VIRTUAL AUTOPILOT SYSTEM FOR HELICOPTERS UAV MISSIONS

Juan Manuel Lema, Pablo Royo, Enric Pastor, et al.
Technical University of Catalonia

Keywords: *UAS, Rotorcraft, Autopilot*

Abstract

Selecting the right autopilot to be integrated in a given UAS to develop a certain mission is a complex task because none of them are mutually compatible. Moving from one autopilot to another may imply redesign form scratch all the remaining avionics in the UAS.

This paper presents the Virtual Autopilot System (VAS), an intermediate subsystem added to the UAS platform to abstract the autopilot from the mission and payload controller in a UAS.

The VAS is a system that on one side interacts with the selected autopilot and therefore needs to be adapted to its peculiarities. On the other side, interacts with all the architecture offering standardized information of the autopilot, and consuming mission and payload orders.

1 Introduction

Unmanned Aircraft Systems (UAS) have been initially used by the military sector for terrain recognition, search and rescue, transportation, among others. During these previous years, there has been a new special interest by the aeronautic society in using these systems for civil applications. Terrain mapping, power line inspection, volcano monitoring, forest fire fighting are just few examples that may be found at the INOUI document [5] of this applications.

There are three different types of UAS that can implement these applications: the fixed wing UAS, rotary wing UAS, and morphing UAS. This last can fly either as a fix wing or a rotary wing UAS. Each type of UAS has its advantages that can be useful for the mission. For example, while the fixed wing can reach to higher speeds, the rotary wing has the ability to hover on a waypoint. Both kinds must be controlled by an autopilot in order to fly autonomous.

There are several commercial autopilots at the market specially designed for working with Unmanned Aircraft Systems (UAS) such as the AP04[1], Piccolo[2], or MicroPilot[3]. It has been detected that most autopilots have similar functionalities and behavior: they send the telemetry, implement flying states, receive commands from the ground station, and manage some of the payload onboard. But each autopilot has its own way of implementing these features. The telemetry is sent in its own format and ratio, the connection link is not standardized, and the flying states depend on the company interests. Therefore the system surrounding the autopilot is usually adapted to it. However, selecting the right autopilot to be integrated in a given UAV is a complex task because none of them is mutually compatible. Moving from one autopilot to another may imply redesigning from scratch all the remaining avionics in the UAS.

In order to solve this problem the ICARUS group designed the **Virtual Autopilot System (VAS)** [4]. It is specially designed to operate as an interface between the autopilot and the mission and payload components in a UAS. It works under a UAS framework for defining and standardizing all the UAS components. This framework is called **UAS Service Abstraction Layer (USAL)** and it is specially designed for UAS civil

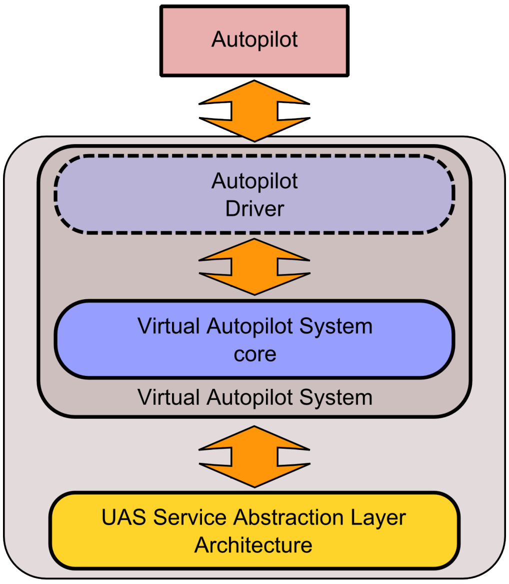

Fig. 1 VAS architecture.

missions.

The VAS is a system that the on one side interacts with the selected autopilot and therefore is adapted specifically to it, and on the other side interacts with all the UAS architecture offering standardized information of the autopilot, and consuming mission and payload orders. VAS operates similarly to drivers in an operating system, abstracting away the implementation details from actual autopilot users. When changing the autopilot (see figure 1) there is no need to redesign the entire platform, just to create the driver for this new autopilot.

The VAS is designed for working with fixed wing autopilots. Since the flight performed by a fixed wing UAS is different from a rotary wing UAS, they need autopilots specially implemented for their features. In order to expand VAS functionalities there is a proof of concept of the integration with a rotary wing UAS.

