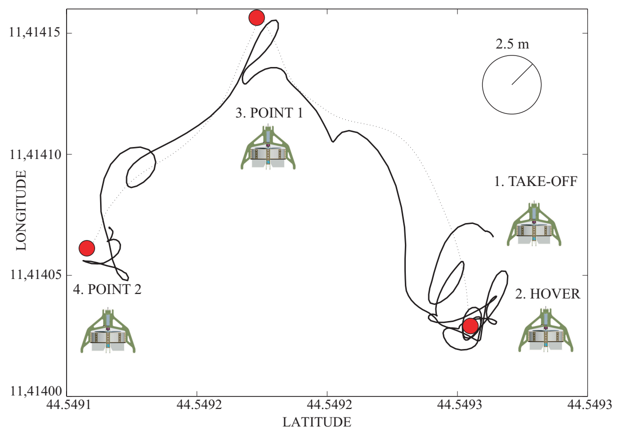

Fig. 3. Flight experiment in *ROV mode* in term of lateral and longitudinal position. The $2.5m$ radius circle represents the GPS horizontal accuracy.

V. Experimental Results

In this Section we present some experimental results obtained with the set-up described in the previous Sections; Table I shows physical parameters of the aircraft. Each maneuver has been realized outdoor where small wind disturbances can eventually have affected the dynamics of the system. The *motor start-up* ($START$ command), *take off* ($JUMP$ command) and *motor power-off* ($STOP$ command) procedures of each flight have been realized by means of *ROV mode* commands which are visible in figure 4 (e). In particular during the *take off*, in correspondence of the $JUMP$ commands, the thrust T is kept at a constant high value in order to bring the vehicle as fast as possible at a certain distance from the ground and to avoid the system to overturn due to air turbulence and frictions with the ground. This open loop control for the thrust T stops at a safe altitude from the ground (2 m) and the control is switched automatically to the closed loop stabilization. The test maneuver refers to a flight performed completely in *ROV mode*. It simulates a possible application scenario in which the MAV, governed remotely by the human operator, has to accomplish a mission which consists on hovering at a particular position and then visiting two different points, as depicted in figure 3. The human operator can govern the aircraft by looking at the output of the wireless camera, the virtual representation on the ground station and also (since we are simulating simple applications in a circumscribed area) by looking at the MAV directly. After the start routine and the take-off (which once again can be recognized in

