

Trabajo Fin de Máster  
Máster en Ingeniería Electrónica, Robótica y  
Automática

Aerial co-workers: a task planning approach  
for multi-drone teams supporting inspection  
operations

Autor: Álvaro Calvo Matos

Tutor: Jesús Capitán Fernandez

**Dpto. Ingeniería de Sistemas y Automática**  
**Escuela Técnica Superior de Ingeniería**  
**Universidad de Sevilla**

Sevilla, 2021





Trabajo Fin de Máster  
Máster en Ingeniería Electrónica, Robótica y Automática

# **Aerial co-workers: a task planning approach for multi-drone teams supporting inspection operations**

Autor:

Álvaro Calvo Matos

Tutor:

Jesús Capitán Fernandez

Associate Professor

Dpto. Ingeniería de Sistemas y Automática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2021



Trabajo Fin de Máster:   Aerial co-workers: a task planning approach for multi-drone teams  
supporting inspection operations

Autor:           Álvaro Calvo Matos  
Tutor:           Jesús Capitán Fernandez

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



# Acknowledgment

---

To my tutor, Jesús, for guiding me in this project, for trusting me to join the research group to which he belongs, for supporting me in my decision to join a doctoral program, for always seeking the best for us despite of his preferences and for his kindness.

To all those department mates who have helped me every time I have needed it and all those who have ever volunteered to help. In particular, I would like to thank Fran and Arturo for all the time they have dedicated to helping me.

To Damián, for accompanying me in the easy and difficult moments, but above all, for being my friend, and for being there unconditionally for whatever I needed.

To my classmates, who despite being a difficult year with social distancing, have been as close as ever.

To all my friends, for being such good friends.

To my entire family, for their unconditional love and support, and for their patience and understanding.

Thanks for everything

*Álvaro Calvo Matos*

*Sevilla, 2021*





# Abstract

---

This master's thesis has addressed problems arising from the recent increase in the applications of cooperative Unmanned Aerial Vehicle (UAV) teams, which are the autonomy to operate over a long period of time with robustness to possible failures, and the difficulty of providing the team with cognitive capabilities to be able to operate in dynamic environments with humans.

Many of these applications are currently being executed by humans, making the activities much more expensive, time-consuming, and in some cases even dangerous. This is why there is currently a great deal of interest and effort being put into developing solutions to the problems posed.

The aim of the work was to develop cognitive planning techniques for coordinating fleets of quadrotors to assist human operators in inspection and maintenance tasks on high-voltage power lines. These techniques should also extend the autonomy of the system, ensure that safety requirements between drones and human workers are met, and ensure the success of the mission.

A software architecture has been proposed based on a central planner and a distributed behaviour manager. To carry out the planning, a cost has been defined, which is calculated for each task. Thus, each one is assigned to the UAV that consumes the least executing it. On the other hand, to control the behaviour of the drones and ensure the safety of the aerial equipment, a behaviour tree has been implemented.

As a result, it has been possible to develop a software architecture capable of dynamically planning missions while ensuring the safety of the equipment involved. This provides a good base that can be easily adapted and from which more complex planners can be developed in the future. Compared to the typical way of implementing behaviour managers, involving complex finite state machines that are difficult to read, reuse and extend, the use of behaviour trees is a great improvement and will allow the creation of increasingly complex behaviours.



# Resumen

---

Este Trabajo de Fin de Máster ha afrontado problemas que surgen del reciente aumento de las aplicaciones de equipos cooperativos de UAV, los cuales son la autonomía para operar de forma prolongada en el tiempo con robustez ante posibles fallos, y la dificultad de aportar al equipo capacidades cognitivas para poder operar en entornos dinámicos con humanos.

Muchas de estas aplicaciones están siendo ejecutadas actualmente por humanos, haciendo las actividades mucho más costosas, lentas, e incluso en algunos casos, peligrosas. Es por eso que actualmente existe un gran interés y se están destinando muchos esfuerzos para desarrollar soluciones para los problemas planteados.