This paper presents the changes needed on VAS [4], in order to work with rotary wing UAS.

2 System Overview

For executing UAS civil missions we use a distributed embedded system [6] that is on board the aircraft and that operate as a payload/mission controller. Over the different distributed elements of the system we deploy software components, called services, which implement the required functionalities. These services cooperate for the accomplishment of the UAV mission. They rely on a middleware [7] called Middleware Architecture for Remote Embedded Applications (MAREA) that manages and communicates the services. The communication primitives provided by MAREA promote a publish/subscribe model for sending and receiving data, announcing events and executing commands among services.

Providing a common infrastructure for communicating isolated UAS services is not enough for keeping the development and maintenance costs for UAS systems low. The existence of an open-architecture avionics package specifically designed for UAS may alleviate the developments costs by reducing them to a simple parameterization. From the study and definition of several UAS missions, one can identify the most common requirements and functionalities that are present among them [8]. The UAS Service Abstraction Layer (USAL) is the set of available services running on top of the UAS architecture to give support to most types of UAS missions [4]. USAL can be compared to an operating system. Computers have hardware devices used for input/output operations. Every device has its own particularities and the OS offers an abstraction layer to access such devices in a uniform way.

The VAS is just one of a set of services defined at the USAL architecture. It is the one in charge of dealing with the autopilot features and functionalities. But there are other services that work together developing other important tasks. Figure 2 shows one possible configuration of the platform where four different services are interacting to perform a certain mission.

The Flight Monitor Service (FMS) is the part which interacts with the on-ground operator in order to track, or command the UAS in real time. The FM may adapt the functionalities depending on the operator's profile: pilot oriented, central controller oriented, dispatcher oriented and maintenance oriented.

The Flight Plan Manager Service (FPMS)

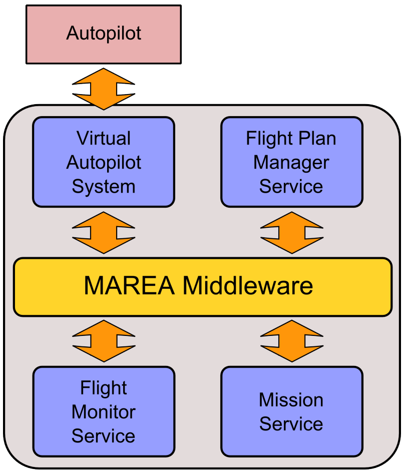

Fig. 2 USAL platform.

[9] has been designed to implement much richer flight-plan capabilities on top of the available capabilities offered by the actual autopilot. The FPMS offers an almost unlimited number of waypoints, waypoint grouping, structured flight-plan phases with built-in emergency alternatives, mission oriented legs with a high semantic level like repetitions, parameterized scans, etc.

Finally, in the figure 2, it is shown the Mission Service. This module is in charge of setting and configuring mission oriented information. Even though the FM gives a path to be followed by the aircraft, the MS is the one that sets the orders to take photos, process the data, or any task referred to the mission the UAS is developing.

As it may be seen, the VAS turns to be an essential part of the architecture. Since it is the one that deals with the autopilot it must correctly interpret all the USAL information and translate it to the hardware.

3 Virtual Autopilot System

The inclusion of the VAS improves the flexibility of the system. The autopilot unit can be replaced by a new version or a different product, but this change will have no impact on the system except for the VAS. Another important motivation is to provide an increased level of functionality. VAS should permit operation with a virtually infinite number of waypoints, thus overcoming a limitation present in all studied UAS autopilots. This increased level of functionality includes the capability to take control of the flight and generate new waypoints in response to contingencies when services in charge of navigation control fail.

UAS autopilots available today are similar in their operation and capabilities, though their implementation details greatly differ. The key to carry out a correct abstraction is to offer in the VAS interface the common functionality and data that can be found in any autopilot. On the first design of VAS [4] the purpose was to organize the information in the following four groups:

1. Flight Telemetry.
2. Navigation Information.
3. Status/Alarm Information.
4. Flight State Management.

This proposal has no impact when changing from fixed wing to rotary wing UAS. Therefore the VAS design will remain the same, but the implementation of each group will differ.

