Safe flying for an UAV Helicopter

A. Mancini  F. Caponetti  A. Monteriù  E. Frontoni  P. Zingaretti  S. Longhi
DIIGA, Università Politecnica delle Marche, Ancona, Italy

Abstract—Today small autonomous helicopters offer a low budget platform for aerial applications such as surveillance (both military and civil), land management and earth sciences. In this paper we introduce a prototype of autonomous aerial vehicle, the Helibot helicopter, specifically designed for applications in cooperative networks. Fundamental steps in the design process of an UAV are shown. We also present work in progress in the field of failure detection and a novel idea to the problem of failure recovery using a terrain vision system.

Index Terms—flying/aerospace systems, navigation and planning, sensor fusion, mobile robotics, failure detection, vision.

I. INTRODUCTION

During last years, Unmanned Aerial Vehicles (UAVs) started to dominate the scene of mobile robotics. The first idea of unmanned vehicle dates back to the First World War (Lawrence Sperry aerial torpedo), even if a real prototype was made only at the beginning of 1960s (US Army Firebee I) [1]. The military technology already permits to make great sophisticated tactical UAVs, which, however, are not accessible for civil purposes. This limitation can be now overtaken due to the possibility of having an UAV with a low budget [2]. So, for example, UAVs can be used in civil applications such as traffic monitoring and surveillance, emergency service assistance, rescue, land management, home security and earth sciences.

In addition to power management to extend the duration of flight, the main problems related to an UAV are full autonomy during a missions and capability of interaction between UAVs. Generally, an UAV must be fully autonomous; the autonomy is accomplished by a complex interconnection of systems, related to flight control, navigation and task-based planning, elaboration of sensor signals, software architecture for reactive behaviours, communication. In first prototypes, all automation hardware was on-board and the system did not required ground-to-helicopter communications, other than the standard remote control transmitter to control the helicopter in case of an emergency [3]. A major challenge is the ability to insert UAVs, together with Unmanned Ground Vehicles (UGVs), in cooperative networks to accomplish a specific task a priori defined. For example, rescue missions are often too hard and dangerous for human beings, particularly in extreme environments like devastated areas or fires. Today, autonomous rescuers are used in these cases; they cannot substitute the human operate, but they can cooperate with humans to carry out the rescue task. A fleet of heterogeneous UAVs can greatly help in many tasks, such as fire detection [4] or aerial surveillance [5]; this requires adopting a framework for cooperative activities. Cooperation is fundamental for a lot of reasons: a set of cooperative UAVs increases the knowledge of mission area, ensures the sharing of information and offers an high grade of redundancy in failure cases.

In this paper we introduce and overview the main characteristics of a prototype of autonomous aerial vehicle working in cooperative environments: the Helibot helicopter, based on the commercial solution Bergen RC Twin Observer. Then we propose our approach to Fault Detection and Isolation (FDI) working in cooperation with a terrain vision system.

The process that transforms a radio controlled industrial helicopter into an UAV is neither straightforward or simple. The first phase of the project, here described, was focused on methodological aspects, identification of sensors to be installed on board, hardware architecture, communication infrastructure, and other minor technical aspects.

II. METHODOLOGICAL ASPECTS

The goal of our project is to achieve complete autonomy of the helicopter; autonomy for take off, landing, flight without a human pilot or a remote control. A human operator assigns the task to the UAV, which then, in case cooperating with other UAVs or UGVs carries out a specific mission autonomously.

Undoubtedly, the identification of methodological aspects was the first activity of our project. The characteristics and the reliability of proprioceptive and eteroceptive sensor systems were analyzed to choose the proper sensors to be installed on board of vehicles. In particular, the characterization of uncertainties and the accuracy of sensors readings were investigated. The uncertainty was characterized both through a model-based analytical description and through a probabilistic approach.

In this first phase a multiple-sensors system was developed to overcome limitations inherent in any single sensor device. Different sensor systems were analyzed. Among the proprioceptive sensors, inertial, odometric, altitude, inclination and velocity sensor systems were investigated for being integrated with dead-reckoning systems. Among the eteroceptive sensors, vision, laser and sonar sensor systems were analyzed for being integrated with GPS systems.

