

Tesis Doctoral
Ingeniería de Automática, Electrónica y de
Telecomunicación

Distributed approaches for coverage
missions with multiple heterogeneous
UAVs for coastal areas



Autor: Fotios Balampanis
Directores: Jesús Ivan Maza Alcañiz, Aníbal
Ollero Baturone

Departamento de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, Mayo 2018



Tesis Doctoral
Ingeniería de Automática, Electrónica y de
Telecomunicación

Distributed approaches for coverage missions with multiple
heterogeneous UAVs for coastal areas

Autor:

Fotios Balampanis

Director:

Jesús Ivan Maza Alcañiz

Profesor Titular de Universidad

Director:

Aníbal Ollero Baturone

Catedrático de Universidad

Departamento de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Mayo 2018

Tesis Doctoral: Distributed approaches for coverage missions with multiple heterogeneous UAVs for coastal areas

Autor: Fotios Balampanis
Director: Jesús Ivan Maza Alcañiz
Director: Aníbal Ollero Baturone

El tribunal nombrado para juzgar la Tesis arriba indicada, compuesto por los siguientes doctores:

Presidente:

Vocales:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

A mi hermano

Acknowledgments

Writing an acknowledgements section to a doctoral dissertation may seem, at a personal level, more important than the Thesis itself. Perhaps it is the only chance to thank all the people who joined you and are an integral part of it. What we are, in essence, is a set of experiences and personalities we met on our way. This is also an element of science: it does not belong to anyone specific because it belongs to everyone. Everyone has in their own way contributed to the evolution of human intelligence. I wish I was laconic on my acknowledgements, but I will probably not be. How could I omit to thank¹ The Queen, The Who, Pavlos (x2), Richard Bach, Matisse, Nikolas, Tom, Darwin, Aristophanes and Robin?

Initially, I would like to thank my professors, Ivan Maza and Anibal Ollero, for their confidence and guidance. Their experience as well as their clear vision always showed me the path, while they managed to understand me very early on. I must also thank them for their patience.

Then, I would like to thank the colleagues at the University of Seville, the group and the laboratory, for their smile and mood to help me made me feel at home from the very first moment. I would also like to thank the colleagues and travelling companions of the MarineUAS program, a group of extremely smart people. They made the trip extremely pleasant, and they managed to open my mind a little more, they took it a little further.

A special thank you to Margarita Costa Faria whose friendship and polite nature embraced the moments of my nagging and reminded me of trusting people. In particular, I would like to thank Leopoldo Rodriguez, the parallel life co-traveller of these three years. Without him this dissertation would not have been realised. In all the critical moments he made me remember that things are simpler and there is always room for a little more chili. I would also like to thank Juan Braga and Rafael Salmoral, partners in crime, for their help and patience with regard to my Spanish level in this journey. Finally, I would like to thank all the ETSI staff and in particular the cafeteria staff who have given me countless quantities of coffee for three years. I want to apologize to them because I never learned

¹ The cover depicts the port of Agios Kirikos and the statue of Icarus in Icaria island. The photo has been shot by Chrisa Karafti

their names. I want to thank them for their patience to serve people who for some strange reason feel superior in front of the bar, although I personally feel them as fellow travellers.

At this point, I feel that I also have to thank all the people who helped me academically and educationally, from nursery to postgraduate. From Voula to Ulysses. I cannot emphasize enough in these lines how much I appreciate what they have done. Thank you dear teachers.

I also want to thank all my friends. The "family by choice". Those with whom we grew up together, those who we eventually grew apart, those who walked with me until the start of the doctorate, those who visited us in these three years, those who stayed close from so far and made me feel like I never was away, like I had them in my pocket. You all know who you are, in Athens, Barcelona and elsewhere. If you still want to see your name in this dissertation, please fill it in here: Thank you _____. Put the briki on, I'm coming for coffee.

In particular, I would like to thank my sister soul Elisa for everything, but more specifically for tolerating me, for correcting, translating and editing the Thesis.

I would like to thank the family. The small and wider, dynamic, wonderful and ever present family. Those who left us and us who remained. Their trust and their smiling eyes give me strength to continue forever. I am not always beside you (in Kesariani, at the Kesariani Embassy in Ilisia, Gerakas, Leros, Ilion, Schino) but I try .. And I feel you here.

I would particularly like to thank my mother. There are moments I just can not find words. Growing up you realise sacrifices and moves that are indiscernible, but that became gigantic with the passage of time. The Butterfly Effect. You understand how difficult it is to pronounce her name without being moved by the fact that as long as there are people like her there will always be hope. But sometimes I find those words. And I know she knows them now.

I would also like to thank my brother, although this dissertation is dedicated to him, so the words are somehow unnecessary. I always said I did not follow him, but his steps always shed light to my own way, made it easier, nothing would be the same without him. Brother, do not worry about me any more, you have managed as you deserve.

Finally, I would like to thank Maria. Normally, there should have been a chapter for her, even though she would tell me to delete it. First of all I would like to thank her for her patience, love and trust. Thank her for choosing to hold my hand and walk with me. Thank you because while my identity had the public title of a doctoral student, yours in the dark had the titles of nurse, psychologist, chef, secretary, housewife, entertainer, travel agent, confidant, dressmaker, lover, friend, mother, sister. I thank her because she is one of the most courageous and sensitive people I have ever known and she is struggling to keep them both a secret. I thank her for the dreams and the everyday life which looks like the everlasting novel I have always imagined. This Thesis should have your name on the cover and it belongs to you.

*Fotis Balampanis
Seville, 2018*

Agradecimientos

Escribir una sección de agradecimientos para una disertación doctoral puede parecer, a nivel personal, más importante que la tesis en sí. Tal vez sea la única oportunidad de agradecer a todas las personas que han estado abordo de este viaje y que son una parte integral del mismo. Lo que somos, en esencia, es un conjunto de experiencias y personalidades que encontramos en nuestro camino. Esto también es un elemento de la ciencia: no pertenece a nadie específico porque pertenece a todos. Todos a su manera hemos contribuido a la evolución de la inteligencia humana. Desearía ser lacónico en mis agradecimientos, pero probablemente no lo sea. ¿Cómo podría omitir agradecer a² The Queen, The Who, Pavlos (x2), Richard Bach, Matisse, Nikolas, Tom, Darwin, Aristófanes y Robin?