El objetivo del trabajo era desarrollar técnicas cognitivas de planificación para coordinar flotas de drones que asistan a operarios humanos en tareas de inspección y mantenimiento en líneas eléctricas de alta tensión. Estas técnicas debían además extender la autonomía del sistema, garantizar que se cumplen los requisitos de seguridad entre drones y trabajadores humanos, y asegurar el éxito de la misión.

Se ha propuesto una arquitectura de software basada en un planificador central y un gestor de comportamiento distribuido. Para llevar a cabo la planificación se ha definido un coste, que es calculado para cada tarea. De esta forma, cada una se asigna al UAV al que cueste menos. Por el otro lado, para controlar el comportamiento de los drones y asegurar la seguridad de los equipos aéreos, se ha implementado un árbol de comportamiento.

Como resultado, se ha conseguido desarrollar una arquitectura de software capaz realizar la planificación de las misiones de forma dinámica asegurando mientras tanto la seguridad de los equipos involucrados. Esto constituye una buena base que se puede adaptar fácilmente y a partir de la cual se pueden desarrollar futuros planificadores más complejos. Comparado con la forma típica de implementar gestores de comportamiento, involucrando complejas máquinas de estados finitas difíciles de leer, reutilizar y ampliar, el uso de árboles de comportamiento supone una gran mejora y permitirá la creación de comportamientos cada vez más complejos.



# Short Outline

---

<i>Abstract</i>	III
<i>Resumen</i>	V
<i>Short Outline</i>	VII
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation	1
1.2 Objectives	2
<b>2 Preliminaries</b>	<b>5</b>
2.1 Current technology	5
2.2 Related work	6
2.3 Tools	6
<b>3 Problem Formulation</b>	<b>7</b>
3.1 Description of tasks	7
3.2 Battery recharges	7
3.3 Connection losses	7
3.4 Task replanning situations	7
<b>4 Design of the proposed solution</b>	<b>9</b>
4.1 Node diagram	9
4.2 Centralized module: task planner	9
4.3 Distributed module: behavior manager	9
4.4 Lower and upper level modules faker	9
<b>5 Results</b>	<b>11</b>
5.1 Task planning	11
5.2 Drone behaviour manager results	11
<b>6 Conclusions and future work</b>	<b>13</b>
6.1 Conclusions	13
6.2 Future work	13
<i>List of Figures</i>	15
<i>List of Tables</i>	17

<i>List of Codes</i>	19
<i>Bibliography</i>	21
<i>Index</i>	25
<i>Glossary</i>	25

# Contents

---

<i>Abstract</i>	III
<i>Resumen</i>	V
<i>Short Outline</i>	VII
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation	1
1.2 Objectives	2
<b>2 Preliminaries</b>	<b>5</b>
2.1 Current technology	5
2.1.1 UAVs	6
2.1.2 Multi-drone teams	6
2.1.3 Aerial co-workers	6
2.2 Related work	6
2.2.1 Inspection applications with UAVs	6
2.2.2 Task planning in multi-drone teams	6
2.2.3 Drone behavior management	6
2.3 Tools	6
2.3.1 ROS	6
2.3.2 Gazebo	6
2.3.3 Rviz	6
2.3.4 UAL	6
2.3.5 Behaviour Trees	6
2.3.6 Groot	6
<b>3 Problem Formulation</b>	<b>7</b>
3.1 Description of tasks	7
3.1.1 Inspection tasks	7
3.1.2 Monitoring tasks	7
3.1.3 Tool delivery tasks	7
3.2 Battery recharges	7
3.3 Connection losses	7
3.4 Task replanning situations	7
<b>4 Design of the proposed solution</b>	<b>9</b>
4.1 Node diagram	9