Fault-Tolerant methods, which monitor the navigation systems for sensor faults in real time, can contribute significantly to improve system reliability and flight safety. Quick detection and isolation of sensor faults can prevent serious damages and possible irreparable consequences [6, 7]. The fault diagnosis technique known as ”structural analysis” was used to detect and isolate navigation sensor faults of the helicopter. This FDI technique is model-based and relies on the analytical redundancy. Analytical redundancy relations are equations that are deduced from the analytical model of the system, which uses only measured variables as input. These relations permit to distinguish between healthy and faulty operations of the system. The structural analysis is performed on the vehicle model to analyse its structural properties, and thus, it provides
Fig. 1. The Bergen Twin Observer Helicopter with the pan tilt system shown in foreground; the camera is not mounted.

the set of redundancy relations necessary for fault diagnosis [8].

A specific analysis was devoted to the use of artificial vision systems and sensor fusion techniques. Various approaches were studied according to the sensors considered. In particular, in the case of radiometric monochromatic or coloured images the study was addressed to the appearance based techniques Principal Component Analysis (PCA) and Weighted Walkthroughs (WW) [9], while in case of depth (range) images classical image processing techniques were analysed to allow the rebuilding of 3D models of the environment [10].

III. THE HELICOPTER

As previously mentioned, our UAV is a customisation of the commercial solution Bergen Twin Observer. In Fig. 1 the physical system is represented.

A great characteristic of this helicopter is the payload, which can reach up to 9 kg. This characteristic is fundamental and permits to install a variety of sensors and devices, increasing the capabilities of the UAV. The main characteristics of Bergen Helicopter are listed in Table I.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>1.52m</td>
</tr>
<tr>
<td>Height</td>
<td>0.50m</td>
</tr>
<tr>
<td>Weight</td>
<td>9kg</td>
</tr>
<tr>
<td>Main Rotor Span</td>
<td>800 mm</td>
</tr>
<tr>
<td>Tail Rotor Span</td>
<td>130 mm</td>
</tr>
<tr>
<td>Fuel Tank</td>
<td>950ml</td>
</tr>
<tr>
<td>Engine</td>
<td>Twin Cylinder 52cc</td>
</tr>
<tr>
<td>Max Payload</td>
<td>9kg</td>
</tr>
<tr>
<td>Autonomy</td>
<td>30min (hovering)</td>
</tr>
</tbody>
</table>

TABLE I
MAIN CHARACTERISTICS OF BERGEN HELICOPTER

Thanks to the reduced dimensions of the helicopter, its landing (take off) can be made also on (from) a mobile platforms. Using an auxiliary fuel tank, we have extended the maximum flight duration up to 60-70 minutes, as short autonomy reduces the set of possible missions.

To extend the field of view of the vision sensor, the system is provided with a pan-tilt system fully controllable by the on board computer or remotely by the pilot.

We also modified the landing gear. The reason was simple: basic movements, like hoovering at very low altitude, forward/backwards and lateral, are usually tested during the first flight; these manoeuvres, even if quite simple, are very hard using automatic control, because the control laws are still embryonal. Using the modified landing gear, the physical integrity is well preserved.

IV. ARCHITECTURE

We derived the architecture of our system from the analysis of methodological aspects. It can be splitted into three elements:

- Avionics;
- Communication;
- Ground Control Station.

A. Avionic Architecture

Avionic architecture is based on connections among the following four subsystems:

- Flight Computer
- Sensors
- Navigation software
- Automatic Pilot Interface

The avionic subsystem was studied and fully implemented, so that processes like navigation and localisation are possible; these tasks are at the lowest level. Every higher level application uses the basic functions provided by the avionic subsystem.

The first component we introduce is the on board flight computer; in this case, the constraints to be respected are the Electromagnetic interference (EMI) immunity, resistance to mechanics vibrations and the continuity of operation in a variable environment (temperature, humidity,...). We have chosen a robust solution as the PC104 produced by Eurotech. The computer consists of a Pentium III 800MHz CPU, a solid state hard disk and a set of expansion modules installed to increase the number of I/O ports (RS232, CAN, USB and Ethernet/Gigabit for remote debug).