Inicialmente, quiero agradecer a mis profesores, Ivan Maza y Aníbal Ollero, por su confianza y orientación. Su experiencia y su visión clara siempre me mostraron el camino, y lograron entenderme desde el principio. También debo agradecerles por su paciencia.

Luego, me gustaría dar las gracias a los colegas de la Universidad de Sevilla, el grupo y el laboratorio, por su sonrisa y ánimo para ayudarme a hacer que me sintiera como en casa desde el primer momento. También me gustaría dar las gracias a los colegas y compañeros de viaje del programa MarineUAS, un grupo de personas extremadamente inteligentes. Hicieron el viaje extremadamente agradable, y lograron abrir mi mente un poco más, la llevaron un poco más allá.

Un agradecimiento especial a Margarita Costa Faria, cuya amistad y naturaleza cortés abrazaron los momentos de mi regaño y me recordaron de confiar en la gente. En particular, me gustaría dar las gracias a Leopoldo Rodríguez, el compañero de vida paralela de estos tres años. Sin él, esta disertación no se hubiera realizado. En todos los momentos críticos, me hizo recordar que las cosas son más simples y que siempre hay espacio para un poco más de chile. También me gustaría dar las gracias a Juan Braga y Rafael Salmoral, otros cómplices en el crimen, por su ayuda y paciencia con respecto a mi nivel de español durante este viaje. Finalmente, me gustaría agradecer a todo el personal de ETSI y, en particular, al personal de la cafetería que me ha dado innumerables cantidades de café durante tres años.

² En la portada, el puerto de Agios Kirikos y la estatua de Ícaro en la isla de Icaria. La foto ha sido tomada por Chrisa Karafti

Quiero disculparme con ellos porque nunca aprendí sus nombres. Quiero agradecerles por su paciencia para servir a personas que por alguna extraña razón se sienten superiores frente al bar, personalmente son también mis compañeros de viaje.

En este punto, también debo agradecer a todas las personas que me ayudaron académica y educativamente, desde la guardería hasta el postgrado. De Voula a Ulysses. No puedo resaltar lo suficiente en estas líneas cuánto aprecio lo que han hecho. Gracias queridos maestros.

También quiero agradecer a todos mis amigos. A mi «familia por elección». Aquellos con quienes crecimos juntos, aquellos con quienes eventualmente nos separamos, aquellos que caminaron conmigo hasta el comienzo del doctorado, aquellos que nos visitaron en estos tres años, aquellos que se quedaron cerca aunque tan lejos y me hicieron sentir como si nunca estuve lejos, como si los tuviera en mi bolsillo. Todos sabéis quienes sois en Atenas, Barcelona y en otros lugares. Si aún deseáis ver vuestro nombre en esta disertación, por favor completad aquí: Gracias _____. Poned el briki, voy a tomar el café.

En particular, me gustaría agradecer a mi hermana del alma Elisa por todo, pero más específicamente por tolerarme, por corregir, traducir y editar la Tesis.

Me gustaría agradecer a la familia. La familia pequeña y amplia, dinámica, maravillosa y siempre presente. Los que nos dejaron y quienes permanecieron. Su confianza y sus ojos sonrientes me dan fuerza para continuar para siempre. No siempre estoy a vuestro lado (en Kesariani, en la Embajada de Kesariani en Ilisia, Gerakas, Leros, Ilion, Schino) pero lo intento... Y os siento aquí.

Me gustaría agradecer especialmente a mi madre. Hay momentos en los que no puedo encontrar palabras. Al crecer, realizas los sacrificios y gestos que son imperceptibles, pero que se vuelven gigantescos con el paso del tiempo: el Efecto Mariposa. Entiendes lo difícil que es pronunciar su nombre sin que te conmueva el hecho de que mientras haya personas como ella siempre habrá esperanza. Pero a veces encuentro esas palabras. Y sé que ella ya lo sabe.

También me gustaría dar las gracias a mi hermano, aunque esta disertación esté dedicada a él, por lo que las palabras son de alguna manera innecesarias. Siempre dije que no lo seguí, pero sus pasos siempre arrojaban luz a mi camino, lo hacían más fácil, nada sería lo mismo sin él. Hermano, no te preocupes más por mí, has logrado lo que mereces.

Finalmente, me gustaría agradecer a Maria. Normalmente, debería haber un capítulo para ella, aunque ella me hubiera dicho que lo borrara. Antes que nada, quisiera agradecerle por su paciencia, amor y confianza. Agradecerle por elegir tomar mi mano y caminar conmigo. Gracias porque si bien mi identidad tenía el título público de estudiante de doctorado, la suya en la oscuridad tenía los títulos de enfermera, psicóloga, chef, secretaria, ama de casa, animadora, agente de viajes, confidente, modista, amante, amiga, madre, hermana. Le agradezco porque es una de las personas más valientes y sensibles que he conocido y está luchando por mantener a ambas calidades en secreto. Le agradezco los sueños y la vida cotidiana que se parece a la novela eterna que siempre he imaginado. Esta Tesis tendría que tener tu nombre en la portada y te pertenece a ti.

*Fotis Balampanis
Sevilla, 2018*

Ευχαριστίες

Το να γράφεις ευχαριστίες σε μια διδακτορική διατριβή ίσως μοιάζει, σε προσωπικό επίπεδο, σημαντικότερο από την ίδια. Ίσως είναι η μοναδική ευκαιρία να ευχαριστήσεις όλα τα άτομα με τα οποία συμπορεύτηκες και είναι κομμάτι της. Αυτό που στην ουσία είναι ο καθένας μας, είναι ένα σύνολο εμπειριών και προσωπικοτήτων που γνωρίσαμε στο δρόμο μας· αυτό είναι και ένα στοιχείο της επιστήμης: δεν ανήκει σε κανέναν καθώς ανήκει σε όλους. Όλοι έχουν με τον τρόπο τους συμβάλει στην εξέλιξη της ανθρώπινης διανόησης. Μακάρι να ήμουν λακωνικός σε αυτές τις ευχαριστίες αλλά μάλλον δεν θα τα καταφέρω. Πως μπορώ να παραλείψω να ευχαριστήσω τους³ The Queen, The Who, Πάυλο (x2), Richard Bach, Matisse, Νικόλα, Tom, Δαρβίνο, Αριστοφάνη και Robin ·

Αρχικά θα ήθελα να ευχαριστήσω τους καθηγητές μου, Ivan Maza και Anibal Ollero για την εμπιστοσύνη και καθοδήγηση τους. Η εμπειρία καθώς και η καθαρή ματιά τους πάντα μου έδειχναν το δρόμο, καταφέροντας να με καταλάβουν από πολύ νωρίς. Πρέπει επίσης να τους ευχαριστήσω και για την υπομονή τους.