4.2	Centralized module: task planner	9
4.3	Distributed module: behavior manager	9
4.3.1	Main tree	9
4.3.2	Inspection task tree	9
4.3.3	Monitoring task tree	9
4.3.4	Tool delivery task tree	9
4.4	Lower and upper level modules faker	9
<b>5</b>	<b>Results</b>	<b>11</b>
5.1	Task planning	11
5.1.1	Battery	11
5.1.2	Connection lost	11
5.1.3	Replanning	11
5.2	Drone behaviour manager results	11
5.2.1	Battery management	11
5.2.2	Connection lost management	11
5.2.3	Replanning management	11
<b>6</b>	<b>Conclusions and future work</b>	<b>13</b>
6.1	Conclusions	13
6.2	Future work	13
6.2.1	Augmented reality	13
	<i>List of Figures</i>	15
	<i>List of Tables</i>	17
	<i>List of Codes</i>	19
	<i>Bibliography</i>	21
	<i>Index</i>	25
	<i>Glossary</i>	25



# 1 Introduction

---

The use of UAVs has grown considerably in recent years for numerous applications including real-time monitoring, search and rescue, providing wireless coverage, security and surveillance, precision agriculture, package delivery and infrastructure inspection [1]. With the rapidly developing technology in this area, and demonstrations of what UAVs can do, there are increasing efforts to bring this technology to other applications. With the expected increase in applications for this technology, new problems and challenges arise, including autonomy, safety, obstacle avoidance and coordination of multi-UAV teams. Developing the technology to solve these problems will be a major effort, but as UAVs have proven to be critical in situations where humans are at high risk or highly inefficient, and they have proven their capacity to evolve and develop even more potential in the short term, companies are investing in developing all sort of UAV-based solutions.

## 1.1 Motivation

With the increase in global electricity demand, a challenge has arisen for electricity supply companies to maintain and repair power grids in a way that minimizes the frequency of outages. According to [2], one of the main causes of power outages is damage to transmission lines due to bad weather or inefficient inspection campaigns.



**Figure 1.1** Operators getting off the helicopter during a maintenance mission.

The strategy often used by electric companies to reduce power outages is to schedule periodic maintenance operations on active lines. This is the most suitable method if the correct functioning of the system is to be ensured and when replacing a circuit is unacceptable [2]. These maintenance missions are carried out by experienced crews on board helicopters and equipped with safety suits and harnesses among other things that prevent the operators from receiving an electric shock (see figure 1.1). The problem with this solution is that these activities are dangerous for the operators, as they are working at high altitude and on electrified lines, are extremely time-consuming and expensive (\$1500 per hour) and are subject to human error [3].

These are the reasons why distribution companies have the need to develop more efficient and safer maintenance methods. Multiple solutions have been proposed to automate this task [4], but the best of them seems to be the use of UAVs because of their flexibility and ability to inspect at different levels [2]. To achieve this, there are still some important barriers to overcome, such as the limited autonomy of these devices, the strong electromagnetic interference to which they would be subjected due to being close to power lines and the ability to detect and avoid obstacles of different nature that can be found in this type of environment [3]. Providing UAVs with the cognitive capability to operate autonomously in such dynamic environments and with the presence of humans, and providing them with a rapid on-line planning method [5], is key to address these complexities and to safely and successfully accomplish the assigned mission with UAV fleets.

A versatile and reliable software architecture will be essential to integrate and interconnect all the heterogeneous components that compose these cognitive multi-UAV systems. In [6], as part of the AERIAL-CORE European project<sup>1</sup>, a multi-layer software architecture is presented for carrying out such missions cooperatively between human operators and a fleet of quadrotors. One of the layers of software that compose it is a high-level task planner. Its function is to coordinate the entire fleet of UAVs to generate high-level behaviours in order to efficiently, safely and successfully complete the maintenance or inspection mission. This type of work has the characteristic of being dynamic, since it is not possible to know in advance what the outcome of the inspection as such will be in order to plan offline, but rather, as the mission develops, new tasks will arise that the fleet will have to attend to. Therefore, the task planner should be able to react to unexpected events (new task, failure of a UAV, loss of connection, less autonomy than calculated, etc.) and to replan online. Thus, this layer will be the main cognitive block of the system [6].