The localisation process is the heart of the avionic subsystem. We use two kinds of sensors: an inertial sensor and an absolute position sensor like GPS. For the inertial sensor we use a low cost strapdown Attitude Heading Reference System (AHRS); the AHRS 3DM-GX1 produced by Microstrain Inc. is the device we adopted. This device (using calibration values) outputs data as accelerations, angular rates, magnetic field intensity, roll, pitch and yaw angles. Using a navigation algorithm we get information about position and attitude of our helicopter. Many tests were conducted to determine the quality of dead reckoning processes. Good results were obtained for a short time range; an advanced error model of sensor can be developed to extend the time range. The use of GPS is strongly recommended to correct periodically the position and attitude of the helicopter. The GPS receiver we used is the GPS-41EBF3V, 12 channel, with WAAS and EGNOS (available in
Europe) correction. We did a characterisation of GPS receiver installing it on a geodetic reference point (the tower of our faculty); when EGNOS corrections are available the quality of localisation is pretty good and comparable with the DGPS performance. A problem that we discovered is the reduction of the numbers of visible satellites when the GPS antenna is not well oriented; this problem can be fixed installing a helicoidal GPS antenna, which is better than the classical patch antenna.

Another sensor mounted on board is a pressure altimeter based on a MPX4100A sensor produced by Freescale Semiconductor. This kind of sensor is mandatory because the altitude estimation from the GPS sensor has a high variance; this aspect makes the difference between a crash and a correct flight. As all altimeters, this sensor must be calibrated at every flight. The information provided by sensors just introduced must be fused to obtain a better estimation of position and attitude. A difficulty consists in the different update rates of each sensor. The AHRS is the sensor with the highest update rate (up to 100Hz), the altimeter and GPS work with 1Hz. This is not a hard obstacle. We are testing various fusion algorithms, based on the Kalman filter to improve the performance and quality of navigation and localisation processes [11, 12, 13, 14].

Another important subsystem is the interface that manages the switching from manual control to automatic control. Manual control is performed by a human operator with a remote radio controller (Futaba T12). The human operator can enable the automatic pilot mode sending a command via the radio controller. An embedded electronic circuit, based on a microcontroller, detects the command and switches the control from the radio controller receiver to the servo controller controlled by the on board computer. This, on the base of control law, at every sampling time, controls the servos using a Pololu controller.

The first choice in the software architecture was that of the Real Time Operating System (RTOS). We analysed and tested two solutions based, respectively, on RTAI and QNX. Owing to the educational program provided by the Italian QNX distributor we chose QNX; in fact, in addition to a hard real time support, it is provided with a very powerful Software Development Kit (SDK). Respecting the philosophy of QNX, we structured the applications into cooperating processes that communicate using Inter Process Communication (IPC). Fig. 2 shows the philosophy of our applications for UAV control and sensor acquisition.

A resource manager runs in protected memory user space; it can be started, stopped and debugged dynamically. Its main characteristic is the POSIX compliance. A client application that would use a resource manager calls the open() function as a file. In this way, using a standard interface, GPS, AHRS, radio modem, altimeter, servo controller, ... are structured as resource managers. A real time software timer was developed into a timing process. The main aim of this one is the synchronisation; at every pulse, generated after \( T_c \) seconds (the frequency of low level UAV control), the timing process sends a pulse via IPC to the navigation process, which will then pass from the sleep to the run state. We remind that the aim of the navigation software is the localisation of UAV.

\[ \text{Fig. 2. Every external entity, sensor or module, is interfaced by a driver; in the QNX language a device driver is known as a Resource Manager.} \]

B. Communication Architecture

Communication is another fundamental subsystem. We have chosen a low bitrate solution because we want to share only useful information for cooperative task. Usually these ones are strongly structured and require little bandwidth. This approach permits to reduce the costs of the communication subsystem. We can distinguish almost two kinds of communication:

- UAV to ground control station
- UAV to UAV/UGV

The first case is represented by the telemetry, whereas the second one is the data exchange in cooperative scenarios. Telemetry data are necessary to monitor in real time the main parameters of the flight. Data are stored by the Ground Control Station (data logger).