Έπειτα θα ήθελα να ευχαριστήσω τους υπόλοιπους συναδέλφους του πανεπιστημίου της Σεβίλλης, του γκρούπ και του εργαστηρίου, που το χαμόγελο και η διάθεση να με βοηθήσουν με έκαναν από την πρώτη στιγμή να αισθανθώ σαν το σπίτι μου. Επίσης θα ήθελα να ευχαριστήσω τους συναδέλφους και συνταξιδιώτες του προγράμματος MarineUAS, μια ομάδα εξαιρετικά ευφυών ανθρώπων. Έκαναν το ταξίδι εξαιρετικά ευχάριστο, και άνοιξαν το μυαλό λίγο ακόμα, το πήγαν λίγο παραπέρα.

Ένα ιδιαίτερο ευχαριστώ στη Margarita Costa Faria που η φιλία και η ευγενική της φύση αγχάλιασε τις στιγμές της γκρίνιας μου και μου θύμισε να έχω εμπιστοσύνη στους ανθρώπους. Ιδιαίτερα θα ήθελα να ευχαριστήσω τον Leopoldo Rodriguez, τον συνοδηγό με τον παράλληλο βίο αυτά τα τρία χρόνια. Χωρίς εκείνον αυτή η διατριβή δεν θα είχε διεκπεραιωθεί. Σε όλες τις κρίσιμες στιγμές με έκανε να θυμάμαι πως τα πράγματα είναι πιο απλά και πάντα υπάρχει χώρος για λίγο παραπάνω τσίλι. Επίσης θα ήθελα να ευχαριστήσω τους Juan Braga και Rafael Salmoral, άλλους δύο συνένοχους στο έγκλημα, για την βοήθεια και υπομονή τους σε σχέση με το επίπεδο

³ Το εξώφυλλο απεικονίζει το λιμάνι του Άγιου Κήρυκου και το άγαλμα του Ίκαρου στην Ικαρία. Η φωτογραφία πάρθηκε από την Χρύσα Καράφτη

των Ισπανικών μου σε αυτό το ταξίδι. Τέλος θα ήθελα να ευχαριστήσω όλο το προσωπικό του ETSI και ιδιαίτερα το προσωπικό της καφετέριας που για τρία χρόνια μου έδωσαν αμέτρητες ποσότητες καφέ. Θέλω να τους ζητήσω συγγνώμη γιατί ποτέ δεν έμαθα τα ονόματά τους. Θέλω να τους ευχαριστήσω για την υπομονή τους να σερβίρουν ανθρώπους που για κάποιο περίεργο λόγο νιώθουν ανώτεροι μπροστά από την μπάρα, παρόλο που προσωπικά τους ένιωθα συνοδοιπόρους.

Σε αυτό το σημείο νιώθω πως πρέπει επίσης να ευχαριστήσω όλους τους ανθρώπους που με βοήθησαν ακαδημαϊκά και εκπαιδευτικά, από το νηπιαγωγείο μέχρι και το μεταπτυχιακό. Από τη Βούλα μέχρι τον Ulysses. Δεν φτάνουν αυτές οι γραμμές για να τονίσω το πόσο εκτιμώ αυτό που έχουν κάνει. Πάντα σου μένουν στο μυαλό οι παιδαγωγοί και οι δάσκαλοι, όχι οι καθηγητές. Ευχαριστώ δάσκαλοι.

Στη συνέχεια θέλω να ευχαριστήσω όλους τους φίλους μου. Την 'οικογένεια από επιλογή'. Αυτούς που μεγαλώσαμε μαζί, αυτούς που τελικά μεγαλώσαμε χώρια, αυτούς που με συντρόφευσαν μέχρι την εκκίνηση του διδακτορικού, αυτούς που μας επισκέφθηκαν αυτά τα τρία χρόνια, αυτούς που παρέμειναν κοντά από τόσο μακριά και με έκαναν να νιώθω σαν να μην έλλειψα ποτέ, σαν να τους είχα μαζί στην τσέπη μου. Όλοι ξέρετε ποιοί είστε, στην Αθήνα, τη Βαρκελώνη και αλλού. Αν παρόλα αυτά θέλετε να δείτε το όνομα σας στην διατριβή, συμπληρώστε εδώ: Ευχαριστώ _____ . Βάλτε το μπρίκι, έρχομαι για καφέ.

Ιδιαίτερα θα ήθελα να ευχαριστήσω την αδερφή ψυχή Ελίζα για όλα, αλλά πιο συγκεκριμένα που με ανέχτηκε, διόρθωσε, μετέφρασε και επιμελήθηκε γραμματικά και συντακτικά την διατριβή.

Θα ήθελα να ευχαριστήσω την οικογένεια. Τη μικρή και την ευρύτερη, την δυναμική, υπέροχη και πάντα εδώ οικογένεια. Αυτούς που έφυγαν από κοντά μας και αυτούς που μένουμε. Η εμπιστοσύνη τους και τα χαμογελαστά τους μάτια μου δίνουν δύναμη να συνεχίζω για πάντα. Δεν είμαι πάντα δίπλα σας, (στην Καισαριανή, στην πρεσβεία της Καισαριανής στα Ιλίσια, στο Γέρακα, στη Λέρο, στο Ίλιον, στο Σχοίνο) αλλά προσπαθώ... Και σας νιώθω εδώ.