## 1.2 Objectives

The overall objective of this project was to develop a cognitive task planner that would be in charge of governing the behaviour of multi-UAV teams for the inspection and maintenance of power lines in a collaborative way with human operators, being one of the software layers that compose the aforementioned software architecture [6] developed for the AERIAL-CORE European project. The fleet of governed UAVs acts as an aerial co-worker and can perform various tasks such as delivering a tool to an operator, inspecting regions of the power line or monitoring a worker while operating to ensure his safety. The planner receives both high-level input and feedback from the different equipments that make up the fleet, and processes all the information to elaborate a plan to manage the UAV team or modify it in reaction to an unforeseen event. To achieve this, the following objectives were defined:

- Ensure that resources are used and tasks are executed efficiently.
- Comply with all security requirements and ensure the integrity of equipment and mission success.
- Be able to replan online to react to unforeseen events.

---

<sup>1</sup> AERIAL-CORE European project homepage: <https://aerial-core.eu/>

- Implement the software layer in Robot Operating System (ROS) and manage the necessary communications with the rest of the software layers and modules that make up the architecture.
- Carry out Software In The Loop (SITL) simulations to prove that the algorithm is able to govern the behaviour of the fleet efficiently and safely, and that it is able to react to unforeseen events dynamically, demonstrating cognitive capabilities.



## 2 Preliminaries

---

This chapter focuses on the current state of the art of those technologies related to this project, as well as on the tools used for the development of the task planner as a software layer of a multi-layer architecture. In addition, the research work carried out on the state of the art in work related to the technologies and techniques used in this project is presented.

### 2.1 Current technology

UAV technology has evolved very rapidly in recent years, benefiting from advances in computing. As processors are becoming more powerful, efficient and compact, UAVs are increasingly capable of higher computing power without increasing their weight or compromising their autonomy. With the increase in the number of operations per second that UAVs can perform, this opens up the possibility of using drones for previously unthinkable applications, applications that require a large amount of processing and usually have to be performed in real time.

