Radio Communication to Control and Run an Autonomous Mission for UAVs via a Mobile App

Thiago R. F. Cavalcante
Master Degree in
Electrical Engineering
Federal University of Amazonas
Amazonas, Manaus
Email: thiagorodrigoengcomp@gmail.com

Erickson H. da S. Alves
Master Degree in
Electrical Engineering
Federal University of Amazonas
Amazonas, Manaus
Email: erickson.higor@gmail.com

Celso B. Carvalho
Department of Electronics
and Computer
Federal University of Amazonas
Amazonas, Manaus
Email: celsocarvalho75@gmail.com

***Abstract*—This paper presents the development of an Android application for the industry in order to start the production and supervise the process in production lines. A unmanned aerial vehicle (UAV) is used, equipped with a robotic gripper, which consists in an industrial scenario with an input warehouse, production lines, and a product depot. In the application, there is a mission planner that produces the needed commands for carrying out a given mission (the production of a product), which includes the delivery of inputs brought from the warehouse to the production line until the final product is delivered to the client (product depot). It is presented how the Android application is useful to assist in the supervisioning process for line managers/leaders.**

I. Introduction

Logistics has become a competitive and fundamental factor for organizations, involving the management, conservation, and supervision of freight transport. In addition, excellent logistics means customer satisfaction; so speed is still an important factor in a successful logistics process [1].

Currently, one of the solutions to this type of problem is the use of unmanned aerial vehicles (UAVs). Nowadays, UAVs are mostly remotely piloted vehicles (RPV), since their operations are carried out by ground operators. If the tasks performed by a UAV are performed autonomously, it would relieve the work of these operators, since they perform tedious and repetitive tasks [2].

One possible improvement of these logistics systems is the increase of the UAVs automation, which results in costs minimization. Consequently, investments and studies related to stand-alone UAVs are important to the smart factories development [3]. However, one of the main problems for using autonomous UAVs is the system's reliability and intelligence. Thus, increased employment of autonomous UAVs requires the development of devices, which are able to perform tasks and interact with the environment in an intelligent and reliable way.

Autonomous UAVs need to know what will happen in a future instant and what is the best decision to make at the present time; therefore, they require strategies not only to decompose their missions into meaningful subtasks, but also to track progress toward mission goals and the evolution of these tasks relative to the autonomous UAVs capabilities [4]. As a consequence, in order to successfully perform a mission, it is recommended to perform task planning [4]. Mission planning problems consist of planning events to meet certain requirements and objectives [5]. Therefore, planning events is one of the main challenges faced in solving this problem.

Both academy and industry have done researches about evaluation and optimization of mission planning in the last years. In [6] is used ant colony to optimize UAV missions. Another paper investigates energy consumption for a factory and evaluates the logistic planning processes using statistical metrics of evaluation [7].

In this work we present an Android application to supervise production lines in real-time, enabling leaders/managers to check statuses and act on failures. It uses the Drone Kit API, which enables sending commands and receiving updates to/from the drone.

In summary, the main contributions of this study are:

- an Android app to supervise production lines;
- methods to control the drone using a smartphone;
- a mission planner directly implemented in the app.

Outline. Section II provides the fundamentals of mission planning and optimization problems. Section II-D describes the UAV movement system used in this work. Section III explains the proposed evaluation methodology in further details. Section IV describes the experimental procedures and results in order to explore and demonstrate the potential of methodology, and, finally, Section V concludes this study and describes future work.

II. Preliminaries

A. Terminology

Key definitions related to the case study and the application developed in this study need to be clarified. All definitions below are adopted in the remainder of this study.

Definition 1: **(Mission Command)** Mission Command is a command created to execute a task such as to go from one location to another, get a package using a robot gripper, and land a UAV.

Definition 2: **(Mission)** Mission is the set of steps and mission commands that the UAV executes to produce the customer's order.

Definition 3: **(Warehouse)** Warehouse is the set of stored raw material available until the moment of entering the productive process. The raw materials, *i.e.*, the inputs available in this work are inputs A, B, and C.

Definition 4: **(Order)** Order is the requisition of products made by the client. In this study, the products are of type X and Y.

Definition 5: **(Production Time)** Production time is the time required to produce a product X or Y, after making available all the needed inputs for the production, given by the production rule.

Definition 6: **(Production Rule)** Production rule describes what and how many inputs are needed to produce a particular product.

Definition 7: **(Mission Planner)** Mission planner is the agent who performs the planning of a mission, that is, produce all steps and commands needed to carry out a given mission.

Definition 8: **(Mission File)** The mission file is a file that is created for the context of this work, with the extension .MISSION containing the mission itself.