As shown in Figure 3, the VAS is organized into four groups. The telemetry group relates to the need of the autopilot to acquire and process attitude and position data. That is, the VAS provides the exploitation of the autopilot telemetry by other applications in the UAS. In this way, telemetry exploitation is not autopilot dependent. The second (Navigation Information) is needed to determine the path that the aircraft follows. This information group increases the autopilot navigation capabilities where before they were just a collection of statically defined waypoints. The Status/Alarm information gives information about its current autopilot and VAS status or alarms. Finally, the Flight State Management is added to the VAS design to provide the aforementioned increased level of functionality. This last group changes the autopilot states when necessary. Also, as displayed in the figure, monitoring and status/alarm information are outgoing flows, while navigation and state management have an input/output direction.

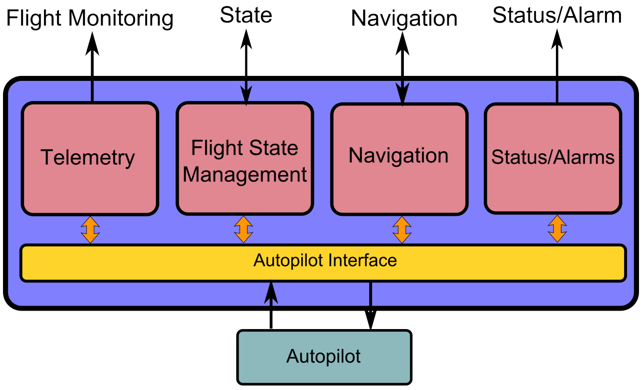

Fig. 3 VAS categories.

These four groups are the core of the VAS. Next there is an explanation of each part redefined for working with rotary wing UAS.

3.1 Flight Telemetry Category

Autopilot manufacturers group all this information in large packets of data, which are sent via a radio modem at a certain frequency. In our service-based architecture, the VAS service will offer this information over a LAN to all the services that need this information. The information will be semantically grouped in a way that this information relates to parameters, situations or attitudes of the aircraft, independently of the real autopilot hardware and sensors.

The telemetry used for implementing a mission barely changes when using fix or rotary wing UAS. This telemetry has to do with the position of the UAS, speed, angles, etc. Most of this information is common on both types of UAS.

But there are some differences on the interpretation of this telemetry. On a fixed wing the angles are indicating the velocity direction of the UAV. A rotary wing UAS can fly in a direction and be addressed to other, therefore the UAV angles defined for fix wing turns ambiguous. On a helicopter there are two important UAV angles: the one that specify the direction and the one that defines the orientation. This is one important difference of working with airplanes or helicopters.

Both UAS need to send the wind estimated information. In spite of that, the interpretation of these parameters differs since both types of UAS generate by itself different wind flow. This information must be treated taking into account which kind of UAS the system is working with.

The UAS Surface Control is a packet of data that cannot be generalized for both aircraft. Since servos on a fixed wing and rotary wing are not the same, this telemetry information must change. Therefore we cannot generalize the aileron, ruder, flaps or elevator, just because there are not present on all UAS as we did for a fixed wing. It is a good generalization when VAS works just with planes. Instead this packet will contain the roll, pitch, collective, tail and throttle for working with rotary UAS.

Table 1 Flight management information published by VAS.

Name	Composition	Unit	Description
uavAngles	Roll Pitch Yaw	radians	Roll, Pitch and Yaw angles of the UAS
uavAcceleration	X Y Z	m/s^2	Acceleration in UAS X, Y and Z axis.
uavRateTurn	X Y Z	rad/s	Rate of turn in UAS X, Y and Z axis
uavPosition	Latitude Longitude Altitude (MLS) Pressure Altitude	radians radians meters meters	3D UAV Position
uavSpeed	North East Down	m/s	3D speed in the UAV
uavAirspeed	Indicated airspeed True airspeed	m/s	Air speed data in UAV
windEstimated	North East Down	m/s	North, east and down wind speed estimated
missionTime	Time	ms	Mission Duration
uavSurfaceControl	Roll Pitch Collective Tail Throttle	Radians	Surface Control position

The packets of data defined in the telemetry section are only those useful for developing a UAS mission. When changing from a fix to a rotary wing UAS, since the flight performed vary some packets must be changed. There is common information, such as the acceleration, rate of turn or mission time, that can remain the same.