**2.1.1 UAVs**

**2.1.2 Multi-drone teams**

**2.1.3 Aerial co-workers**

**2.2 Related work**

**2.2.1 Inspection applications with UAVs**

**2.2.2 Task planning in multi-drone teams**

**2.2.3 Drone behavior management**

**2.3 Tools**

**2.3.1 ROS**

**2.3.2 Gazebo**

**2.3.3 Rviz**

**2.3.4 UAL**

**2.3.5 Behaviour Trees**

**2.3.6 Groot**

## 3 Problem Formulation

---

**L**orem ipsum

### 3.1 Description of tasks

3.1.1 Inspection tasks

3.1.2 Monitoring tasks

3.1.3 Tool delivery tasks

### 3.2 Battery recharges

### 3.3 Connection losses

### 3.4 Task replanning situations





## 4 Design of the proposed solution

---

Lo rem itsum

### 4.1 Node diagram

### 4.2 Centralized module: task planner

### 4.3 Distributed module: behavior manager

#### 4.3.1 Main tree

#### 4.3.2 Inspection task tree

#### 4.3.3 Monitoring task tree

#### 4.3.4 Tool delivery task tree

### 4.4 Lower and upper level modules faker



# 5 Results

---

**L**orem ipsum

## 5.1 Task planning

### 5.1.1 Battery

### 5.1.2 Connection lost

### 5.1.3 Replanning

## 5.2 Drone behaviour manager results

### 5.2.1 Battery management

### 5.2.2 Connection lost management

### 5.2.3 Replanning management



# **6 Conclusions and future work**

---

## **6.1 Conclusions**

## **6.2 Future work**

### **6.2.1 Augmented reality**



# List of Figures

---

1.1	Operators getting off the helicopter during a maintenance mission	1
-----	---	---





## List of Tables

---



## List of Codes

---



# Bibliography

---

- [1] H. Shakhathreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, “Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges,” *IEEE Access*, vol. 7, pp. 48 572–48 634, 2019.
- [2] J.-Y. Park, S.-T. Kim, J.-K. Lee, J.-W. Ham, and K.-Y. Oh, “Method of operating a gis-based autopilot drone to inspect ultrahigh voltage power lines and its field tests,” *Journal of Field Robotics*, vol. 37, no. 3, pp. 345–361, 2020.
- [3] H. Baik and J. Valenzuela, “Unmanned aircraft system path planning for visually inspecting electric transmission towers,” *Journal of Intelligent & Robotic Systems*, vol. 95, no. 3, pp. 1097–1111, 2019.
- [4] C. Martinez, C. Sampedro, A. Chauhan, J. F. Collumeau, and P. Campoy, “The power line inspection software (polis): A versatile system for automating power line inspection,” *Engineering applications of artificial intelligence*, vol. 71, pp. 293–314, 2018.
- [5] R. Pěnička, J. Faigl, and M. Saska, “Physical orienteering problem for unmanned aerial vehicle data collection planning in environments with obstacles,” *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 3005–3012, 2019.
- [6] G. Silano, J. Bednar, T. Nascimento, J. Capitan, M. Saska, and A. Ollero, “A multi-layer software architecture for aerial cognitive multi-robot systems in power line inspection tasks,” 2021.
- [7] M. Santarini, “Cosmic radiation comes to asic and soc design,” May 2005. [Online]. Available: <https://www.edn.com/cosmic-radiation-comes-to-asic-and-soc-design/>
- [8] H. G. Miranda, “Aportaciones a las técnicas de emulación y protección de sistemas microelectrónicos complejos bajo efectos de la radiación,” Ph.D. dissertation, Universidad de Sevilla, May 2010.
- [9] J. M. Mogollón, J. Nápoles, H. Guzmán-Miranda, and M. A. Aguirre, “Real time seu detection and diagnosis for safety or mission-critical ics using hash library-based fault dictionaries,” in *2011 12th European Conference on Radiation and Its Effects on Components and Systems*, 2011, pp. 705–710.
- [10] M. G. Valderas, M. P. García, C. López, and L. Entrena, “Extensive seu impact analysis of a pic microprocessor for selective hardening,” in *2009 European Conference on Radiation and Its Effects on Components and Systems*, 2009, pp. 333–336.

- [11] C. Carmichael, "Triple module redundancy design techniques for virtex fpgas," *Xilinx Application Note XAPP197*, vol. 1, 2001.
- [12] Zhou Jing, Liu Zengrong, Chen Lei, Wang Shuo, Wen Zhiping, Chen Xun, and Qi Chang, "An accurate fault location method based on configuration bitstream analysis," in *NORCHIP 2012*, 2012, pp. 1–5.
- [13] W. Tao and W. Xingsong, "Fault diagnosis of a scara robot," in *2008 15th International Conference on Mechatronics and Machine Vision in Practice*, 2008, pp. 352–356.
- [14] S. Jian, J. Jiang, K. Lu, and Y. Zhang, "Seu-tolerant restricted boltzmann machine learning on dsp-based fault detection," in *2014 12th International Conference on Signal Processing (ICSP)*, 2014, pp. 1503–1506.
- [15] R. Pettit and A. Pettit, "Detecting single event upsets in embedded software," in *2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC)*, 2018, pp. 142–145.
- [16] N. Naber, T. Getz, Y. Kim, and J. Petrosky, "Real-time fault detection and diagnostics using fpga-based architectures," in *2010 International Conference on Field Programmable Logic and Applications*, 2010, pp. 346–351.
- [17] Su Wei, Fan Tongshun, and Du Mingfang, "Research for digital circuit fault testing and diagnosis techniques," in *2009 International Conference on Test and Measurement*, vol. 1, 2009, pp. 330–333.
- [18] S. Wei, Z. Shide, and X. Lijun, "Research on digital circuit fault location procedure based on lasar," in *2008 ISECS International Colloquium on Computing, Communication, Control, and Management*, vol. 2, 2008, pp. 322–326.
- [19] B. K. Sikdar, N. Ganguly, and P. P. Chaudhuri, "Fault diagnosis of vlsi circuits with cellular automata based pattern classifier," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 7, pp. 1115–1131, 2005.
- [20] S. S. Yau and Yu-Shan Tang, "An efficient algorithm for generating complete test sets for combinational logic circuits," *IEEE Transactions on Computers*, vol. C-20, no. 11, pp. 1245–1251, 1971.
- [21] S. S. Yau and M. Orsic, "Fault diagnosis and repair of cutpoint cellular arrays," *IEEE Transactions on Computers*, vol. C-19, no. 3, pp. 259–262, 1970.
- [22] V. Amar and N. Condulmari, "Diagnosis of large combinational networks," *IEEE Transactions on Electronic Computers*, vol. EC-16, no. 5, pp. 675–680, 1967.
- [23] D. R. Schertz and G. Metze, "A new representation for faults in combinational digital circuits," *IEEE Transactions on Computers*, vol. C-21, no. 8, pp. 858–866, 1972.
- [24] J. P. Roth, W. G. Bouricius, and P. R. Schneider, "Programmed algorithms to compute tests to detect and distinguish between failures in logic circuits," *IEEE Transactions on Electronic Computers*, vol. EC-16, no. 5, pp. 567–580, 1967.
- [25] A. D. Friedman, "Fault detection in redundant circuits," *IEEE Transactions on Electronic Computers*, vol. EC-16, no. 1, pp. 99–100, 1967.
- [26] R. Zhang, L. Xiao, J. Li, X. Cao, C. Qi, and M. Wang, "A fast fault injection platform of multiple seus for sram-based fpgas," in *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, 2017, pp. 1–5.

