Development of a test-bed to implement and validate real-time control strategies for aerial vehicles

A. Berna ∗ P. Castillo *** G. Sanahuja *** F. González ** P. García ∗ P. Albertos ∗

* Universitat Politècnica de València, 46020 Valencia, España.
  e-mail: anberfe1@upvnet, {pggil, pedro} @isa.upv.es
** Universit´e de Technologie de Compi`egne - CNRS, UMR 6599
  e-mail: {gsanahuju, castillo}@hds.utc.fr
*** Universidad Central de las Villas, Cuba. e-mail: Fidel@uclev.edu.cu

Abstract: Nowadays, there exist several platforms or experimental prototypes of aerial vehicles, most of them being specific for certain applications. The implementation of the control laws is, in most cases, restricted to given criteria and pre-established conditions defined by the commercial system (or the builder). In this paper, a new generic platform operating under Linux-RT, XtratuM and Partikle systems is introduced. Analyze, evaluate and improve the performance of future control embedded systems are the goals of the proposed platform. The platform is composed of a ground control unit (GCU) and an aerial system. In order to validate the platform, some experiments stabilizing the orientation and the altitude of the aircraft have been executed in free flight. The experimental results show the good performance and the robustness of the proposed control scheme.

Keywords: Non-linear control, UAV, Real-time systems, Robustness, Real time application

1. INTRODUCTION

An Unmanned Aerial Vehicle (UAV) is, usually, controlled by an embedded system. This system uses the information obtained from several sensors, like Integrated Measurement Units (IMU), cameras, radars, lasers, infra-red or ultrasound sensors, etc., to be able to navigate through an unknown environment, see Borenstein (1992) and Simons (1987). Its applications are multiple and they extend to several fields, from military or civil (fire detection, rescue missions frontier control, surveillance of highways and roads, weather forecast, etc.) to commercial (agriculture, fishing, logistics, surveillance of rail tracks and electrical networks, aerial security, etc.), see Castillo et al. (2005).

In many applications, Vertical Take-Off and Landing (VTOL) vehicles, also known as rotorcraft vehicles, are preferred over fixed wing vehicles because of their higher manoeuvrability. The quad-rotor is the most popular VTOL rotorcraft. This type of vehicle attempts to achieve stable hovering and precise flight by balancing the forces produced by the four rotors. This aerial configuration has allowed the development of different platforms for research, games or educational purposes.

Nowadays, there exist several platforms or experimental prototypes of aerial vehicles, most of them being specific for certain applications, see Quanser (2007). In those systems the implementation of the control laws is, in most cases, restricted to given criteria and pre-established conditions defined by the commercial system (or the builder). If the prototype has been developed with an embedded control system, this has been usually implemented in a cyclic algorithm without defining specific system priorities, see Sanahuja (2010) and Lozano (2010). In all the previous cases, these experimental systems are limited by the control software and restricted to use a predefined operative system.

A crucial problem, when working with embedded control systems, is the lack of a non-commercial platform to validate and/or evaluate the control strategy (before being implemented in the microprocessor), the sensors and the electronic circuits. This pre-validation allows, among other functions, to save a lot of time in the controller parameters tuning. In the same way, this kind of platform, could be used to analyze, visualize and detect possible failures of the vehicle control system, when faced with aggressive perturbations, and be able to avoid possible material losses. In our knowledge, there are few platforms of this kind, allowing the interchange and/or implementation of different operative systems and, at the same time, being able to adjust, on-line, different control strategies How et al. (2008). Although these platforms have a good performance the price is so expensive.

In this paper, a new real-time platform for the development and validation of control algorithms for UAVs is presented. The main features of the platform are: the real-time validation of control schemes by respecting tasks priorities, its capacity and versatility to operate with sev-
eral O.S., its adaptability to change different controlled
systems and its manoeuvrability to on-line modify the
control parameters. A description of these characteristics
of the platform is given in section 2. In order to validate
the test-bed, the stabilization of a helicopter (quad-rotor)
needs to be realized. In section 3, the dynamic model of
this vehicle is described. A nonlinear control law based on
saturation functions to ensure the helicopter hovering is
presented in section 4. The performance of the proposed
platform has been illustrated when the helicopter makes
hovering. These results are introduced in section 5. Finally,
conclusions are discussed in section 6.