Ιδιαίτερα θα ήθελα να ευχαριστήσω τη μητέρα μου. Υπάρχουν στιγμές που απλώς δε βρίσκω λόγια. Μεγαλώνοντας αντιλαμβάνεσαι θυσίες και κινήσεις της που είναι αδιόρατες, αλλά στο πέρασμα του χρόνου γιγαντώνονται. Το φαινόμενο της πεταλούδας. Κατανοείς το πόσο δύσκολο είναι να προσφέρεις το όνομά της χωρίς να συγκινείσαι από το γεγονός πως όσο υπάρχουν τέτοιοι άνθρωποι θα υπάρχει πάντα ελπίδα. Αλλά μερικές φορές τα βρίσκω τα λόγια. Και το ξέρω πως πλέον τα γνωρίζει.

Θα ήθελα επίσης να ευχαριστήσω τον αδερφό μου, αν και η διατριβή αυτή είναι αφιερωμένη σε εκείνον, οπότε τα λόγια περιττεύουν. Πάντα έλεγα πως δεν τον ακολουθώ, αλλά τα βήματά του πάντα έδιναν φως στο δικό μου δρόμο, τον έκαναν πιο εύκολο, τίποτα δεν θα ήταν ίδιο χωρίς εκείνον. Αδερφέ μην ανησυχείς πλέον για εμένα, τα κατάφερες όπως σου αξίζει.

Τέλος θα ήθελα να ευχαριστήσω τη Μαρία. Κανονικά θα έπρεπε να υπήρχε ένα κεφάλαιο για εκείνη, παρόλο που θα μου έλεγε να το σβήσω. Να την ευχαριστήσω αρχικά για την υπομονή, αγάπη και εμπιστοσύνη της. Να την ευχαριστήσω που επιλέγει να μου κρατάει το χέρι και περπατάμε μαζί. Να την ευχαριστήσω που ενώ η ταυτότητά μου έλεγε δημόσια διδακτορικός φοιτητής εκείνης στο σκοτάδι έλεγε νοσοκόμα, ψυχολόγος, σεφ, γραμματέας, οικιακά, διασκεδαστής, γραφείο ταξιδιών,

εξομολογητής, μοδίστρα, ερωμένη, φίλη, μητέρα, αδερφή. Να την ευχαριστήσω γιατί είναι από τους πιο θαρραλέους και ευαίσθητους ανθρώπους που έχω γνωρίσει και παλεύει να τα κρατήσει κρυφά και τα δύο. Να την ευχαριστήσω για τα όνειρα, για την καθημερινότητα που την κάνει να μοιάζει το διαρκές μυθιστόρημα που πάντα φανταζόμουν. Αυτή η διατριβή θα έπρεπε να είχε το όνομά σου στο εξώφυλλο και σου ανήκει.

Άντε, και καλή τύχη μάγκες...

Φώτης Μπαλαμπάνης
Σεβίλλη, 2018

Abstract

This Thesis focuses on a high-level framework proposal for heterogeneous aerial, fixed wing teams of robots, which operate in complex coastal areas. Recent advances in the computational capabilities of modern processors along with the decrement of small scale aerial platform manufacturing costs, have given researchers the opportunity to propose efficient and low-cost solutions to a wide variety of problems. Regarding marine sciences and more generally coastal or sea operations, the use of aerial robots brings forth a number of advantages, including information redundancy and operator safety.

This Thesis initially deals with complex coastal decomposition in relation with a vehicles' on-board sensor. This decomposition decreases the computational complexity of planning a flight path, while respecting various aerial or ground restrictions. The sensor-based area decomposition also facilitates a team-wide heterogeneous solution for any team of aerial vehicles.

Then, it proposes a novel algorithmic approach of partitioning any given complex area, for an arbitrary number of *Unmanned Aerial Vehicles (UAV)*. This partitioning schema, respects the relative flight autonomy capabilities of the robots, providing them a corresponding region of interest.

In addition, a set of algorithms is proposed for obtaining coverage waypoint plans for those areas. These algorithms are designed to afford the non-holonomic nature of fixed-wing vehicles and the restrictions their dynamics impose.

Moreover, this Thesis also proposes a variation of a well-known path tracking algorithm, in order to further reduce the flight error of waypoint following, by introducing intermediate waypoints and providing an autopilot parametrisation.

Finally, a marine studies test case of buoy information extraction is presented, demonstrating in that manner the flexibility and modular nature of the proposed framework.

Resumen

Esta tesis se centra en la propuesta de un marco de alto nivel para equipos heterogéneos de robots de ala fija que operan en áreas costeras complejas. Los avances recientes en las capacidades computacionales de los procesadores modernos, junto con la disminución de los costes de fabricación de plataformas aéreas a pequeña escala, han brindado a los investigadores la oportunidad de proponer soluciones eficientes y de bajo coste para enfrentar un amplio abanico de cuestiones. Con respecto a las ciencias marinas y, en términos más generales, a las operaciones costeras o marítimas, el uso de robots aéreos conlleva una serie de ventajas, incluidas la redundancia de la información y la seguridad del operador.

Esta tesis trata inicialmente con la descomposición de áreas costeras complejas en relación con el sensor a bordo de un vehículo. Esta descomposición disminuye la complejidad computacional de la planificación de una trayectoria de vuelo, al tiempo que respeta varias restricciones aéreas o terrestres. La descomposición del área basada en sensores también facilita una solución heterogénea para todo el equipo para cualquier equipo de vehículos aéreos.

Luego, propone un novedoso enfoque algorítmico de partición de cualquier área compleja dada, para un número arbitrario de vehículos aéreos no tripulados (UAV). Este esquema de partición respeta las capacidades relativas de autonomía de vuelo de los robots, proporcionándoles una región de interés correspondiente.

Además, se propone un conjunto de algoritmos para obtener planes de puntos de cobertura para esas áreas. Estos algoritmos están diseñados teniendo en cuenta la naturaleza no holonómica de los vehículos de ala fija y las restricciones que impone su dinámica.

En ese sentido, esta Tesis también ofrece una variación de un algoritmo de seguimiento de rutas bien conocido, con el fin de reducir aún más el error de vuelo del siguiente punto de recorrido, introduciendo puntos intermedios y proporcionando una parametrización del piloto automático. Finalmente, se presenta un caso de prueba de estudios marinos de extracción de información de boyas, que demuestra de esa manera la flexibilidad y el carácter modular del marco propuesto.