- [27] A. da Silva and S. Sanchez, "Leon3 vip: A virtual platform with fault injection capabilities," in *2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools*, 2010, pp. 813–816.
- [28] J. M. Mogollon, H. Guzmán-Miranda, J. Nápoles, J. Barrientos, and M. A. Aguirre, "Ftun-shades2: A novel platform for early evaluation of robustness against see," in *2011 12th European Conference on Radiation and Its Effects on Components and Systems*, 2011, pp. 169–174.
- [29] Wikipedia, "Distancia de levenshtein — wikipedia, la enciclopedia libre," 2020, [Internet; descargado 15-junio-2020]. [Online]. Available: [https://es.wikipedia.org/w/index.php?title=Distancia\\_de\\_Levenshtein&oldid=125248609](https://es.wikipedia.org/w/index.php?title=Distancia_de_Levenshtein&oldid=125248609)
- [30] M. Muñoz-Quijada, S. Sanchez-Barea, D. Vela-Calderon, and H. Guzman-Miranda, "Fine-grain circuit hardening through vhdl datatype substitution," *Electronics*, vol. 8, no. 1, p. 24, 2019.
- [31] "Vhdl implementation of fft algorithm(s)," Available online: <https://github.com/thasti/fft>, accessed on 17 June 2020.
- [32] "Vhdl standard fifo," Available online: <http://www.deathbylogic.com/2013/07/vhdl-standard-fifo/>, accessed on 17 June 2020.
- [33] "Fpga4student. a low pass fir filter for ecg denoising in vhdl," Available online: <https://www.fpga4student.com/2017/01/a-low-pass-fir-filter-in-vhdl.html>, accessed on 17 June 2020.
- [34] "I<sup>2</sup>s interface designed for the pcm3168 audio interface from texas instruments," Available online: <https://github.com/wklimann/PCM3168>, accessed on 17 June 2020.
- [35] "Simple uart controller for fpga written in vhdl," Available online: <https://github.com/jakubcabal/uart-for-fpga>, acceded on 17 June 2020.
- [36] Wikipedia, "Momentos de imagen — wikipedia, la enciclopedia libre," 2020, [Internet; descargado 18-junio-2020]. [Online]. Available: [https://es.wikipedia.org/w/index.php?title=Momentos\\_de\\_imagen&oldid=124767713](https://es.wikipedia.org/w/index.php?title=Momentos_de_imagen&oldid=124767713)
- [37] C. Wolf, J. Glaser, and J. Kepler, "Yosys-a free verilog synthesis suite," in *Proceedings of the 21st Austrian Workshop on Microelectronics (Austrochip)*, 2013.