2. EXPERIMENTAL PLATFORM

The proposed platform is composed of a ground control
unit (GCU) and an aerial system. The GCU has been
developed using the Hypervisor XtratuM (see Xtratum
(2009)). Its main characteristic is allowing the concurrent
execution of different operative systems. The aerial system,
is composed of a mini-helicopter, an inertial sensor, em-
ployed to measure the vehicle’s orientation, and a group of
infrared and ultrasound sensors for measuring the altitude,
see Figure 1. Additionally, a spinning top structure has
been developed to carry out the tuning of the parameters
of the control algorithms in an easy and safe way before
the validation procedure.

Ground Control Unit - GCU

We define the ground control unit as the computer where
the different control algorithms, the operative systems and
the peripheral equipment (receiver/transmitter) will be
implemented, measured, manipulated and executed.

The computer is essentially an Intel Pentium Dual Core
and it has a PCI bus connected to the data acquisition
card PCI 1440 of the company PMC (PMC - Motion
Control (1987)). The PCI card is plugged in a peripheral
potential amplifier to regulate the control signal before
being sent to the helicopter by the radio transmitter. The
radio transmitter is a Futaba of 6 channels and it is only
used to transmit the signals to the embedded unit in the
mini-helicopter (Futaba (2006)).

For our application, to measure the attitude of the heli-
ocopter, a wireless IMU is used being connected by an USB
port to the GCU. Likewise, a joystick is also plug in the
computer in order to manually control the helicopter. It
is important to remark that it is also possible to include
more external peripherals, if it is considered necessary.

Real-Time Operative System

The Ground Control Unit uses an open standard Based
RTOS, with a combination of Linux RT, XtratuM and
PaRTiKle systems. The use of Xtratum, like a Hyper-
visor, and Linux RT and PaRTiKle, as O.S., provides
the capacity to test and execute the control strategies in
different systems. XtratuM is designed as an hypervisor
for embedded devices and provides a framework to run
multiple concurrent operating systems in a robust par-
tioned environment (Masmano et al. (2009a)). On the
other hand, PaRTiKle is an embedded real-time operating
system, distributed under the terms of the GNU Public
License (Peiro et al. (2009), Masmano et al. (2009b)).

It is well known that certain values of the jitter can degrade
the control performance and in extreme cases even cause
instability of the closed-loop system, see (Stothert and
MacLeod (1998), Shin and Cui (1995)). Jitter represents
the deviation in or displacement of some aspect of the
pulses in a high-frequency digital signal. Jitter can cause
a display monitor to flicker; affect the ability of the pro-
cessor to perform as intended; introduce clicks or other
undesired effects in audio signals, and loss of transmitted
data between network devices. Electromagnetic interfer-
ence (EMI) and crosstalk with other signals are among
the causes of jitter.

A real-time analysis of the proposed O.S. is done in
order to prove the robustness and to verify the behavior
of some important parameters like latency, jitter and
bandwidth. Figures 2 - 3 (with different time scales) show
the jitter performance for Linux, Linux RT and PaRTiKle
under processor stressed conditions (video visualization
programs).

Notice that, the better jitter behavior is obtained by
PaRTiKle, whilst in Linux the jitter is longer. These
results can be observed in Table 1 where a statistical
analysis of data displayed in the previous figures has been
done. Remember that, in a hard real-time system the jitter
is typically on the order of a microsecond to a few tens
of microseconds, and usually, the critical jitter value is
related with the scheduling period required for a control
task. The tighter deadlines are the more significant is the
impact of the timing jitter.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Standard Dev</th>
<th>Variance</th>
<th>Worst Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>59772.53</td>
<td>3.572 755.653.21</td>
<td>328.585</td>
</tr>
<tr>
<td>Linux RT</td>
<td>3469.54</td>
<td>12.037 677.38</td>
<td>13.062</td>
</tr>
<tr>
<td>PaRTiKle</td>
<td>646.01</td>
<td>417.330.74</td>
<td>3.563</td>
</tr>
</tbody>
</table>