3.2 Navigation Category

At the USAL architecture, the Flight Plan Manager Service (FPMS) is in charge of generating the navigation commands to the VAS [9]. In most cases these commands will take the form of waypoints or requests for changing the autopilot state. The VAS feeds the autopilot with its internals waypoints as it consumes system waypoints and commands.

The next group of information is the output Navigation group. This information basically states where to the UAV is going at any moment, in which direction is moving and which waypoint is flying. Table 3 defines this information.

Table 2 Input navigation information by VAS.

Name	Composition	Unit	Description
qnhGround	Pressure	Pascals	Sets the QNH pressure for the pressure altimeter
gndLevel	Ground level altitude (MLS)	Meters	Set the ground level altitude to the autopilot
maxTimeMission	Time	ms	Set the mission time.
deflectSurfCont	Roll Pitch Collective Tail Throttle	UAV Range	Control packet for direct action on surface
NewWp	Latitude Longitude Altitude (MLS) Speed Fly Over Identifier	radians radians meters m/s N/A N/A	Read waypoint information where the UAS goes.
newUavSpeed	IAS	m/s	Set the indicated air speed of UAV
newUavAltitude	Altitude (MLS)	meters	Set the UAS altitude
newBearing	Direction	radians	Set the UAS bearing
newMainRwy	Latitude Longitude Altitude (MLS) Heading Length	radians radians meters radians meters	Set the coordinates of the main runway.
newAltRwy	Latitude Longitude Altitude (MLS) Heading Length	radians radians meters radians meters	Set the coordinates of the alternative runway.
changeVasState	State Type State Parameters	N/A N/A	Set the VAS state.
clearWps	Event	N/A	Clear all the flight plan waypoints
skipWp	Waypoint identifier	N/A	Skip one waypoint and pass to another.

The next group of information is the input Navigation group. This information basically tells the VAS configuration parameters for the autopilot operation, as well as parameters to configure the operative parameters of the states in which the VAS may operate. Table 2 defines this information.

Most of the navigation information has been designed for implementing the mission, and has little to do with the flight. Parameters such as waypoints, runways, or orders involving them are treated in this part of the VAS. All the events or functions invoked at the navigation part of VAS (table 3 and table 2) have to do with the USAL architecture, not with the autopilot. All this information will remain the same in spite of changeing the UAS.

Table 3 Output navigation information by VAS.

Name	Composition	Unit	Description
currentWp	Latitude Longitude Altitude (MLS) Number	radians radians meters	Waypoint information where the UAS goes.
previousWp	Latitude Longitude Altitude (MLS) Number	radians radians meters	Previous waypoint information.
rwySituation	Latitude Longitude Altitude (MLS) Heading Length	radians radians meters radians meters	Runway information from the autopilot.
altRwySituation	Latitude Longitude Altitude (MLS) Heading Length	radians radians meters radians meters	VAS alternative runway information.
uavDirection	True Airspeed Altitude (MLS) Bearing Heading	m/s meters radians radians	Current target true airspeed, altitude, bearing and heading.
vasState	(See Flight States Category)	N/A	Current VAS state.

3.3 Alarm Category

An autopilot is a complex hardware that needs to be monitoring every time. With the information defined in table 4 we can monitor the autopilot and the VAS status; when any part of these devices has a failure the VAS will send an alarm to the network. All of these alarms are sending by the network as events for two reasons.

Table 4 Status and Alarms Information.

Name	Description
gpsApAlarm	GPS Alarm.
outRangeTempApAlarm	Temperature outside range.
voltSysApAlarm	Bus voltage alarm (System).
voltServApAlarm	Bus voltage alarm (Servos).
accApAlarm	Autopilot acceleration alarm.
rateTurnApAlarm	Autopilot rate of turn alarm.
ImuApAlarm	Autopilot IMU alarm.
magnetometerApAlarm	Autopilot magnetometer alarm.
pressureAltimeterApAlarm	Autopilot pressure altimeter alarm.
anemometerApAlarm	Autopilot anemometer alarm.
missionAlarm	Mission time reached alarm.
wpRangeAlarm	Waypoint outside range alarm.
wpProcessAlarm	Error processing parameters.
lackMainRwy	None main runway.
speedAlarm	Speed outside range.

