Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-30 (2016) 036–041

Control Platform for Multiple Unmanned Aerial Vehicles

Iulisloi Zacarias, Carlos E. T. Leite, Janana Schwarzrock, Edison P. de Freitas

Institudo de Informtica, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS 91501-970 Brasil (e-mail: cetleite@inf.ufrgs.br, epfreitas@inf.ufrgs.br, izacarias@ inf.ufrgs.br, jschwarzrock@ inf.ufrgs.br).

Abstract: With the prospect of regulating the use of UAVs, we expect a significant market growth in this sector. The control of a UAV individually is a problem already well treated and described in the literature; however, the control of multiples UAVs still presents challenges. This paper aims to present a platform for controlling multiple UAVs created from an existing application for only one vehicle named DroidPlanner. To carry out the proposed extension and to meet the requirements that this new application demands, changes are presented both in the communication infrastructure and in the user interface. For the experiments, we used a UAV simulator.

Keywords: UAVs applications, remote control, mobile, remote sensor.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are increasingly becoming popular, not only for military use for which they were originally conceived but also for civilian purposes. Thus, digital systems capable of control and visualize these devices in real time are crucial to the success of this market segment.

There are several well known solutions of platforms that allow to control only one vehicle at a time Oborne (2015); Geeksville Industries (2014). On the other hand, when working simultaneously with these vehicles type, new approaches are needed to solve a whole new range of problems arising from the use of multiple UAVs as the communication of such vehicles with the control station and the support of the control station to visualize multiple vehicles.

This work takes into account the current situation and needs of the UAV market, this paper aims to present a controller for multiple vehicles based on the development of an already established platform capable of control just one UAV called DroidPlanner Benemann (2013). To acquire this, the communication platform has been extended to accommodate multiple connections between the control station and UAVs controlled by it. This paper also discusses the creation of new messages to the control protocol.

This paper is organized as follow: Section 2 presents related work and a revision of relevant concepts to work. Section 3 discusses the architecture for multiple UAV controller. Section 4 discusses the communication infrastructure details that enables the control of multiple UAVs. Experiments and results are presented in Section 5, while Section 6 concludes the paper indicating possible future directions.

2. RELATED WORK AND CONCEPTS REVIEW

2.1 Related Work

The work presented by Luo et al. (2012) describes a proposal to control multiples UAV through a ground station. In this proposal, the UAV communication network is organized in a mesh and the ground station communicate with the nearby UAV. This UAV forwards the messages to others according to network topology. The differential proposed in Luo et al. (2012), regarding presented in this paper is that it does not allow the ground station to communicate with multiple UAVs, the communication is always made between the ground station and one UAV, and then, UAV to UAV until the messages are delivered to its destination. In the present work, the ground station can directly control each of the UAVs, and it is possible to configure the UAVs networks to perform communication via the relay structure.

A ground station simulator for UAVs, used to control UAVs, using the *Hardware In the Loop (HITL)* the technique is presented in Ajami et al. (2013). Through this ground station it is possible to configure a wide range of parameters for UAV flight missions as well receive flight telemetry data. However the communication occurs in a point-to-point form, among the ground station and each of UAV and to control multiple UAVs it is necessary to open multiple instances of the control software. Compared to the work presented here, this platform does not scale increasing the number of UAVs in the system. The same comparison is valid for the solution proposed by Koh et al. (2010). In this proposal, despite the possibility of load multiple tasks for multiple UAVs, the communication with each UAV also occurs as point-to-point.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.11.119

2.2 DroidPlanner

DroidPlanner is a control station for UAVs, also called *Ground Control Station (GCS)* which allows to control a single UAV. The first version of this software was launched in 2013 and it is currently in the third version, called Tower and provided by 3D Robotics (2015) company.

The application was released under the *General Public License (GNU)* that allows the user to modify and share the software. For this reason, other developers have been engaged to project over time, making the collaborative development.