Flight Autonomous System

The free flight autonomous system is composed of the
mini-helicopter, the sensors and a spinning top structure.
The helicopter is from X-UFO company having four rotors
controlled in velocity, more details see Xuf (2002). The
onboard control system in the helicopter was inhibited
in order to use it only like an interface to distribute the control signals to the rotors.

To measure the orientation and the angular rates of the helicopter, inertial sensors (IMU) were used, see (Microstrain (2007)). This IMU uses the wireless connection and the wireless receiver is plugged in the GCU to get the attitude measurements. On the other hand, to estimate the altitude of the structure, several infra-red and ultrasound sensors were employed. These sensors are connected to a microcontroller sending the information to the GCU using also wireless connection. Likewise, to tune easily the control parameters and to avoid damage with the helicopter and sensors, a spinning top structure has been designed. This structure allows evolving the helicopter in 2D with minimum ground friction (only one support point with sliding surface). In addition, the size of the structure is such that it minimizes the well known ground effect. Moreover, the interpolation of the control parameters for those who are in the free flight mode is simpler due to the lightness of the structure.

The dynamical model of the quad-rotor helicopter is well known and studied in the literature (see Fantoni and Lozano (2002), Castillo et al. (2005), Lozano (2010), Pounds and Maloney (2002) and Gafvert (2001)). This model can be obtained using the Newton-Euler’s or the Euler-Lagrange’s or the Quaternions’ approach. In our study, we only describe the more important parts to conceive the nonlinear model. In addition, this mathematical description will be obtained by representing the helicopter as a solid body evolving in a three dimensional space and subject to external forces and torques.

Let us define the position and the orientation of the rotorcraft by \( \xi = (x, y, z) \in \mathbb{R}^3 \) and \( \eta = (\psi, \theta, \phi) \in S^3 \), respectively, where \( x \) and \( y \) denote the helicopter’s position in the horizontal plane whilst \( z \) represents the vertical position, \( \psi \) is the yaw angle around the \( z \) axis, \( \theta \) is the pitch angle around the \( y \) axis, \( \phi \) is the roll angle around the \( x \) axis. Thus, the generalized coordinates of the helicopter can be represented as \( q = (\xi, \eta) \in \mathbb{R}^6 \).

The nonlinear dynamic model is obtained from Euler-Lagrange equations with external generalized forces

\[
\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \begin{bmatrix} Ru \\ \tau \end{bmatrix},
\]

where \( Ru \) denotes the translational force applied to the rotorcraft due to main thrust, \( u \), \( R \) defines the rotational matrix \( R(\psi, \theta, \phi) \) representing the orientation of the aircraft relative to a fixed inertial frame, \( \tau \) describes the yaw, pitch, and roll moments, and \( L \) is Lagrangian equation that is composed of the kinetic and potential energies. Moreover, \( L \) is defined by

\[
L(q, \dot{q}) = T_{trans} + T_{rot} - U
\]

where \( T_{trans} = \frac{m}{2} \dot{\xi}^T \dot{\xi} \) is the translational kinetic energy, \( T_{rot} = \frac{1}{2} \Omega^T I \Omega \) is the rotational kinetic energy, \( U = mgz \) is the potential energy of the aircraft, \( z \) is the rotorcraft altitude, \( m \) denotes the mass of the quad-rotor, \( I \) is the vector of the angular velocity, \( \Omega \) is the inertia matrix, and \( g \) is the acceleration due to gravity. The angular velocity vector \( \Omega \) resolved in the body fixed frame can be related to the generalized velocities \( \dot{\eta} \) (in the region where the Euler angles are valid) using the standard kinematic relationship, \( \Omega = W_\eta \dot{\eta} \), with \( W_\eta \) is a transformation matrix (for details see Goldstein (1980)).