First, because the alarms are very important for the system and it is needed that these notifications safely arrive to all the services that process them. Second, we will only need to know the status when something is wrong.

3.4 Flight States Category

The commercial autopilots are much focused in flight states. But since there is no standard to follow every manufacture implements its own flying states. Usually they have the waypoint navigation, heading navigation and manual navigation. However a mission can be composed of many different states, for example, we can have different behaviors for the contingencies, which need different types of response. Many autopilots solve this sort of problems just coming back to base station. However, we want the UAS to be able to enter in safe states where it can try to recover the situation.

Figure 4 shows all the VAS states for rotary wing UAS. There have been some modifications from the VAS state proposal for fixed wing UAS. This section explains the important changes between the different proposals.

One important feature that a rotary wing UAS offers over fixed wing UAS is the possibility to hover on a location in the space. This can be a useful characteristic for the development of the mission. For example if a certain area must be explored by taking several photos over a waypoint, the hovering patter can be helpful. The previous VAS design has not contemplated this feature since it did not apply to aircraft UAS.

The same occurs when resolving a problematic situation such as running out of waypoints. The safe reaction state at the previous VAS design was to make a hold pattern. The goal in this state is to keep the fixed wing UAS near the last location, so that when a new waypoint was entered, continue the flight plan as similar as it initially was. When working with rotary wing UAS, it can stay still at the last waypoint hovering over it until new information is set.

The Hover state will have an altitude/velocity restriction since this technique can be very aggressive for the UAS. The user must be sure that the UAS can stay still at a point. If the velocity or the altitudes are not appropriate, and the Hover state is required, VAS will turn to Hold instead.

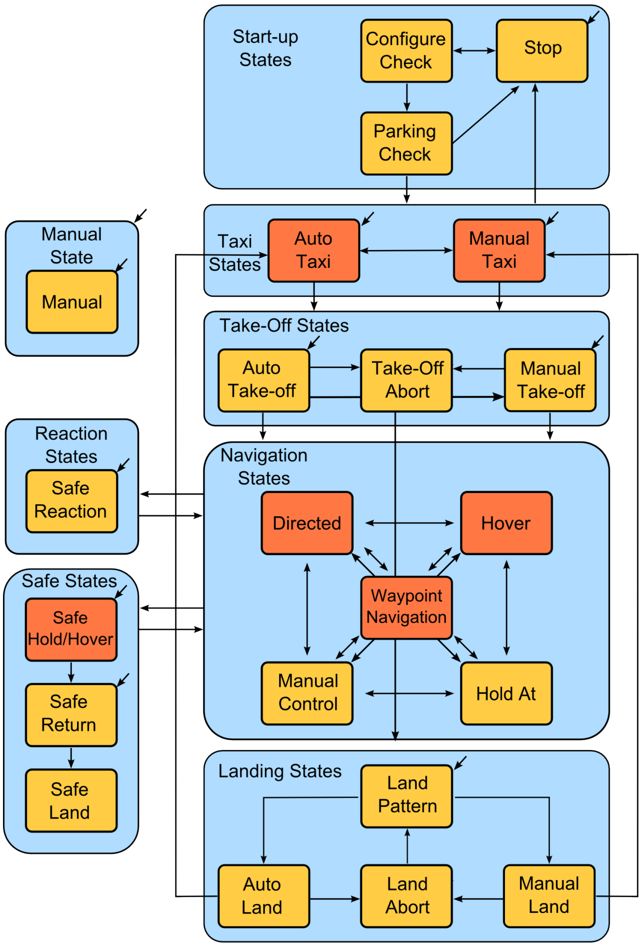

Fig. 4 VAS State Diagram.

The Directed State will also suffer some changes. On a fixed wing UAS the direction of the UAS is mostly defined by the yaw angle. There are special situations when the heading and the yaw are different, for example when there is strong wind. A helicopter may be flying in one direction but be addressed to another. This feature must be exploited by the VAS. The new Directed state will not just have the heading parameter. It will also specify the direction where the UAS will be following, and also the orientation.

The Taxi states defined at the previous VAS and the one presented in this paper implement different concepts. When talking about fixed wing UAS, the aircraft will set to the optimum position to then begin the take-off. With rotary wing UAS, in order to move the aircraft it must previously lift from the ground. Then it will perform the taxi maneuver performing the path in order to do a correct take-off.

Note that the changes in this category are not just focused on the parameters set for the state as the Directed state is, but also in the implementation, as it may be seen with the Taxi State.

All these changes must be taken into account in order to perform a safe mission when changing from fixed wing to rotary wing UAS.

4 Conclusions

Since the flight performed by a fix wing UAS and a rotary wing UAS is different, the autopilot will differ its peculiarities. Therefore there must be a new VAS design for this type of UAS.

The fact of abstracting the autopilot implementation and generalizing it, without regarding the UAS vehicle that is working for, leads to lose the advantages of these aircrafts. Making reference to the flight VAS requires some modifications already explained. However the way of treating the data may also change.

Since an airplane has a limited bank angle, and therefore limited turning direction, the autopilot may calculate an anticipation distance to a current waypoint in order to arrive to the next waypoint, performing a fly-by strategy. When a rotary wing UAS is flying waypoints, it can approach to the current waypoint as much as desired without caring of the following, since once it has arrived, it may hover at that point flying over it waiting for new information. In addition, the turning direction rate of a helicopter is faster than the airplane; therefore the anticipation distance to the current waypoint is not that necessary.

The waypoint structure defined at USAL has the speed, altitude, waypoint identifier and location in space fields. A rotary wing UAS autopilot may also include a yaw angle that defines the entering direction of the helicopter to that waypoint. The orientation of the helicopter at the current waypoint may be interesting for going to the next one, or for making a concrete action at that location in space.

The changes that have to be made to VAS for working with helicopters are either architectural or conceptual modifications. For these reason the best solution will be to create two VAS: one oriented to airplanes, and the other oriented to helicopters.

References

[1] UAV Navigation AP04 Product Page, *http://www.uavnavigation.com/ap04.html*. UAVNavigation 2008.

[2] Cloud Cap Technology Page, *http://www.cloudcaptech.com/*. CloudCapTechnology 2007.

[3] MP2028 Series Autopilots Page, *http://www.micropilot.com*. MicroPilot 2007.

[4] P. Royo, J. López, E. Pastor, C. Barrado, *Service Abstraction Layer for UAV Flexible Application Development*. 46th AIAA Aerospace Sciences Meeting and Exhibit, AIAA, Reno, Nevada, 2008.

[5] Innovative Operational UAS Integration, *D1.2 Concept for Civil UAS Applications*. INOUI 2008.

[6] E. Pastor, J. Lopez, and P. Royo, *UAV Payload and Mission Control Hardware/Software Architecture*. IEEE Aerospace and Electronic Systems Magazine, Vol. 22, No. 6, 2007.

[7] J. López, P. Royo, E. Pastor, C. Barrado and E. Santamaria, *A Middleware architecture for unmanned aircraft avionics*. Proceedings of the 8th ACM/IFIP/USENIX international conference on Middleware, Newport Beach, California, 2007.

[8] European Civil Unmanned Air Vehicle Roadmap, *http://www.uavnet.com*. Volume 1- Overview, 2005.

[9] E. Santamaria, P. Royo, C. Barrado, E. Pastor, J. López, and X. Prats, *Mission Aware Flight Planning for Unmanned Aerial Systems*. AIAA Guidance, Navigation and Control Conference 2008, AIAA, Hawaii, 2008.

Authors

In addition to the authors cited at the front page there are Juan Lopez, Eduard Santamaria, Luis Delgado, and Cristina Barrado from the Technical University of

Catalonia, Barcelona, Spain. Also Anders la Cour-Harbo, Morten Bisgaard form the Aalborg Universitet. For further information please contact Juan Manuel Lema on jlema@ac.upc.edu.

Acknowledgments

This work has been partially funded by Ministry of Science and Education of Spain under contract CICYT TIN 2007-63927. This work has been co-financed by the European Organisation for the Safety of Air Navigation (EUROCONTROL) under its CARE INO III programme. The content of the work does not necessarily reflect the official position of EUROCONTROL on the matter. This investigation has been supported by UAV Navigation by using the AP04 autopilot and the Sniper II aircraft.

Copyright Statement