The basic operation of the application is based on the exchange of messages among mobile devices and UAVs using the MAVLink protocol. Messages sent from DroidPlanner to the UAV are actions used to control the vehicle. The messages sent from UAV are a set of information that describe their current state. In addition to the commands, the control station is able to create tasks, defining points of interest on the map to be visited by the UAV.

2.3 The MAVLink Protocol

The communication protocol used in this work was the MAVLink (Micro protocol Air Vehicle Link). This protocol was developed in 2009 by Meier (2009), using the LGPL license.

The MAVLink protocol is a communication protocol widely used by small UAVs. Being lightweight it is ideal for exchange small information between the control station platform and the UAV, avoiding high processing costs to handle these messages. Its message structure is displayed in Fig. 1. It consists of a set of mandatory fixed size fields (1 Byte each) and an optional payload filed containing the message data to be transmitted.

Each MAVLink message is identified by a value in the identifier field (MSG field in Fig. 1) in its packet header and has its built-in content in the payload data field. In addition, these messages are logically defined by an XML document, which can represent the types of data being transmitted and the order in which the application should interpret the received bytes.

MAVLink protocol also allows to specify new messages for communication between applications. To set a new message it is necessary to represent it in a XML file, facilitating the automatic conversion to code with he aid of an automatic generation tool.

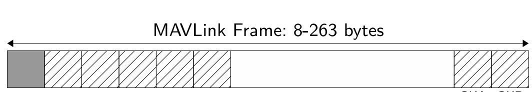

Fig. 1. Example of a MAVLink packet. Each field has 1 byte length, except the payload filed that can assume a variable length. The payload field is optional

3. MULTIPLE UAV CONTROLLER ARCHITECTURE

The architecture employed in the project was based on the *Model-View-Controller (MVC)* pattern, the same pattern employed in the original application, since it is a good choice of organization allowing different modules to be implemented independently facilitating integration of elements. However, some changes in the original MVC architecture had to be made to enable the introduction of new features.

In the Fig. 2 there is a representation, by means of a diagram, of the adopted architecture and the communication among the three components of the MVC model used in this paper. The control tier is responsible both for receiving data as sending data from the platform to UAV. The view corresponds to multiple instances of the original interface in the current solution. Finally the model represents the virtualization of multiple UAVs controlled by the system.

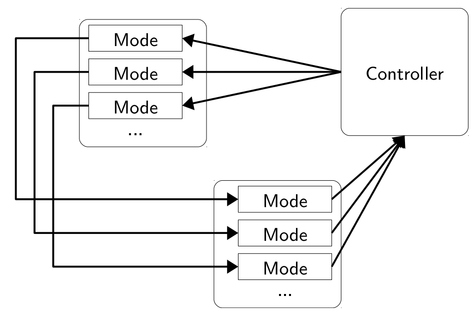

Fig. 2. Representation of modified MVC architecture used in this project

3.1 Model

The model component is responsible by virtualizing physical UAVs in the system, saving and changing their status based on MAVLink messages received. In addition to storing information about UAVs, this component has the important function to notify the interface about the changes in their status.

In this work, the model had to be modified to be able to represent multiple vehicles instances simultaneously. For this reason, this component of the MVC architecture was replicated in several times. These components has similar structure, but each of them contains different information, about different virtualized vehicles.

The Fig. 3 shows the internal view of each template block. All messages received by the control station and destined for the corresponding UAV are processed by this model changing the necessary components to finally notify the interface about changes.

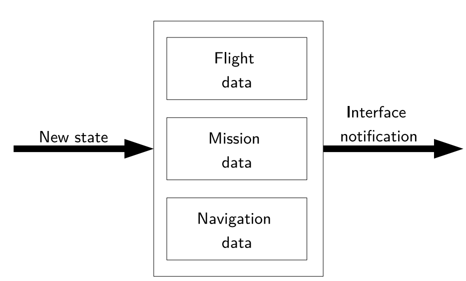

Fig. 3. Inside view of model representing of an UAV