After some simple algebraic manipulations and assuming small angles the following result is obtained

\[
\begin{align*}
m \ddot{x} &= -u \sin \theta \\
m \ddot{y} &= u \cos \theta \sin \phi \\
m \ddot{z} &= u \cos \theta \sin \phi - mg \\
\dot{\psi} &= \tau_\psi \\
\dot{\theta} &= \tau_\theta \\
\dot{\phi} &= \tau_\phi
\end{align*}
\]

(1a) (1b) (1c) (1d) (1e) (1f)

The control inputs \( u, \tau_\psi, \tau_\theta \) and \( \tau_\phi \) are the main thrust and the angular moments (pitch, yaw and roll), respectively.
4. CONTROL ALGORITHM

The main goal of this paper is to show the performance and the robustness of the proposed platform when controlling a system with fast dynamics where missing the information of one sampling period could be critical. In order to simplify the experiments, we are interested in controlling only the attitude dynamics of the helicopter. Notice from (1) that these dynamic equations are represented by two integrators in cascade and the stabilization of this subsystem has been deeply studied in the literature, see for example Teel (1992a,b); Sussmann et al. (1994); Teel (1993); Lin and Saberi (1993); Marconi and Isidori (2000); Grognard et al. (1999); Sussmann et al. (1994) and Castillo et al. (2005).

Recently, Sanahuja et al. (2009) have improved the previous results to stabilize n integrators in cascade using saturation functions for each state. This method is based on the idea of easy control parameters tuning in real-time application. Thus, the control strategy executed in the experiment is obtained as a particular case of the control law proposed in Sanahuja et al. (2009).

Then, it follows that

\[ \tau_\gamma = -\sigma_{\gamma_2} - \sigma_{\gamma_1} \quad \forall \gamma = \psi, \theta, \phi \]  

(2)

with

\[ \sigma_{\gamma_2} = \text{sat}(k_d, \gamma_1, \zeta_{\gamma_2}) \]
\[ \sigma_{\gamma_1} = \text{sat}(k_p, \gamma - \gamma_d, \zeta_{\gamma_1}) \]

where \( k_{d}, k_{p}, \zeta_{\gamma_2}, \zeta_{\gamma_1} \) are positive constant, \( \gamma_d \) is the desired value and \( \text{sat} \) is the saturation function defined as

\[ \text{sat}(s, \zeta) = \begin{cases} \zeta & \forall s > \zeta \\ s & \forall -\zeta \leq s \leq \zeta \\ -\zeta & \forall s < -\zeta \end{cases} \]

To ensure the convergence to zero, the next inequalities have to be validated, see Sanahuja et al. (2009)

\[ \zeta_{\gamma_2} > \zeta_{\gamma_1} \]
\[ k_{d}^2 > k_{p} \]

(3)

Introducing (2) into (1d), (1e) and (1f), it follows that \( \ddot{\gamma}, \dot{\gamma} \to 0 \) y \( \gamma \to \gamma_d \).

5. EXPERIMENTAL RESULTS

One of the main achievements of this work is the experimental validation of the developed platform. Several experiments have been designed and executed in order to know the performance of the platform. The application is developed in two steps; first, the goal is to stabilize the orientation of the helicopter close to the floor (the maximum distance will be the height of the spinning top structure). In this part, the control parameters will be tuned. Once this task completed, the helicopter will receive more ‘gas’ in order to make it hover.