The second case, as already mentioned, is useful in cooperative missions where two or more UAVs are involved. Data are sent using an advanced radio modem that transmits and receives on the 868MHz hand. The radio modem we adopted is produced by Aerocomm; in particular, the module is the AC4848-250M with 250mW RF power output. The communication range can reach up to 15km when the antennas are line of sight. The main characteristic is the possibility to create complex networks using two or more modules. Architectures as point to point, point to multipoint, multipoint to multipoint and peer to peer are available; these are very useful in cooperative scenarios. The information sharing between agents is really easy and reliable.

C. Ground Control Station Architecture

The Ground Control Station, has a lot of capabilities among which telemetry data acquisition and data logger for post flight analysis. In a cooperative scenario it is responsible for mission and task allocation and supervision. Data are collected and sent using the same radio modules installed on board of UAVs.

V. Modelling and Testing

First tests generally are the most dangerous; a lot of unexpected problems arise as EMI and mechanical breaks. For this reasons, we designed and then made a mechanical structure.
to limit the risk of crashes while performing tests. It resulted very useful for tuning the parameters of some controllers, testing new electronic circuits and so others. The movements allowed are rotation and translation around \( z \) axis (yaw). This mechanical structure is shown in Fig. 3.

A fundamental component is the **Avionic Box**, so called because contains all the necessary hardware for navigation and control. This box permits to shield the internal devices from the external RF disturbances, especially coming from the sparks. We cut down the weight of the box using light material as aluminium. Fig. 4 shows the avionic box.

**VI. FAULT DIAGNOSIS**

A conceptual structure of fault tolerant control, widely accepted by scientific community, comprises four main stages (see Fig. 5): the plant itself (including sensors and actuators), the FDI unit, the feedback controller and the supervision system. Generally speaking, the plant is assumed to have faults in sensors, actuators or other components. The FDI unit provides the supervision system with information about the onset, location and severity of any fault. The supervision system will reconfigure the sensor set and/or actuators to isolate the faults, and it will tune or adapt the controller to accommodate the undesired fault effects.

The core of a general fault tolerant control scheme is the FDI unit, as shown in Fig. 5. This work focuses on FDI unit. Application of a model-based technique, known as Structural Analysis [15], permits to design a FDI unit for the identification and isolation of Helibot’s actuator faults.

Helibot’s actuators are five servos which permit to command the engine throttle, tail rotor pitch, collective and cyclic pitch of main rotor (one servo for collective, two servos for lateral and longitudinal cyclic). If one of the last four servos fails for some reason, it could be still moved by the remaining three servos. Thus, if the servo fault is identified and isolated, the Helibot can continue to safely move through the autopilot or human pilot. The FDI unit has been designed to detect and isolate faults in all four helicopter actuators: main rotor collective, tail rotor collective and both pitching and rolling cyclic inputs. In this study, faults have been assumed as stuck of the considered servos. A faulty servo is unusable. In a faulty situation, once the FDI unit will detect and isolate the faulty servo, the supervision unit sends a signal to disable the faulty servo and for carrying back it on its neutral position, thus letting the other servos to work. This permits to avoid a hard or crash landing in a servo fault situation.

The proposed FDI scheme is shown in Fig. 6 where \( u(t) \) and \( y(t) \) are inputs and outputs vectors of the helicopter, respectively, and \( \hat{y}(t) \) is the estimation of the Helibot’s outputs.

**A. Residual generation**

The helicopter residual generation is based on the available non-linear mathematical model of the Helibot. The proposed FDI scheme introduces the FDI functions using two modules: a residual generation module and a residual evaluation module. The residual generator module is able to generate the residuals...
exploiting the main ideas of the structural analysis [16, 15]. This technique has been widely applied in robotics [16, 17, 6, 8, 7]. The observer residual generation is based on the available information coming from the observer. A linear mathematical model of the Helibot has been derived by means of well known identification scheme, choosing the Auto Regressive eXogenous (ARX) input-output model. Therefore, a Luenberger observer has been developed to estimate the Helibot outputs. The residuals are generated as comparison between the actual outputs of the helicopter, and the estimated outputs, provided by the observer.

B. Residual evaluation

The residual evaluation module detects a change in the mean of an observed and distributed random sequence obtained by a sequential change detection algorithm. Thus, residual evaluation reduces to the problem of detecting this change in the residuals. A particle filtering based decision module has been proposed and applied to estimate the probability density functions of each residual (pdfs). 