Η παρούσα διατριβή επικεντρώνεται στην πρόταση ενός δομικού πλαισίου υψηλού επιπέδου για ετερογενή, σταθερής πτέρυγας εναέρια ρομπότ, που λειτουργούν σε πολύπλοκες παράκτιες περιοχές. Οι πρόσφατες εξελίξεις στις υπολογιστικές δυνατότητες των σύγχρονων επεξεργαστών, σε συνδυασμό με το μειωμένο κόστος κατασκευής των αεροσκαφών μικρής κλίμακας, δίνουν στους ερευνητές την ευκαιρία να προτείνουν αποτελεσματικές και χαμηλού κόστους λύσεις σε μια ευρεία ποικιλία προβλημάτων. Όσον αφορά τις θάλασσες επιστήμες αλλά και γενικότερα τις παράκτιες ή θαλάσσιες δραστηριότητες, η χρήση εναέριων ρομπότ παρουσιάζει ορισμένα πλεονεκτήματα, μεταξύ των οποίων είναι και η πλεονάζουσα πληροφορία και η ασφάλεια των χειριστών.

Αρχικά, αυτή η διατριβή ασχολείται με την αποσύνθεση μιας πολύπλοκης παράκτιας περιοχής, σε σχέση με τον αισθητήρα της ρομποτικής πλατφόρμας. Αυτή η αποσύνθεση έχει σαν αποτέλεσμα τη μείωση της υπολογιστικής πολυπλοκότητας του σχεδιασμού ενός σχεδίου πτήσης καθώς και την ασφαλή αντιμετώπιση διάφορων εναέριων ή επίγειων περιορισμών. Η αποσύνθεση της περιοχής που βασίζεται στον αισθητήρα του οχήματος, ενισχύει επίσης τη συνοχή μιας καθολικής λύσης για ετερογενείς ομάδες εναέριων οχημάτων.

Στη συνέχεια, προτείνει μια νέα αλγοριθμική προσέγγιση για τη διαίρεση μιας περιοχής, για οποιοδήποτε αριθμό μη επανδρωμένων εναέριων οχημάτων. Αυτή η πρόταση διαίρεσης και διαμοιρασμού της περιοχής, σέβεται τις σχετικές ικανότητες αυτονομίας της ομάδας των οχημάτων, παρέχοντας σε κάθε ένα μια περιοχή ενδιαφέροντος.

Επιπλέον, προτείνεται ένα σύνολο αλγόριθμων για την απόκτηση σχεδίων πτήσης κάλυψης μιας ολόκληρης υπο-περιοχής. Αυτοί οι αλγόριθμοι έχουν σχεδιαστεί λαμβάνοντας υπόψη τον μη ολονομικό χαρακτήρα των οχημάτων σταθερής πτέρυγας και τους περιορισμούς που επιβάλλουν τα δυναμικά χαρακτηριστικά τους.

Επίσης, αυτή η διατριβή προτείνει μια παραλλαγή ενός γνωστού *path tracking* αλγόριθμου, προκειμένου να μειωθεί περαιτέρω το σφάλμα πτήσης, εισάγοντας ενδιάμεσα σημεία αναφοράς στο σχέδιο πτήσης και παρέχοντας μια παραμετροποίηση του αυτόματου πιλότου.

Τέλος, παρουσιάζεται μια μελέτη σε ένα σενάριο θαλάσσιων επιστημών, η οποία αποσκοπεί στην εξαγωγή πληροφορίας από θαλάσσιους σημαντήρες (buoys), παρουσιάζοντας έτσι την ευελιξία και την αφθρωτή φύση του προτεινόμενου πλαισίου.

Contents

<i>Abstract</i>	XI
<i>Resumen</i>	XIII
<i>Σύνοψη</i>	XV
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	3
1.3 Outline and main contributions	4
1.4 Thesis framework	6
1.5 Scientific output	8
Journal publications	8
Conference publications	8
University stay	9
1.6 Participation in training and other dissemination activities	9
Workshops	9
Poster presentations	10
2 Decomposition of complex coastal areas	11
2.1 Introduction	11
2.2 Related work	12
2.3 Complex coastal areas	16
2.3.1 A safe separation strategy	17
2.3.2 A Decomposition proposal	19
2.4 A Constrained Delaunay Triangulation method	19
2.4.1 FoV based decomposition	22
Simple sensor scenario	22
More complex scenarios	24
2.4.2 Sensor specific - orientation agnostic segregation cells	25
Considered model for the on-board sensors	26

2.4.3	Lloyd optimisation grid enhancement	27
2.5	Results	29
	Comparison with a square grid method	30
2.6	Conclusions	32
3	Area partitioning for multiple vehicles	33
3.1	Related work	34
3.2	Partitioning	35
3.2.1	Baseline Area Partitioning Algorithm	37
3.2.2	Reverse Watershed Schema	39
3.3	Deadlock handling	39
3.4	Results	44
3.5	Conclusions	55
4	Coverage plans	57
4.1	Related work	58
4.2	Waypoint prioritization	59
4.2.1	Spiral coverage plans	60
4.2.2	Spiral plans and area characteristics	61
4.3	Results	65
4.3.1	Simulation framework	66
4.3.2	Comparison with a grid decomposition and simulations	69
	Inlet coefficient method	70
	Sample area	71
	Actual area scenario	74
	Configuration comparison	76
4.3.3	Field experiments	76
4.4	Conclusions	78
5	Path tracking	81
5.1	Related work	81
5.2	Path tracking and following	83
5.3	An offline-online hybrid approach	85
5.4	Results	88
5.4.1	Simulations	88
	Lookahead table extraction	88
	Simulated test flight	90
5.4.2	Field experiments	92
	A tuning control strategy	93
	Pure pursuit flight	98
5.5	Conclusions	99
6	A decentralized framework for coastal missions	101