Stabilization of the mini helicopter in the spinning top configuration

For security reasons, the pitch and roll angles were firstly stabilized. The desired values for pitch and roll angles were \( \theta_d = \phi_d = 0^\circ \). Once these tasks were completed, the yaw values were tuned. At first, \( \psi_d = 0^\circ \), the yaw control parameters were computed, carrying out a new experiment with different desired values for the yaw angle using the following sequence

\[ \psi_d = \begin{cases} 0^\circ & t < 40s \\ 40^\circ & 40s \leq t < 45s \\ -20^\circ & 45s \leq t < 53s \\ 0^\circ & 53s \leq t < 83s \\ 40^\circ & 83s \leq t < 88s \\ -20^\circ & t > 88s \end{cases} \]

The control parameters values are shown in Table 2. Notice that these gains do not assure the fulfillment of condition (3). The helicopter contains an electronic circuit board with three gyros introducing a gyro stabilization damping into the system enabling the quad-rotor to be manually controlled. Deactivation of these components means degradation of the dynamic performance of the vehicle in such a way that it is very difficult for a pilot to manually control the rotorcraft. However, gyro stabilization represents only an angular speed feedback, \( \ddot{\gamma} \), which is accomplished by adding it to the derivative control gain, \( k_d \), generating a total derivative gain \( k_d + k_d \), in the controller.

Table 2. Control parameters values

<table>
<thead>
<tr>
<th>( k_{\psi} )</th>
<th>( k_{\phi} )</th>
<th>( \zeta_{\gamma_1} )</th>
<th>( \zeta_{\gamma_2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi )</td>
<td>0.002</td>
<td>0.012</td>
<td>0.4</td>
</tr>
<tr>
<td>( \theta )</td>
<td>0.002</td>
<td>0.012</td>
<td>0.4</td>
</tr>
<tr>
<td>( \psi )</td>
<td>0.002</td>
<td>0.026</td>
<td>0.4</td>
</tr>
</tbody>
</table>

On the other hand, the good performance of the controller when controlling in real-time the orientation of the helicopter can be observed in Figures 4 - 6. Notice from Figure 6 the good behavior of the proposed control scheme when changing the desired values in the yaw angle.

Note also, from these figures, the good performance of the proposed platform. This fact can be observed when changing the desired values or parameters in the control...
strategy, the closed-loop system remaining stable. It is important to remark that the platform works with priority tasks, giving the highest priority to the control stability.

**Stabilization in free flight**

This experiment is called semi-autonomous flight because the orientation is autonomously controlled by the ground control unit, while the translational movements are controlled by the pilot.

The control objective is to manually move the helicopter to a desired height, to keep it in a certain distance from the floor without using any feedback height control input.

The control parameters are shown in Table 2. Figure 7 shows a picture of the helicopter hovering whilst in Figures 8 - 10 the good performance of the proposed experiment can be observed. Notice that the pitch and roll errors are smaller in free flight than in the spinning top structure. This fact is due to the friction produced when the spinning top structure slips. In these figures, the dotted lines describe the desired value, while the solid line represents the measured value.

Similarly that in previous experiment, the desired yaw angle is changed using the following sequence:

\[
\psi_d = \begin{cases} 
0^\circ & t < 65s \\
40^\circ & 65s \leq t < 70s \\
-20^\circ & 70s \leq t < 74s \\
0^\circ & 74s \leq t < 85s \\
80^\circ & 85s \leq t < 95s \\
0^\circ & t > 95s 
\end{cases}
\]
Notice that $\psi_d$ at $t = 85s$ changes abruptly and this fact can make unstable the system. Now, observe in Figure 10 that the system remains stable. This point is realized due to the priority given by the platform when executing the control law and also due to the velocity when sending the control signals.

6. CONCLUSION

A new test-bed platform has been presented in order to validate and improve the control for aerial vehicles. The platform is based on a ground control unit and an aerial system. Linux RT and Partikle have been used as OS in order to implement the control strategies and to guarantee priority tasks. The proposed platform was validated by controlling the attitude of a helicopter with four rotors. Several tasks (modification of the control parameters and the desired values on line) were executed in order to perturb the system, and the experimental results have shown the good performance of the platform by stabilizing the helicopter even in presence of abrupt changes in the desired values.

REFERENCES


Fig. 10. $\psi$–angle response. Notice that $\psi_d$ at $t = 85s$ changes abruptly and this fact can make unstable the system. Now, observe in Figure 10 that the system remains stable. This point is realized due to the priority given by the platform when executing the control law and also due to the velocity when sending the control signals.