Let \( r_i (kT_s) \) be the \( i \)-th residual with pdf \( p_0 (r_i) \) depending upon one scalar parameter \( \theta \), which is the mean value of the residual. Before an unknown change time, \( k_0 T_s \), \( \theta \) is equal to \( \theta_0 \). At time \( k_0 T_s \), it changes to \( \theta = \theta_1 \neq \theta_0 \). The hypotheses are:

\[
\mathcal{H}_0 : \theta = \theta_0 \quad \text{for} \quad \forall k
\]

\[
\mathcal{H}_1 : \theta = \theta_0 \quad \text{for} \quad k \leq k_0 - 1 \quad \text{and} \quad \theta = \theta_1 \quad \text{for} \quad k \geq k_0
\]

Consider the logarithm of the likelihood ratio of an observation \( r_i (kT_s) \), which is a function of random variable \( r_i (kT_s) \), defined by:

\[
s (r_i (kT_s)) = \ln \frac{p_{\theta_1} (r_i (kT_s))}{p_{\theta_0} (r_i (kT_s))}
\]

where \( p_{\theta_0} (r_i (kT_s)) \) \((b = 0, 1)\) is a pdf parameterized by \( \theta_0 \). The key statistical property of this ratio \[18\] with \( E_{\theta_0} \) and \( E_{\theta_1} \) denoting the expectations of the random variables with distributions \( p_{\theta_0} (r_i) \) and \( p_{\theta_1} (r_i) \) respectively, is reflected through:

\[
E_{\theta_0} = \int_{-\infty}^{+\infty} s (r_i) p_{\theta_0} (r_i) \, dr_i < 0
\]

\[
E_{\theta_1} = \int_{-\infty}^{+\infty} s (r_i) p_{\theta_1} (r_i) \, dr_i > 0
\]

Therefore, any change in parameter \( \theta \) is reflected as a change in the sign of the mean value of the LLR \[18\]. Moreover, observations \( r_i (kT_s) \) are independent of each other; the joint LLR for observations from \( r_i (lT_s) \) to \( r_i (kT_s) \) may be expressed as:

\[
S^k_i = \sum_{j=l}^{k} \ln \frac{p_{\theta_1} (r_i (jT_s))}{p_{\theta_0} (r_i (jT_s))}
\]

This joint LLR \( S^k_i \) shows a negative drift before change, and a positive drift after change. This behavior is used for detecting any change between two known pdfs \( p_{\theta_0} (r_i) \) and \( p_{\theta_1} (r_i) \). Note that the pdfs \( p_{\theta_0} (r_i (jT_s)) \) and \( p_{\theta_1} (r_i (jT_s)) \) are non-Gaussian and have to be estimated on-line \([18, 19, 20]\). The particle filter based approach found in \([21, 22]\) is used to estimate both pdfs.

Let us assume that the normal behavior and all possible actuator faults can be described by a given finite set of linear stochastic state space models indexed by \( m = 0, 1, \ldots, M \):

\[
x_i^{(m)} ((k + 1) T_s) = x_i^{(m)} (kT_s) + \sum_{l=0}^{M} a_i^{(m)} x_l^{(m)} (kT_s) + v_i^{(m)} (kT_s)
\]

where state \( x_i^{(m)} (\cdot) \) is the normalized sensor fault magnitude, \( v_i^{(m)} (\cdot) \) is the nominal residual without faults independent of past and present states and \( a_i^{(m)} \) is the estimation of the actuator fault magnitude obtained from experimental results.