Definition 9: **(Movement Function)** The movement function are functions created in Python using the drokekit API to send commands to the UAV by MAVLink protocol.

Definition 10: **(Production Mission)** The production mission is the set of steps to produce all the product required by the client.

B. UAV Iris+

The Iris+, introduced by 3DRobotics in Fall 2014, is a 550mm class (measured motor-to-motor) ready-to-fly quadcopter [8]. It is powered by a 5100mAh battery driving 4 950kV motors through a 4-in-1 electronic speed controller. It includes a Pixhawk flight control board (described below), an internal uBlox GPS with integrated 3-axis compass for navigation, and a 915MHz radio (or 433MHz, depending on country) for communication with a ground control station. Also included is a full-featured remote control transmitter with a preconfigured receiver onboard the drone. The Iris+ has an advertised flight time of 16-22 min. As of January 2016, the Iris+ retails for $600, very reasonable for a vehicle with its capabilities.

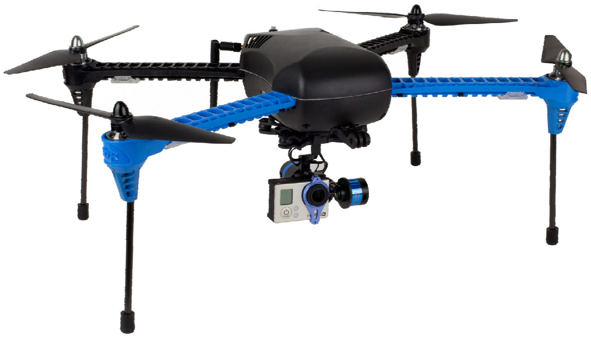


Fig. 1: The 3DR Iris+.

Control of the Iris+ is facilitated via the Pixhawk flight control board [9]. Developed by researchers at ETH Zurich, the Pixhawk is open source and very widely adopted. It is suitable for use in copters, planes and ground-based rovers. Connectivity options are numerous, allowing for use of a wide array of external devices such as GPS, range finders and companion computers. Internally, the unit integrates a three-axis accelerometer, a three-axis gyroscope, a three-axis compass and a barometer. These sensors allow for determination of motion, orientation, heading and relative altitude, respectively.

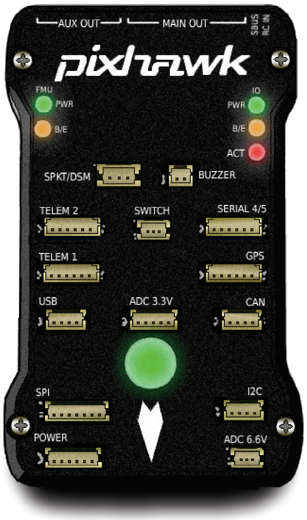


Fig. 2: The Pixhawk flight control board. Numerous ports allow for connection of a multitude of external sensors and devices.

C. Software

In this section we discuss the third-party open-source software that is integral to the vehicle. It is distinct from that we have developed as part of subsequent research projects. Firmware for the Pixhawk comes in the form of two distinct, but cooperating, open source efforts: APM

[10] and PX4 [11]. Both flight stacks have robust and supportive development communities and are updated frequently. MAVLink [12], a message-based communication protocol, underlies both efforts. In our vehicles, we use the APM flight stack.

MAVLink provides messages that allow for accessing and changing vehicle parameter values, checking vehicle status and navigation, changing, for example, vehicle position, attitude and velocity. The protocol is extensible, allowing users to define new messages for their purposes. However, the provided message set is sufficiently robust that we have not had the need to do this.

In 2015, 3DR introduced DroneKit-Android, a Android API for developing MAV applications [13]. DroneKit frees researchers and developers from many of the low-level aspects of MAVLink, providing a high-level interface for connecting to, monitoring and controlling a vehicle.

Vehicle configuration and sensor calibration are easily facilitated by ground control station software (GCS). Mission Planner (Windows only) and APM Planner (multi-platform) are both freely available. We use Mission Planner due to its ease of installation.

D. UAV Movement System

The core hardware of the UAV IRIS+ is the Pixhawk and we can control it using a Python library [13], which uses Micro Air Vehicle Link (MAVLink) protocol [9]. MAVLink is a protocol for communicating with small unmanned vehicle, which is designed as a header-only message marshalling library.

The UAV IRIS+ is integrated into a robot gripper to take and leave packages during missions (cf. Definition 2). We have connected a servo motor to the Pixhawk by one of the pulse width modulation (PWM) outputs. Figure 3 shows the system hardware architecture and the interconnections between each component module. In the hardware architecture shown in the Figure 3, we can see the UAV hardware component connections where there is the Pixhawk (flight controller) and its connections between other components such as the compass, GPS, PWM outputs, battery and etc. Moreover, it shows the connection with a robot gripper using a PWM outputs as a signal control for the servo motor in the robot gripper. Finally, it shows the communication between a personal computer (PC) and the UAV via radio control (RC) signal.

Command	Description
`TakeOff`	takes off the UAV
`GoTo`	moves the UAV to a certain location
`TakePackage`	takes an input/product (gripper)
`LeavePackage`	leaves an input/product (gripper)
`Wait`	makes the UAV to hover (wait)
`Land`	lands the UAV

TABLE I: Description of movement functions

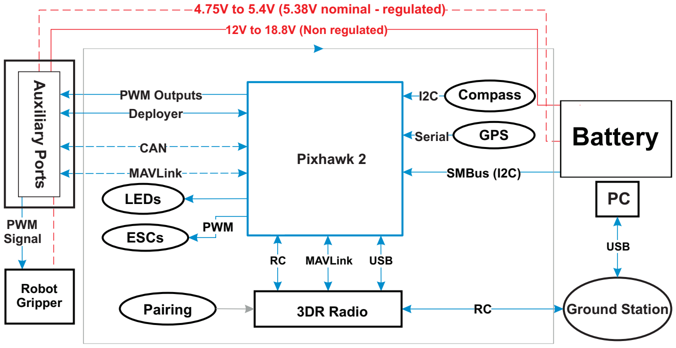


Fig. 3: System Hardware Architecture.

In the software architecture, the mission planner (cf. Definition 7) reads the warehouse inputs and client order and generates the production mission, which contains the list of mission commands needed for producing the required client order. Figure 4 shows the mission planning framework software components.
In order to control the UAV from a Android application, we have used the dronekit API that translates MAVLink commands to a Java function. The application talks to the UAV using a radio module connected via USB. We have created a bunch of functions in the control program for the most common UAV actions. The movement functions (cf. Definition 9) are described in Table I.

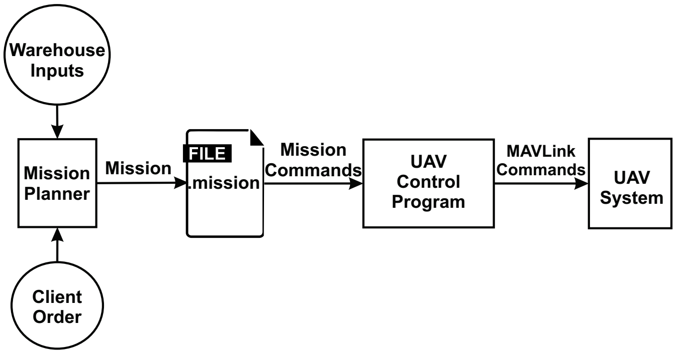


Fig. 4: System's Architecture.

III. Methodology

A. Case Study: UAV Intralogistics Mission

In order to model the mission planning problem as an optimization problem, the case study shown in Figure 5 is used.
Figure 5 shows that there are three types of inputs in the warehouse (*i.e.*, A, B, and C) and two production lines

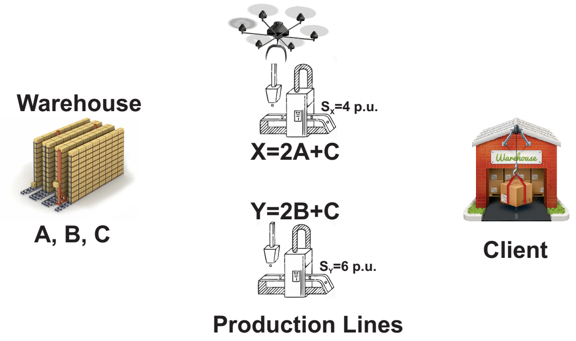


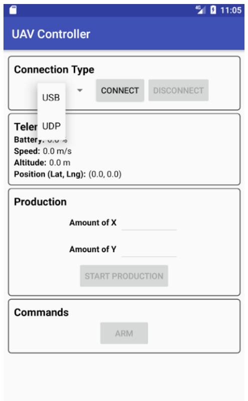
Fig. 5: Case Study Representation.

that produce two different products (*i.e.*, X and Y). Each production line produces only one type of product and has a characteristic production time (cf. Definition 5). Figure 5 shows that to produce a product of type X, two inputs of type A and one input of type C are required, and to produce a product of type Y, two inputs of type B and one input of type C are required. The production time of a X product is $4p.u.$ and the time of production of product Y is $6p.u.$. A production unit ($1p.u.$) is considered to be a `GoTo` command performed by the UAV.

The task to be performed is the production of the client order (cf. Definition 4), where a given UAV collects supplies from the warehouse, takes that to the production line, and once the production of a certain product is finished, the UAV delivers it to the client.

B. UAV Controller App

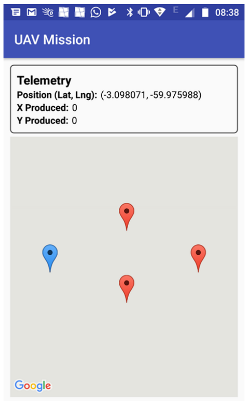
We developed an Android application to connect to the UAV via usb (radio signal module) or udp (simulated uav) in order to create a client order and make the UAV executes it autonomously and the user can visualize and supervise the operation.



Teler
Battery: 0.0 %
Speed: 0.0 m/s
Altitude: 0.0 m
Position (Lat, Lng): (0.0, 0.0)

Amount of X
Amount of Y

Fig. 6: Main Screen.



Telemetry
Position (Lat, Lng): (-3.098071, -59.975988)
X Produced: 0
Y Produced: 0

Fig. 7: Secondary Screen.

Internally in the app, it's implemented the mission planner and the controller module to send commands to the UAV via radio using the MAVLink protocol in the dronekit API.

IV. Experimental Evaluation

In this section we can see the experimental results of the Android application to control a UAV as described in Section III-A.

The Table II shows the experiments tested in a real UAV and a virtual UAV, in order to compare the results in both environments. As we can see, when the virtual UAV is used, the mission gets a smaller flight time than in the real UAV. That is because the radio communication used in the real drone is affected by noises in the environment.

	Real UAV	**Virtual UAV**
Tests	*Flight Time (s)*	*Flight Time (s)*
1	441,72	430,830
2	440,18	436,885
3	447,51	441,681
4	438,19	441,277
5	445,85	451,865

TABLE II: Mission Flight Time Tested in Real and Virtual UAV.

V. Conclusion

We have presented an evaluation methodology for UAV mission planner in an industrial production scenario, as described in Section III. We have developed an Android application to supervise production status and allow real-time monitoring for line managers/leaders.

In addition, we have developed a framework for mission planning, command and control for intralogistics mission using a commercial UAV. We used an UAV to solve intralogistics problems using the dronekit API to control and command adopting a high-level programming language.

Future work includes the use of computational vision for the recognition of inputs, and improvements of the optimization problem modeling for better results in cost evaluation. Supplementary, we will make experiments in a cooperative work environment where the number of UAVs is greater than two. In order to improve our results, we will develop more planner strategies such as an algorithm that produces different types of products simultaneously. Moreover, we will evaluate the communication between the drone and the application using different frameworks/protocols, in order to improve the supervision step.

References

[1] Service-drone, "Drones for logistic and transport | multirotor," 2014, accessed in: 12/05/2017. [Online]. Available: www.service-drone.com/en/production/logistics-and-transport

[2] D. Pascarella, S. Venticinque, and R. Aversa, "Autonomic agents for real time UAV mission planning," in *IEEE 10th International Conference on Autonomic and Trusted Computing*. IEEE, 2013, pp. 410–415.

[3] A. Hern, "DHL launches first commercial drone 'parcelcopter'delivery service," *The Guardian*, 2014.

[4] A. Finn and S. Scheding, *Developments and challenges for autonomous unmanned vehicles*. Springer, 2012.

[5] J. A. Krozel, "Search problems in mission planning and navigation of autonomous aircraft," Tech. Rep., 1988.

[6] C. Schwarz and J. Sauer, "Towards decentralised agv control with negotiations." in *STAIRS*, 2012, pp. 270–281.

[7] E. Müller, H. Hopf, and M. Krones, "Analyzing energy consumption for factory and logistics planning processes," in *IFIP International Conference on Advances in Production Management Systems*. Springer, 2012, pp. 49–56.

[8] D. Robotics, "Iris+," 2014, accessed in: 13/06/2017. [Online]. Available: 3drobotics.com/iris-plus

[9] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A system for autonomous flight using onboard computer vision," in *IEEE international conference on Robotics and automation*, 2011, pp. 2992–2997.

[10] A. Developers, "Ardupilot development site," 2015, accessed in: 13/06/2017. [Online]. Available: dev.ardupilot.com

[11] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," 2015.

[12] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Mavlink common message set," 2015, accessed in: 13/06/2017. [Online]. Available: pixhawk.ethz.ch/mavlink

[13] I. 3D Robotics. Dronekit-android. Accessed in: 20/05/2017. [Online]. Available: android.dronekit.io