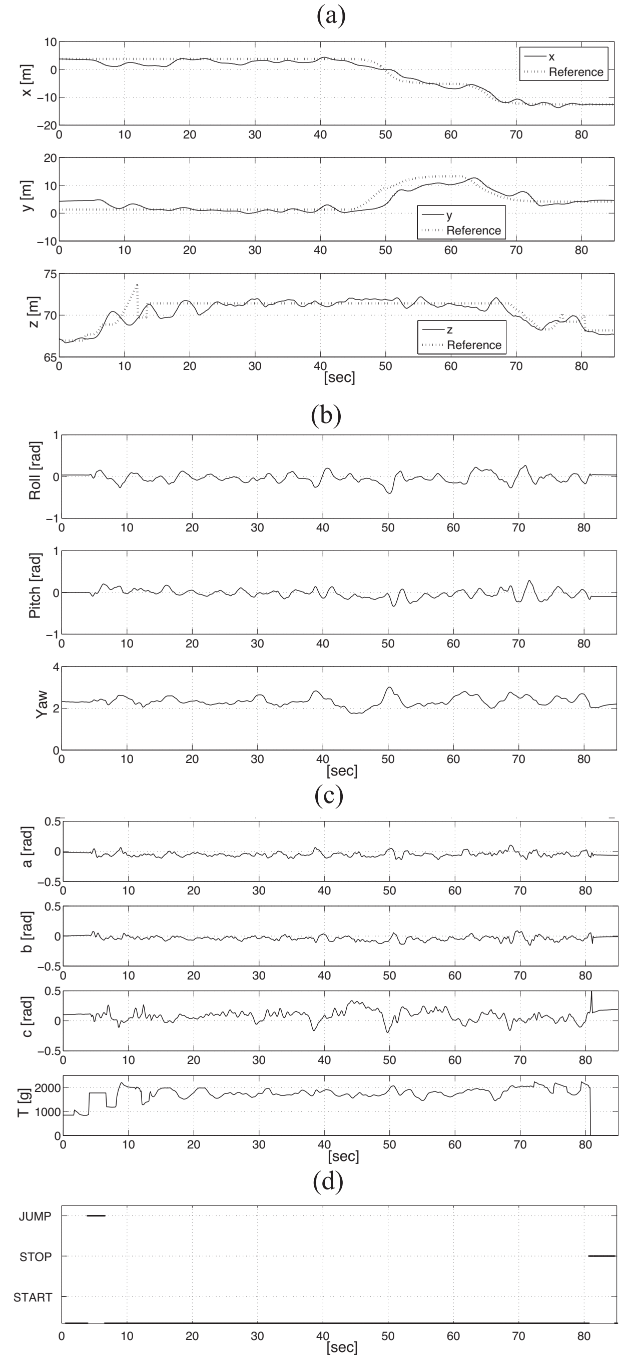

Fig. 4. Flight experiment in *ROV mode*: (a) positions (solid line) and the reference signals generated from the sequence of *ROV mode* packets received (dashed line). (b) Roll, Pitch and Yaw angles. (c) Control inputs. (d) The *ROV mode* commands.

figure 4 (e) looking at the commands which are sent to the aircraft and in figure 4 (d) by the thrust behavior), the human pilot maintains a hovering flight for about 30 seconds and then start moving towards lateral and longitudinal directions, as depicted in figures 4 (a) and 4 (b), in order to reach the first and second point shown in figure 3. We plot all the references generated along the lateral, longitudinal and vertical axis by means of the *ROV mode* interaction. Observe that the position in figure 4 (a) is closed to the desired one throughout the whole flight and the tracking error is still compatible with measurement errors. Basically this experiment shows the capability of our low cost framework to succeed in simple tasks which can be part of a more complex application scenario.

VI. Conclusion

In this paper we have shown results relative to the prototyping, modelling and development of a control framework for a ducted-fan miniature aerial vehicle. The prototype consists of a main propeller and a set of active aerodynamic surfaces controlled to compensate the motor torque and to direct the propeller main thrust in the desired direction. The overall hardware and software architecture is described emphasizing two different control scenarios: the first one characterized by a continuous interaction with the human operator (denoted as *Remotely Operated Vehicle Mode*), and the second one characterized by preloaded desired references to be followed (*Trajectory Mode*). Experiments have been shown to validate the overall design.

References

[1] I. H. Abbott and A. E. von Doenhoff. *Theory of Airfoil Sections*. Dover, 1959.

[2] J. C. Avila, B. Brogliato, A. Dzu, and R. Lozano. Nonlinear modelling and control of helicopters. *Automatica*, 39:1584–1596, 2003.

[3] M. Bejar, F. Cuesta, and A. Ollero. On the use of soft computing techniques for helicopter control in environment protection mission scenarios. *To appear in Intelligent Automation and Soft Computing*, 2007.

[4] P. Castillo, A. Dzul, and R. Lozano. Real-time stabilization and tracking of a 4 rotor mini-helicopter. *IEEE Transactions on Control Systems Technology*, 12:510–516, 2004.

[5] H. Chao, Y. Cao, and Y. Chen. Autopilots for small fixed-wing unmanned air vehicles: A survey. *in Proc. of the International Conference on Mechatronics and Automation (ICMA) 2007*, 2007.

[6] J. Fleming and T. Jones. Improving control system effectiveness for ducted fan vtol uavs operating in crosswinds. *2nd AIAA "Unmanned Unlimited" Systems, Technologies and Operations - Aerospace*, 2003.

[7] V. Gavrilets. Aerobatic maneuvering of miniature helicopters. *Ph.D. Thesis*, 2003.

[8] I. Guerrero, K. Londenberg, P. Gelhausen, and A. Myklebust. A powered lift aerodynamic analysis for the design of ducted fan uavs. *2nd AIAA "Unmanned Unlimited" Systems, Technologies, and Operations*, 2003.

[9] Crossbow Technology Inc. Mnav user manual.

[10] Gumstix Inc. http://www.gumstix.com.

[11] A. Isidori, L. Marconi, and A. Serrani. *Robust Autonomous Guidance*. Springer, 2003.

[12] E.N. Johonson and M.A. Turbe. Modeling, control and flight testing of a small ducted fan aircraft. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[13] L. Marconi and R. Naldi. Robust nonlinear full degree of fredom control of an helicopter. *Automatica*, 42:1584–1596, 2007.

[14] F. Mueller. A library implementation of posix threads under unix. *in Proc. of Winter USENIX*, 1993.

[15] J.M. Pflimlin, P. Soures, and T. Hamel. Hovering flight stabilization in wind gusts for ducted fan uav. *42nd IEEE Conf. on Decision and Control*, 2004.

[16] R.F. Stengel. *Flight Dynamics*. Princeton University Press, 2004.

[17] R.H. Stone. Control architecture for a tail-sitter unmanned air vehicle. *5th Asian Control Conference*, 2004.

[18] A. Tayebi and S. McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transactions Control Systems Technology*, 14:562–571, 2006.