The central idea of the proposed method is to compute the joint likelihood of the observations conditioned on each hypothesized model through Monte-Carlo estimation that uses the complete sample-based pdf information provided by the particle filter, and then activating in parallel \( M = 1, 2, \ldots, M \) versus \( \mathcal{H}_0 \). Specifically, the joint LLR to be computed in this case is:

\[
S_k^i (m) = \sum_{l=1}^{k} \ln \frac{p_{\theta_1} (r_i (jT_s) \mid \mathcal{H}_m, Z_{j-1})}{p_{\theta_0} (r_i (jT_s) \mid \mathcal{H}_m, Z_{j-1})}
\]

where the likelihood of the observation \( r_i (jT_s) \) gives its past values \( Z_{j-1} = \{ r_i (T_s), r_2 (2T_s), \ldots, r_i ((j - 1) T_s) \} \), i.e., \( p (r_i (jT_s) \mid \mathcal{H}_m, Z_{j-1}) \) \((m = 0, 1, \ldots, M)\) is the one step output prediction density based on \( \mathcal{H}_m \) defined by the \( m \)-th measurement model and the known statistics of \( v^{(m)} (jT_s) \). Hence, the decision function \( g_i (kT_s) \) may be obtained as:

if \( \max (0, S_k^i (m)) = 0 \) accept \( \mathcal{H}_0 \) and set \( g_i (kT_s) = 0 \)

if \( \max (0, S_k^i (m)) > 0 \) accept \( \mathcal{H}_1 \) and set \( g_i (kT_s) = +1 \).

VII. HELICOPTER NAVIGATION BASED ON A TERRAIN VISION SYSTEM

This paragraph presents the work in progress on vision navigation and control of an autonomous helicopter given only measurements from a camera fixed on the ground. The goal is to develop an alternative to traditional INS/GPS and on-board vision aided systems. A typical application would involve landing an UAV simply using a camera fixed on the landing pad. This approach can be useful in case of failure of some device of the on board avionic. From another point of view, the on board avionics can be greatly simplified (i.e., no IMU), the approach may present significant cost benefits for controlling small UAVs. The state estimation combines the vision measurements with the vehicle’s dynamic model in a recursive filtering procedure based on Extended Kalman Filter (EKF). The estimation of the helicopter’s current state (position, attitude, velocity, and angular velocity) is fed back in real-time to the base station controller to generate radio control commands to the helicopter. Estimation of the vehicle’s state requires formulating a process model of vehicle motion, a measurement model of the video observations, and an inference algorithm to calculate the state estimation given the
models and observations. The model of vehicle motion for the vision-only navigation is different to that of the traditional INS/GPS kinematics equations. Since the inertial measurement unit is absent for vision-only navigation, direct measurements of the helicopter’s accelerations and angular velocities are not available. Two possible models are investigated. The first uses a simple random walk model, which approximates accelerations by stochastic processes. This model is similar to what is used in classic vision-based tracking applications. However, since we dispose of a dynamic model of the vehicle, which we used for the control design, we may also employ this full model for state estimation [23, 24].

The measurement model uses point features derived from the sequence of video images. Feature points are detected using an improved real-time SIFT approach [25, 26]. Expected feature point locations are found using the known geometry of the pre-defined feature points on the helicopter projected from 3D to the 2D camera plane using a perspective projection parameterized by the helicopter’s current position and attitude. The differences between the corresponding points are used to update the state estimation. Combining a motion model with a set of observations may be performed with an EKF.

All experiments for this initial work were carried out in simulation. This provides more flexibility for evaluations as well as the ability to implement the closed-loop feedback control given the vision-based state estimation (real-time closed-loop operation with the actual flight vehicle had not been implemented). Comparisons to a traditional INS/GPS system for state-estimation showed very similar characteristics. This is still a work in progress and further experiments will confirm the usefulness of this approach as a system to recover from possible failures to INS/GPS system and for bringing the helicopter to land.

VIII. CONCLUSIONS

In this paper we have presented our prototype UAV helicopter for cooperative tasks. The main peculiarities of our prototype are in the communication oriented architecture, which, as well as the software infrastructure, demonstrated to be fully scalable and robust. Besides, despite the helicopter is provided with a variety of sensors, the actual weight of the avionic box is lower than maximum payload, so that further sensors could be added to enhance the performance of the system. Then we have introduced our approach for safe flying, based on an FDI technique for the emergency recovery of the helicopter and on a terrain vision system to aid the UAV during landing in a critical situation consequent to some failure.

Future works will be focused on the development of more tests to try new control laws and identifying the parameters of the mathematical model [23, 24] by using neural networks [27]. However, the prototype supported well preliminary tests and so it can already be used in task oriented applications.

REFERENCES