6.1	A market based decentralized architecture	102
6.2	Test case: a multi-layer framework for buoy coordinated data acquisition	106
6.2.1	Data acquisition layer and buoy specifications	107
6.2.2	Communication layer as a feedback communication control model	108
6.2.3	Single UAV dynamic role reconfiguration	109
	Dynamic team reconfiguration	111
6.3	Experimental setup	112
6.3.1	Buoy	113
6.3.2	Dynamic model for the relay UAVs	113
6.3.3	Energy estimation	115
6.3.4	Throughput estimation	116
6.4	Results	117
6.4.1	QoS Simulations	117
	Enhancing link connectivity	118
	Saving total propulsion energy	119
6.4.2	Role reassignment and team reconfiguration experiments	120
	Team reconfiguration	120
6.5	Conclusions	124
7	Conclusions and future development	125
7.1	Conclusions	125
7.1.1	Decomposition of complex coastal areas	126
7.1.2	Area partitioning for multiple vehicles	127
7.1.3	Coverage Plans	127
7.1.4	Path tracking	128
7.1.5	A distributed framework	128
7.2	Future work	129
	<i>List of Figures</i>	131
	<i>List of Tables</i>	135
	<i>Bibliography</i>	137
	<i>Index</i>	145
	<i>Glossary</i>	147

1 Introduction

Rising fences
Soaring jumpers

LEFTERIS STAMELLOS

In this introduction a brief description of the Thesis motivation, objectives, contributions and scope is presented. Then, the Thesis outline for every Chapter is described, as well as the framework and the project in which it has been developed. Finally, a list of main contributions and project related events is included.

1.1 Motivation

European countries have vast coastal regions which makes their management a challenging task. These regions along with the increased interest of activities in remote and harsh environments, request of a sustainable resource management schema. In order to properly manage and protect these environmentally fragile zones, the need of enabling key technologies is highlighted.

One of these technologies which has recently draw a lot of attention are the teams of unmanned aerial robots, also referred to as *Unmanned Aerial System (UAS)*. The use of these systems offers a number of considerable advantages such as high endurance and relatively low maintenance cost, flexibility, rapid deployment, information redundancy and a decreased risk for the involved operators, in comparison with traditional manned aircraft or in-situ sea vessel operations. The use of smaller and energy efficient vehicles also has a



Figure 1.1 Multiple heterogeneous robots can participate in coastal or marine scenarios, where a ground station (top) monitors the operation of different aerial platforms, for data acquisition from buoys (bottom) in the sea surface.

positive impact on the environment due to significantly lower energy requirements and footprint. In several cases, depending on the application for which these aerial platforms are used, the use of *Unmanned Aerial Vehicle (UAV)* with different characteristics like on-board sensors or energy constraints can be observed. Moreover, synergistic relationships are often created between aerial vehicles and sea surface vehicles or sensors like *buoys*. In these cases, these systems can be characterised as *heterogeneous* (see Fig. 1.1).

Regarding operations in coastal areas, the complexity of the shores along with various aerial restrictions like no-fly zones, domestic or commercial activities as well as reserved airspace zones, impose critical safety considerations to be addressed by any framework.

As it will be explained in this Thesis, some methods in literature propose rather simplistic area decomposition methods which are later used to produce flight plans for the aerial

robots. These flight plans are considered to be followed exactly and the considered obstacles do not impose strict constraints. Moreover, the heterogeneity of a robotic team, especially in aerial vehicles, introduces attributes which are not treated by a uniform framework. Finally, in many cases the *non-holonomic* nature of the aerial robots is not taken into consideration, reducing in that manner the spectrum of application which can be applied to.

The main motivation of this Thesis is the proposal of a high-level framework for teams of fixed-wing aerial robots, in order to achieve complex area coverage, introducing methods which will reduce the flight over no-fly or restricted zones. This framework has to be able to be integrated in any modern, low-cost platform, for mission execution in the context of the MarineUAS project. As such, this framework has to be characterised by individual component modularity, while addressing issues faced in coastal mission scenarios. Even though this Thesis mainly proposes high-level solutions and algorithmic strategies to address those issues, extensive simulated and field experiments are conducted in particular scenarios in order to prove the feasibility and applicability of the system in real world situations.

As described in the following Section, this motivation is analysed in a list of high-to-low level objectives.

1.2 Objectives

The main objectives of this Thesis are:

- To provide a high-level distributed architecture for UAV teams performing coverage tasks. This architecture must take into account heterogeneous vehicle characteristics and account for any complex area.
- To propose a complex area decomposition schema according to the vehicle properties where all area restrictions are taken into consideration.
- To provide platform capability-based area partitioning algorithms, which will assign to individual robots a sub-area of interest to cover. These algorithms have to be able to perform the partitioning methods either in a centralised or in a decentralised scenario, with local or global area information.
- Coverage flight plans must be produced, which will guarantee individual sub-areas coverage, respecting the aforementioned area restrictions.

- To propose path tracking methods which will reduce the flight error, especially over no-fly zones. This constraint is assumed for the non-holonomic vehicles case considered by this Thesis.
- To provide platform parametrisation methods for low-cost vehicles, in order to optimise their overall performance.
- The algorithms and methods presented have to be validated by simulations and tested in actual field experiments.

The following Section presents the outline of the Thesis for every Chapter, where the aforementioned objectives are met.

1.3 Outline and main contributions

The objectives of this Thesis are met and implemented in a modular manner, as presented in Fig. 1.2. Each of the individual parts of the proposed framework is developed and analysed in every Chapter, while being tested in either a simulated environment or in actual field experiments. Then, the last Chapter presents the structure of the whole framework while presenting a case study of its use.

This Thesis contributions for area decomposition, partition and coverage flight plans generation are mainly based on computational geometry algorithms and graphs. The path tracking techniques are based on a state-of-the-art path tracker, found in many modern on-board controllers and autopilots for small aerial vehicles. The proposed distributed framework is based on a market-based architecture. Each of the proposed methods and algorithms has been validated in a *Software-In-The-Loop (SITL)* framework, while the on-line methods proposed have been also tested in extensive field experiments. A summary of the contents of each individual Chapter is presented in the following paragraphs.

This Chapter introduces the motivation and main objectives of this Thesis, while describing an outline of it. Furthermore, it provides some information on the context of the MarineUAS program which this Thesis is part of. Finally, it presents the peer-reviewed papers which were published as well as the project-wide events of MarineUAS.

Chapter 2 introduces a decomposition method for complex coastal areas, by the means of a *Constrained Delaunay Triangulation (CDT)*. This method permits the treatment of any area as a *simplicial complex*. Moreover, it permits the decomposition into a set of triangular cells, each one having a size relative to the projection of the on-board sensor field of view of a UAV to the ground. The proposed strategy is based on the manipulation

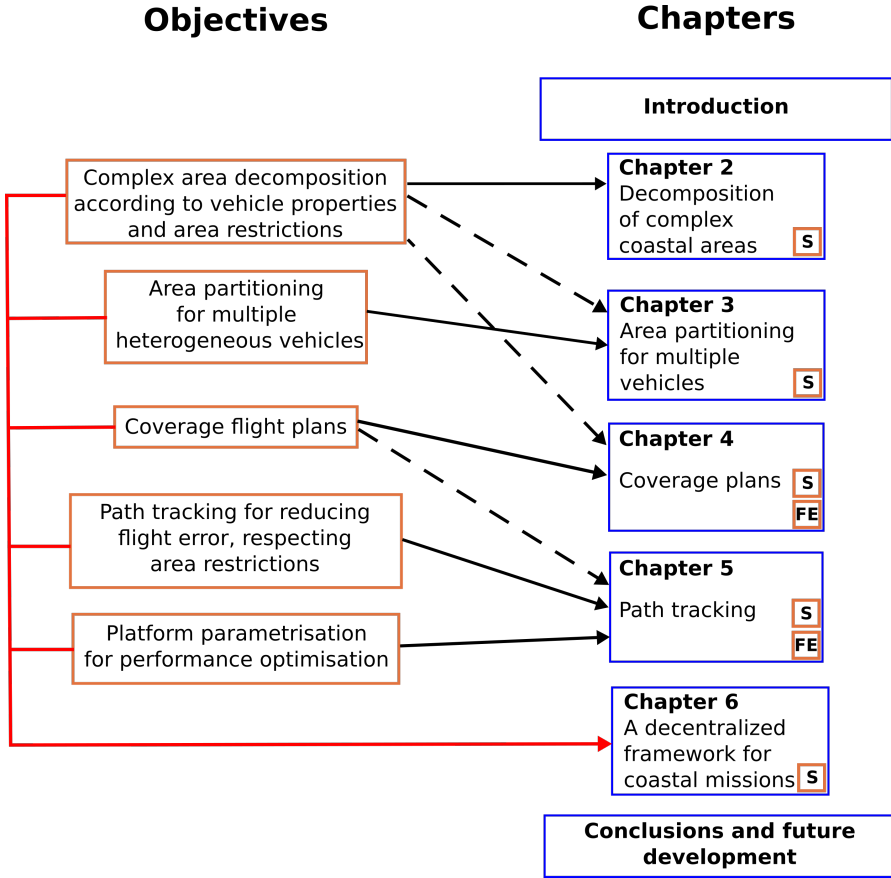


Figure 1.2 The relation between the objectives of this Thesis and the Chapters structure. Each objective is initially met in the Chapter indicated by a solid arrow, while dashed arrows indicate the previously developed module dependencies. **S** and **FE** indexes in every Chapter indicate if **S**imulations, **F**ield **E**xperiments or both have been conducted.

of the area as graph, an approach which is used in the rest of the methods presented in the Thesis.

Chapter 3 uses the aforementioned area-as-a-graph approach in order to propose a set of algorithms for area partition. This schema allows a fair partitioning between an arbitrary number of robots, according to their relative flight capabilities. In that sense, heterogeneous teams with different autonomy capabilities can perform coverage tasks in an optimised manner.

Chapter 4 further utilises the graph representation of the area in order to extract coverage waypoint flight plans. These lists of waypoints are produced by having in mind the non-

holonomic properties of fixed-wing aerial robots and the constraints those impose. Thus, the waypoints lists try to minimise the total path length by not visiting a waypoint or a sub-area twice. At the same time, they try to maximise the overall turn angle of each waypoint-to-waypoint flight plan in order for the vehicles to perform smoother trajectories and reduce the flight over restricted zones.

Chapter 5 proposes a variant of a known path tracking algorithm in order to reduce the flight error between waypoints in the flight plan. Limitations of kinematics and dynamics of fixed-wing vehicles dictate the need of methods to follow a predefined path in an optimal manner. While literature provides a wide variety of path tracking algorithms, practise with *Commercial-Off-The-Shelf (COTS)* platforms shows that uncertainties in motion along with the need of fine tuning of controllers, allow further investigation for practices which can reduce the error. In that Chapter, an off-line method of preflight fine tuning is presented; then, intermediate waypoints at a specific rate are sent during flight.

Chapter 6 introduces the aforementioned methods in a framework which can be part of a market-based architecture. In order to validate the flexibility of the methods proposed as well as the capability of being part of a variety of applications, a test-case is presented where a team of UAVs is used in a multi-layered architecture for data acquisition from sea buoys.

Finally, **Chapter 7** summarises and concludes the Thesis, while proposing some guidelines for further research and experimentation.

1.4 Thesis framework

This Thesis has been conducted as part of MarineUAS¹, a European Union's Horizon 2020 research and innovation programme, under the Marie Skłodowska-Curie grant agreement No 642153. The purpose of this program is to strengthen the research and training on UAS, having in mind the needs of marine and coastal monitoring of the European shores and remote oceanic and polar regions. The program's structure included a wide range of academic and industrial partners from several countries, in order to provide the fellows and researchers a high-level training regarding scientific training and skills acquisition. The main motivation of this program is the need to protect and manage the natural environment and marine resources in a sustainable manner, as has been manifested in European legislation such as the European Strategy for Marine and Maritime Research.

This programme is characterised as an "Innovative Training Network". As such, its goal is to recruit doctoral fellows and provide them intense training by the following means:

¹ <http://marineuas.eu>

- Training-by-research.
- Complementary, interdisciplinary and transferable skills training, e.g. presentation skills training.
- Secondment and academic visits.
- Hands-on UAS operator training.
- Network-wide training events that cover UAS technology.
- Rules and regulations.
- Operations in non-segregated airspace.
- Air traffic management.
- Marine and coastal monitoring and science.
- Integration of the air, surface and underwater segments.

The programme had consortium partners from five universities across Europe as well as seven institutes, organisations and companies. Each of the enlisted early stage researcher fellows was responsible for a project; these projects were mainly focused in control theory and aeronautics. Iceberg detection, performance optimisation of vehicles, cooperative control and cooperative motion planning are some indicating keywords of the projects' spectrum.

In order to fulfil the main objectives of the programme and since the majority of the consortium projects acquired a low-level engineering approach, a complementary organizational level for this Thesis was chosen. As such, this Thesis proposes a high-level, decentralised framework for planning coverage tasks, having in mind the modularity and flexibility that heterogeneous teams of vehicles should have. The research mainly focuses on non-holonomic, *COTS* fixed-wing platforms, in order to follow the experimental framework used by the consortium partners and other fellows. Moreover, the choice of focusing mainly on fixed-wing vehicles was taken due to the fact that they present larger autonomy and durability, a crucial consideration in long range missions and harsh environments. This implies that the proposed methods could easily be adopted in other research efforts in the project as well as in real world applications.

Furthermore, a critical consideration is adopted in this Thesis: the need for successfully integrating the use of UAS in the future *Air Traffic Management (ATM)* context. The increased use of these systems in a variety of scientific and commercial applications, implies an increased risk for collisions between them as well as with current vehicles utilising the airspace. Thus, the safe separation between vehicles for collision mitigation and the

appropriate mechanisms for monitoring each of them, implies a proper decomposition and area partition dynamic schema. The proper treatment of complex coastal areas as well as the reduction of flight error are the main considerations for the developed framework.

In the next section, a list of the published and submitted peer-reviewed publications is presented, along with a description of the MarineUAS university stay.

1.5 Scientific output

The following peer-reviewed publications have been presented in the context of this Thesis:

Journal publications

- Balampanis, F., Maza, I., & Ollero, A. (2017). Area Partition for Coastal Regions with Multiple UAS. *Journal of Intelligent & Robotic Systems*, 88 (2-4), 751–766.
- Balampanis, F., Maza, I., & Ollero, A. (2017). Coastal Areas Division and Coverage with Multiple UAVs for Remote Sensing. *Sensors*, 17(4), 808.

Conference publications

- Balampanis, F., Maza, I., & Ollero, A. (2016, June). Area decomposition, partition and coverage with multiple remotely piloted aircraft systems operating in coastal regions. In *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on* (pp. 275-283). IEEE.
- Rodriguez, L., Balampanis, F., Cobano, J. A., Maza, I., & Ollero, A. (2017). Wind efficient path planning and reconfiguration of UAS in future ATM. In *Twelfth USA/Europe Air Traffic Management Research and Development Seminar (ATM 2017)* (p. 140).
- Balampanis, F., Maza, I., & Ollero, A. (2017, June). Spiral-like coverage path planning for multiple heterogeneous UAS operating in coastal regions. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on* (pp. 617-624). IEEE.
- Braga, J., Balampanis, F., Aguiar, P., Sousa, J., Maza, I., & Ollero, A. (2017). Coordinated Efficient Buoys Data Collection in Large Complex Coastal Environments using UAVs. In *OCEANS MTS/IEEE, Anchorage, AL, 2017 (OCEANS'17 MTS/IEEE)*.

- Rodríguez, L., Balampanis, F., Cobano, J.A., Maza, I., & Ollero, A. (2017, September). Energy-efficient trajectory generation with spline curves considering environmental and dynamic constraints for small UAS. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 2017, pp. 1739-1745.
- Balampanis, F., Aguiar, A. P., Maza, I., & Ollero, A. (2017, October). Path tracking for waypoint lists based on a pure pursuit method for fixed wing UAS. In Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS 2017), Linköping, pp. 55-59.
- Braga, J., Balampanis, F., R. Praveen Jain, Aguiar, P., Sousa, J., Maza, I., & Ollero, A. (2018, May). Dynamic Reconfiguration and Load Management for a UAV Based Buoy Data Extraction Architecture. In OCEANS MTS/IEEE, Kobe (OCEANS'18 MTS/IEEE).
- Balampanis, F., Rodríguez, L., Cobano, J.A., Maza, I., & Ollero, A. (2018). A path enhancement experimental framework for heterogeneous non holonomic flying vehicles. Submitted - Under Review.

Moreover, in the context of the objectives of the MarineUAS project, the following academic stay and workshops have been attended:

University stay An academic visit was made at the Underwater Systems and Technology Laboratory (LSTS) of the Faculty of Engineering in the University of Porto in Portugal, from April of 2017 until July of the same year. During this visit, several path following methods and algorithms have been researched and a path tracking method has been developed in order to reduce the error in flight for non-holonomic vehicles following straight line trajectories. This method has been tested in simulations during the visit, while actual experiments have been conducted after its completion. This study has been published in the Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS) of the same year, as noted in the publications list.

In the next section, a list of the participation in training events of the MarineUAS project is presented.

1.6 Participation in training and other dissemination activities

Workshops

- UAS Technology Workshop (2016, February), Andalucian Foundation for Aerospace Technology - Advanced Centre for Aerospace Technologies (FADA-CATEC), Villacarillo, Spain.
- UAS Operator Training (2016, April), Northern Research Institute (NORUT), Tromsø, Norway.
- Marine and Coastal Surveillance Workshop (2016, April), Maritime Robotics, Trondheim, Norway
- Entrepreneurship workshop (2016, April), Maritime Robotics, Trondheim, Norway
- Ethics workshop (2016, April), Norwegian University of Science and Technology (NTNU), Trondheim, Norway
- Marine and Coastal Science Workshop (2016, June), Instituto Superior Tecnico (IST, Lisboa) and Instituto do Mar (IMAR, Azores), Azores, Portugal
- Workshop on UAS in non-segregated airspace (2017, January), Honeywell, Brno, Czech Republic
- MarineUAS First summer school (2017, June), Instituto Superior Tecnico (IST, Lisboa), Lisbon, Portugal
- Presentation skills workshop (2017, June), