PLANNING AND EVALUATION OF UAV MISSION FOR INTRALOGISTICS PROBLEM

Thiago Cavalcante*, Iury V. de Bessa†, Lucas C. Cordeiro‡

*Graduate Program in Electrical Engineering, Federal University of Amazonas, Manaus, AM, Brazil

†Department of Electricity, Federal University of Amazonas, Manaus, AM, Brazil

‡Department of Computer Science, University of Oxford, Oxford, United Kingdom

Emails: thiagorodrigoengcomp@gmail.com, iurybessa@ufam.edu.br, lucas.cordeiro@cs.ox.ac.uk

Resumo— Este artigo apresenta o desenvolvimento de planejadores de missão na intralogística para um veículo aéreo não tripulado comercial, equipado com uma garra robótica, em um ambiente industrial onde há almoxarifado de insumos, linhas de produção e deposito de produtos. Neste trabalho, o planejador gera comandos necessários para realizar uma missão a qual compreende desde a entrega de insumos trazidos do almoxarifado à linha de produção, até a entrega do produto final ao cliente. Foram desenvolvidas duas abordagens diferentes para planejamento de missão: na primeira abordagem, utilizou-se uma simples heurística que resolve o problema; já na segunda abordagem, utilizou-se uma técnica com escalonamento de tarefas (processo de produção). Estas abordagens seguem algumas regras de produção que serão apresentadas ao longo deste trabalho. Foi realizado uma avaliação dos planejadores de missão desenvolvidos, verificando o custo de ambos, realizando algumas medidas de tempo de execução, bem como comparando estes resultados com o custo ótimo obtido com a ferramenta de otimização CPLEX.

Palavras-chave— Planejamento de Missão, Sistemas de Manufatura, Problemas de Otimização.

Abstract— This paper presents the development of mission planners in intralogistics for a commercial unmanned aerial vehicle equipped with a robotic gripper in an industrial environment, which consists of an input warehouse, production lines, and a product warehouse. In this study, the planner produces the needed commands for carrying out a given mission, which includes the delivery of inputs brought from the warehouse to the production line until the final product is delivered to the customer (product warehouse). Two different approaches are developed for mission planning: in the first approach, a simple heuristic is used to solve the mission problem; in the second approach, a technique with task scheduling (production process) is employed; both approaches follow a set of production rules. An evaluation of the developed mission planners is performed, verifying the cost of both approaches, measuring the execution time, and comparing those results with the optimum cost obtained with the IBM ILOG CPLEX optimizer.

Keywords— Mission Planning, Manufacturing Systems, Optimization Problems.

1 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 (Service-drone, 2014). 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 (Pascarella et al., 2013).

A probable 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 (Hern, 2014). 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, and therefore require strategies not only to decompose their missions into meaningful sub-tasks, but also to track progress toward mission goals and the evolution of these tasks relative to the autonomous UAVs capabilities (Finn and Scheding, 2012). As a consequence, in order to successfully perform a mission, it is recommended to perform task planning (Garecht, 2010). Mission planning problems consist of planning events to meet certain requirements associated with the plan and also improving mission objectives (Krozel, 1988). Therefore, this is one of the main challenges faced in solving this type of problem.

Professor, nao consegui encontrar o erro de concordancia, soh mudei pra voz ativa... The academy has done some great researches about evaluation and optimization of mission planning in the last years. Schwarz and Sauer (2012) has

used ant colony to optimize missions for an automated guided vehicle (AVG). Another paper investigates energy consumption for a factory and evaluates the logistic planning processes using statistical metrics of evaluation (Müller et al., 2012). An evaluation metrics consists of a set of measures that follow a common underlying evaluation methodology. It is used to evaluate the efficacy of information retrieval systems and to justify theoretical and/or pragmatical developments of these systems (Pehcevski and Piwowarski, 2009). In this work, we use an optimal measure to compare with a calculated value of a mission cost.
This paper presents a methodology that evaluate the cost of mission planners for a commercial UAV. We developed a evaluation metrics that evaluates the relative cost of a planning strategy related with the optimal cost generated by the CPLEX optimizer.
In summary, the main contributions of this study are:

- a novel evaluation methodology for UAV intralogistics mission planners algorithms, which allows predicting the planners performance and also obtaining optimal algorithms and missions;
- development of an intralogistcs mission planner framework that provides mission commands for a UAV system;
- use of a commercial UAV system in intralogistics missions to demonstrate the evaluation methodology efficiency.

Outline: Section 2 describes previous studies related to mission planning, optimization, and evaluation. Section 3 provides the fundamentals of mission planning and optimization problems. Section 5 explains the proposed evaluation methodology in further details. Section 6 describes the experimental procedures and results in order to explore and demonstrate the potential of methodology, and, finally, Section 7 concludes this study and describes future work.

2 Related Work

In the literature, there are attempts to implement UAV guidance systems that perform mission planning. Doherty et al. (2009) presented a framework architecture for mission planning and execution tracking applied to an unmanned helicopter. During the mission execution, knowledge is acquired through sensors, which was used to create state structures. These structures allow constructing a logical model, representing the real system development and its environment over time. Then, the planning and monitoring modules use temporal action logic (TAL) to reason about actions and changes.
The NASA/U.S. Army autonomous helicopter project has developed a guidance system for the autonomous surveillance planning problem for multiple and different targets (Whalley et al., 2005), which generates mission plans using a theoretical approach for decision making. A high-level standalone control is provided by the framework Apex (Baer-Riedhart, 1998), a reactive procedure-based scheduler/planner used to perform mission-level tasks. Apex synthesizes a course of action primarily by linking elemental procedures expressed in procedural definition language (PDL), a notation developed especifically for the Apex reactive planner. This guidance system is integrated into a robotic helicopter and tested in more than 240 scenarios.
A similar project, called Ressac (Research and Rescue by Cooperative Autonomous System), is conducted by the French Aerospace Laboratory (ONERA) for a search and rescue scenario (Fabiani et al., 2007). This architecture for an exploration mission is developed based on the idea of decomposing the mission into a sequence of tasks or macro-actions associated with rewards. The problem is modeled using a Markov decision process framework (MDP) and dynamic programming algorithms for mission planning. Konigsbuch (Teichteil-Königsbuch and Fabiani, 2007) extends the guidance system and integrates with a robotic helicopter.
Finally, the German Aerospace Center (DLR) has also developed a mission management system based on the behavior paradigm (Adolf and Andert, 2010), which has been integrated with the ARTIS helicopter and validated in different scenarios, including follower of waypoints and search and tracking mission.
Existing approaches for evaluation of mission planners for intralogistics problems are either empirical or theoretical. The paper describes an approach combining both aspects. In addition, we test the metrics in a real environment, *i.e.*, we use a real UAV to performs a mission and we measure the cost (mission execution time) of the mission.

3 Preliminaries

3.1 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.

3.2 Mission Planning

Firstly, a mission can be defined as a goal that needs to be completed (cf. Definition 2). In the context of this study, the UAV mission is the packages delivery according to a set of well defined rules. A definition to mission planning for UAV is the process of planning the locations to visit (waypoints) and the actions that the vehicles can perform (e.g., loading/dropping a load and taking videos/pictures), typically over a time period (Ramirez-Atencia et al., 2014). An important term in this study is the concept of a planner, which is the agent (software implementation) that generates a mission. Functionally, mission planning lies above the trajectory planning process, where the mission planner (cf. Definition 7) generates a desired mission plan, and then the trajectory planner generates the flight plan (trajectories) between the waypoints.

3.3 Optimization Problems

An optimization problem is related to finding the best solution (relative to a certain criterion) among a set of available alternatives. For instance, the popular bin packaging problem that aims to find the number of boxes of a certain size to store a set of objects of indicated sizes; optimization involves, for example, finding the least amount of boxes. An optimization problem is usually represented as follows:

$$\begin{array}{ll} \min & f(\mathbf{x}), \\ \text{subject to} & \mathbf{x} \in \Omega. \end{array} \tag{1}$$

Where Ω is a set of constraints of the problem.
An optimization problem can be defined as a finite set of variables, where the correct values for the variables specify the optimal solution. If the variables are of the set of real, the problem is called continuous, and if they can only have a finite set of distinct values, the problem is called combinatorial (Francq, 2011).

4 UAV Movement System

In this section, we first investigate the UAV platform used (3DR IRIS+) in this study, describing the hardware characteristics and the control framework developed for intralogistics missions. The core hardware of the UAV IRIS+ is the Pixhawk and we can control it using a Python library (3D Robotics, n.d.), which uses Micro Air Vehicle Link (MAVLink) protocol (Meier et al., 2011). MAVLink is a protocol for communicating with small unmanned vehicle, which is designed as a header-only message marshalling library.
The IRIS+ UAV 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 1 shows the hardware architecture of system and the interconnections between each component module. Figure 2 shows the mission planning framework software components.

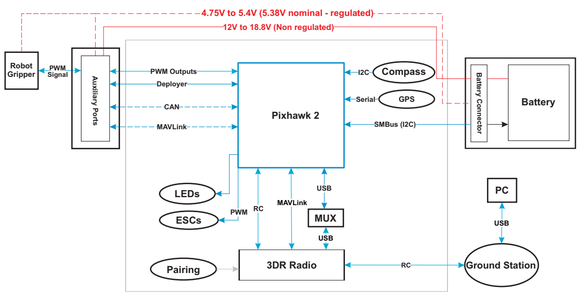

Figure 1: System Hardware Archtecture. Thiago, poderias aumentar a fonte do texto da figura?

The Mission Planner (cf. Definition 7) reads the warehouse inputs and client order and produces a `.mission` file, which contains the list of mission commands needed for producing the required client order. This `.mission` file is used by a UAV Control Program to control the UAV and to produce the low-level movement commands (cf. Definition 1).
In order to control the UAV from a PC, we have used the dronekit API that translates MAVLink commands to a Python function. In the ground station, the PC is running the UAV Control Program that controls the UAV using a radio module connected to the PC 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 below:

- `TakeOff`: takeoff command of the UAV;
- `GoTo`: command to move the UAV to a certain location;
- `TakePackage`: command to collect an input/product through a robot gripper;
- `LeavePackage`: command to leave an input or product from a robot gripper;
- `Wait`: command to make a UAV to hover (wait);
- `Land`: command to make a UAV to land.

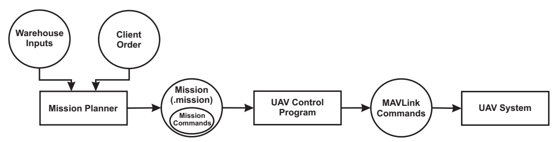

Figure 2: System's Architecture. Thiago, poderias aumentar a fonte do texto da figura?

5 Methodology of Time Cost Evaluation, UAV Use and Mission Planning

In this section, we evaluate the algorithm mission cost and describe a case study to a UAV and contents about mission planning.

5.1 Case Study

In order to model the mission planning problem as an optimization problem, the case study shown in Figure 3 is used.
Figure 3 shows that there are three types of inputs in the warehouse (i.e., A, B, and C) and two production lines that produces 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 3 shows that to produce a product of type X, two inputs of type A and one input of type C are required,

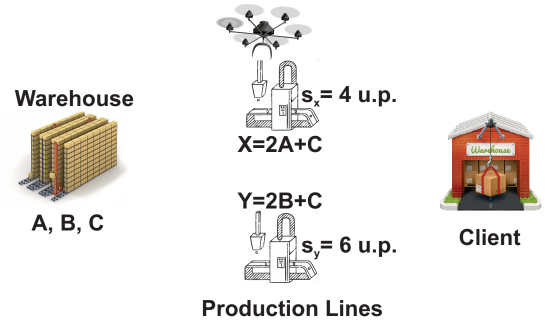

Figure 3: Case Study Representation.

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. Tempo de producao de produto eh medido em p.u.? Esta parte ficou confusa...isso mesmo professor...em seguida eu explico oq eh p.u, q eh exatamente o numero de comandos goto que tem na missao...
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.

5.2 Modelling a UAV Intralogistics Mission as an Optimization Problem

The purpose of this subsection is make the mission planning problem into an optimization problem, creating a modelling for the problem; in order to find, afterwards, the shortest execution time of all tasks (minimization), based on the case study explained in Section 5.1. The notation used is given below:

- $\mathcal{T} = \{T_j | j \in \mathbb{N}^*, j \leq N\}$ is the set of N tasks;
- $\mathcal{M} = \{m_i | i \in \mathbb{N}^*, i \leq M\}$ is the set of M production lines (machines);
- $\mathcal{P} = \{p_j | j \in \mathbb{N}^*, j \leq N\}$ is the processing time of each j-th task;
- $S = \{s_i | s \in \mathbb{N}^*, i \leq M\}$ is the setup (production) time of each i-th production line;

Decision variable The variable x_{ij} is a binary decision variable that takes the value 1 if the task j is running on the machine i, and 0 otherwise. The variable $C_{mission}$ is the variable that we want to optimize.

$$x_{ij} = \begin{cases} 1, & \text{if the task } j \text{ is running in} \\ & \text{the machine (production line) } i \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Objective function The objective function is the total mission cost $C_{mission}$ (total process execution time) that can be modelled as follows.

$$C_{mission} = \sum_{i=1}^{M} \sum_{j=1}^{N} (p_j + s_j) x_{ij}, \tag{3}$$

Eq. (3) represents the sum of the duration time p_j of each travel from one place another in the case study explained in Section 5.1, considering the production time (cf. Definition 5) s_j in each production line.

Constraints

- Each task must be executed/processed in a unique machine:

$$\sum_{i=1}^{M} \sum_{j=1}^{N} x_{ij} = 1 \tag{4}$$

- Execution time of each machine:

$$C_{mission} \leq C_{max} \tag{5}$$

Eq. (5) indicates that the mission cost is always less or equal than a maximum cost denoted by C_{max}, obtained empirically.

Resulting optimization problem The resulting optimization problem consists in minimizing C_{max} w.r.t. the decision variable (2) constrained to the condition in Eq. (4) and (5). Thus, the optimization problem is represented as follows:

$$\begin{array}{ll} \min & C_{mission}, \\ \text{s.t.} & \sum_{i=1}^{M} \sum_{j=1}^{N} x_{ij} = 1, \\ & C_{mission} \leq C_{max} \end{array} \tag{6}$$

5.3 Planner Evaluation Methodology

The main contribution of this study is a methodology to evaluate UAV mission planner algorithms and to find minimum cost planner. For this purpose, a generalized evaluation metrics is developed. The objective (cost) function modeled in Section 5.2 is related to the total time spent for the mission execution. Our evaluation metrics compares the cost of a planner algorithm with the best cost computed by the CPLEX solver (CPLEX, 2003). The metrics presented here is called Mission Planner Cost Index ($MPCI$).
Firstly, the optimal cost of the problem is obtained by means of the CPLEX solver, which returns the optimum value (minimum mission execution time). The model proposed in Section 5.2 is implemented using the CPLEX solver library available for C++.

The cost of each planner strategy (algorithm 1 and 2) c_X is obtained by counting the number of `GoTo` commands which represents a process (task). Finally, the evaluation of each mission planner is computed w.r.t. the optimal cost, therefore, the $MPCI_X$ of a planner X is computed as follows:

$$MPCI_X = \frac{c_o}{c_X}, \tag{7}$$

where c_o is the optimal cost obtained by the CPLEX solver, c_X is the cost of the solution generated by planner X, and $0 \leq MPCI_X \leq 1$. Note that as close to 1 the $MPCI_X$ is, the solution cost becomes smaller.

6 Experimental Evaluation

This section describes the experimental results obtained in this project, as well as the cost evaluation of two techniques used, which are compared to the optimum cost implemented with the CPLEX solver.

6.1 Experimental Environment and Objectives

In order to verify the efficiency of the metrics shown in the Section 5, our experimental evaluation aims to answer the following research questions: TODO

RQ1 **(performance)** which of the Planner strategies are more efficient?

RQ2 **(sanity check)** ?

TODO
In order to clarify the evaluation methodology efficiency, two mission planner algorithms are evaluated using the proposed metrics, and the evaluation result is compared with practical execution time of algorithms that are implemented in the simulation tool SITL[1] and in a real UAV system (3DR IRIS+). The mission planning algorithms were executed in a computer running Linux Mint, core i7 processor and 8 GB of RAM. In order to control the UAV, we run the control program, which uses the dronekit API as an interface between a high level program language (Python) and the protocol that the UAV understands (MAVLink), in the same computer where there is a radio module connected via USB communicating with the UAV radio module.

6.2 Mission Planners

In this study, we considered that mission planner is a software that generates a production mission given the warehouse and customer order. This program generates a `.mission` extension file containing a set of mission commands, as described in

[1]Simulator that allows executing a Plane, Copter or Rover without the need of a hardware)

Section 4. Two examples of planners are presented in this work and are employed to demonstrate the cost evaluation methodology.

6.2.1 Planner A

In Algorithm 1, we show one strategy to solve the mission planning problem and we denoted it as Planner A. In this particular algorithm, the production of X products has a higher priority over Y, *i.e.*, the inputs are firstly allocated to production of X orders, and the production of Y products begins if there is no other X to be produced. The general steps of planner A are described in Algorithm 1.

Algorithm 1 Planner A

```
Input: warehouse
Input: order
Output: mission file .mission
begin
    check the order;
    repeat
        go to the warehouse;
        repeat
            get input A;
            bring to the production line X;
        until until bring 2 A elements;
        go to the warehouse;
        get the input C;
        bring to the production line X;
        wait X to be produced;
        bring X to the client;
    until production of all X elements finish;
    repeat
        go to the warehouse;
        repeat
            get input B;
            bring to the production line Y;
        until until bring 2 B elements;
        go to the warehouse;
        get the input C;
        bring to the production line Y;
        wait Y to be produced;
        bring Y to the client;
    until production of all Y elements finish;
end
```

6.2.2 Planner B

The strategy for Planner B is a bit more complex than Planner A. In the Planner B strategy, the UAV starts to bring all the necessary inputs to make the first product X, taking all the A, B and C inputs, respectively, to the production line X. After bring all the necessary inputs to produce the first X product, the production line X starts to produce the X product and while the production line X is producing, the UAV goes to the warehouse to get the necessary inputs to produce the next product (either Planner A and Planner B produce fistly the X products e then all the Y products). However, when the X product finishes producing, the UAV knows the instant and goes to the production line to get the X product to bring to the client place; and after that, the UAV goes back to bring the rest of the inputs. The UAV keeps work in the same way until it brings all the products to the client place. Differently to the Planner A, the Planner B does not wait the production in the production line. The UAV works as a scheduler and executes the mission faster than Planner A strategy. The general steps of planner B are shown in the Algorithm 2.

Algorithm 2 Planner B

```
Input: warehouse
Input: order
Output: mission file .mission
begin
    initialize t_x;
    initialize t_y;
    check the order;
    repeat
        if the counter of that X is not t_x then
            go to the warehouse;
            repeat
                get the input A;
                bring to the production line X;
            until until bring 2 A elements;
            go to the warehouse;
            get the input C;
            bring to the production line X;
            start the counter of this X
            (production time);
            keep producing;
        else
            go back to the production line X;
            bring X to the client;
            go back to producing;
        end
    until production of all X elements finish;
    repeat
        if the counter of that X is not t_y then
            go to the warehouse;
            repeat
                get the input B;
                bring to the production line Y;
            until until bring 2 B elements;
            go to the warehouse;
            get the input C;
            bring to the production line Y;
            start the counter of this Y
            (production time);
            keep producing;
        else
            go back to the production line Y;
            bring X to the client;
            go back to producing;
        end
    until production of all X elements finish;
end
```

6.3 Cost Evaluation

The results of each mission planner is compared to the optimal solution obtained with the branch-and-cut algorithm of the IBM/ILOG CPLEX 12.4 tool developed in C++ (CPLEX, 2003). In order to obtain better results to perform the comparison, it was considered only the time in which the UAV takes to finish the production of a product.

	Time (s)
Planner A	420
Planner B	404
CPLEX	134

Table 1: Time execution for comparison with the optimal solution time

The Table 1 shows the mission execution times obtained using the planner algorithm A, the planner algorithm B, and the minimum value given by the solver. Using the metrics shown in the section 5, then:

$$Creal_A = \frac{134}{420} = 0.319 \tag{8}$$

$$Creal_B = \frac{134}{408} = 0.328 \tag{9}$$

It can be seen in the Equation 9 that the proposed evaluation methodology indicates that planner B performs the mission more quickly and has a lower cost than planner A.

6.4 Practical Results

To verify the practical results, as well as a cost comparison between the different approaches of mission planners developed in this work, the flight time measurement was performed using the two mission planning algorithms developed, using the case study shown in 5.1.

Table 2 presents the performance (total flight time) of both planners algorithms.

	Planner A	**Planner B**
Tests	***Flight Time (s)***	***Flight Time (s)***
1	460,405	430,830
2	460,693	436,885
3	462,080	441,681
4	457,719	441,277
5	461,227	451,865

Table 2: Mission Planners Flight Time - Simulator.

Table 2 indicates that the mission time of planner B is lower then the time of planner A in all five tests, ensuring the results of the evaluation methodology. Figure 4 shows a bar graph of the simulation flight times of both planners in the five tests.

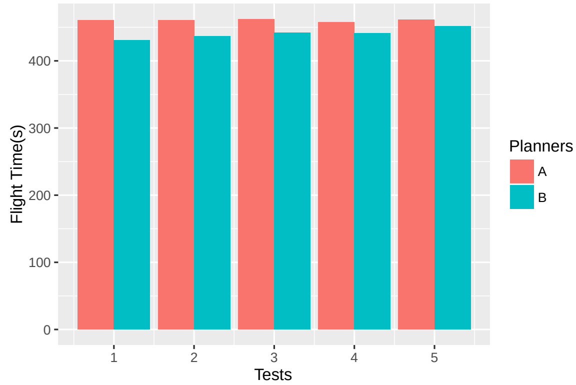

Figure 4: Flight Time Graph of Mission Planners Relative to 5 Tests in Simulator.

Additional experiments are performed with the real UAV system (3DR IRIS+). Figure 5 shows the map of experimental environment (Faculty of Physical Education and Physiotherapy of Federal University of Amazonas).

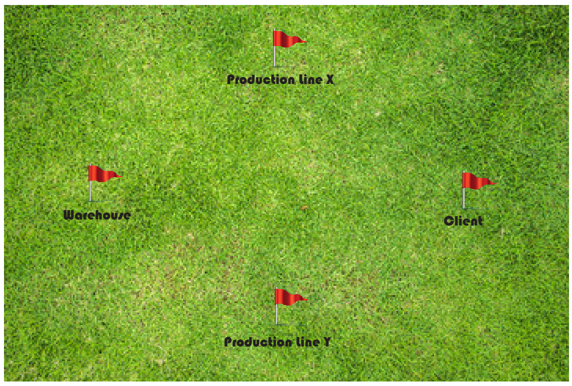

Figure 5: Warehouse, Production Line X, Production Line Y and Costumer in the Map.

Table 3 shows the the flight times for both planners. The results indicates that the planner B also shown a shorter time compared to planner A in real environment.

	Planner A	**Planner B**
Tests	***Flight Time (s)***	***Flight Time (s)***
1	455,12	441,72
2	456,93	440,18
3	457,19	447,51
4	460,25	438,19
5	459.47	445,85

Table 3: Mission Planners Flight Time - Real.

Figure 6 shows a bar graph of the flight times of both planners in the five tests with IRIS+.

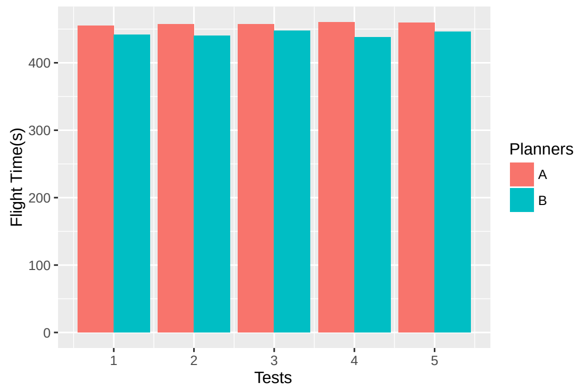

Figure 6: Flight Time Graph of Mission Planners Relative to 5 Tests in a Real Environment.

The aforementioned results confirmed the prediction provided by the planner evaluation methodology, and the planner B is faster than planner A in all the tests with simulations and IRIS+ tests. Thiago, nao ficou claro pra mim os objetivos dos experimentos... poderias escrever claramente quais seriam as research questions que devemos responder nos experimentos? Alem disso, precisas descrever uma secao sobre ameacas da validade dos resultados... como exemplo, olhe este artigo: https://arxiv.org/pdf/1610.04761.pdf

7 Conclusions

Thiago, precisas melhorar substancialmente esta secao... a conclusao deve ser mais forte. Precisas enfatizar as contribuicoes e os principais resultados do seu trabalho.

We have developed an evaluation methodology for UAV mission planner in an industrial production scenario, as described in Section 5. We used this methodology of evaluation to perform two different algorithms to verify the performance of them. In addition, we have used a commercial UAV to perform our methodology. We have created a framework with a set of mission commands to control the UAV using dronekit API as presented in the Section 4 and 5.

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.

References

3D Robotics, I. (n.d.). Dronekit.
URL: *dronekit.io*

Adolf, F. and Andert, F. (2010). *Onboard mission management for a VTOL UAV using sequence and supervisory control*, INTECH Open Access Publisher.

Baer-Riedhart, J. (1998). Nasa environmental research aircraft and sensor technology program, *Aerospace America, January* .

CPLEX, S. (2003). Ilog, *Inc., Armonk, NY* .

Doherty, P., Kvarnström, J. and Heintz, F. (2009). A temporal logic-based planning and execution monitoring framework for unmanned aircraft systems, *Autonomous Agents and Multi-Agent Systems* **19**(3): 332–377.

Fabiani, P., Fuertes, V., Piquereau, A., Mampey, R. and Teichteil-Königsbuch, F. (2007). Autonomous flight and navigation of vtol uavs: from autonomy demonstrations to out-of-sight flights, *Aerospace Science and Technology* **11**(2): 183–193.

Finn, A. and Scheding, S. (2012). *Developments and challenges for autonomous unmanned vehicles*, Springer.

Francq, P. (2011). Optimization problems. Accessed in: 20/09/2016.
URL: *http://www.otlet-institute.org/wikics/Optimization_Problems.html*

Garecht, J. (2010). How to write a successful fundraising plan. Accessed in: 17/10/2015.
URL: *http://www.thefundraisingauthority.com/fundraising-basics/fundraising-plan/*

Hern, A. (2014). Dhl launches first commercial drone 'parcelcopter'delivery service, *The Guardian* .

Krozel, J. A. (1988). Search problems in mission planning and navigation of autonomous aircraft, *Technical report*.

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

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

Pascarella, D., Venticinque, S. and Aversa, R. (2013). Autonomic agents for real time uav mission planning, *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, IEEE, pp. 410–415.

Pehcevski, J. and Piwowarski, B. (2009). Evaluation metrics for structured text retrieval, *Encyclopedia of Database Systems*, Springer, pp. 1015–1024.

Ramirez-Atencia, C., Bello-Orgaz, G., R-Moreno, M. D. and Camacho, D. (2014). Solving uav mission planning based on temporal constaint satisfaction problem using genetic algorithms, *Proceedings of Doctoral Program of the 20th International Conference on Principles and Practice of Constraint Programming*, pp. 65–71.

Schwarz, C. and Sauer, J. (2012). Towards decentralised agv control with negotiations., *STAIRS*, pp. 270–281.

Service-drone (2014). Drones for logistic and transport | multirotor. Accessed in: 12/02/2016.
URL: *http://www.service-drone.com/en/production/logistics-and-transport*

Teichteil-Königsbuch, F. and Fabiani, P. (2007). A multi-thread decisional architecture for real-time planning under uncertainty, *3rd ICAPS'07 Workshop on Planning and Plan Execution for Real-World Systems*.

Whalley, M., Freed, M., Harris, R., Takahashi, M., Schulein, G. and Howlett, J. (2005). Design, integration, and flight test results for an autonomous surveillance helicopter, *Proceedings of the AHS International Specialists' Meeting on Unmanned Rotorcraft*.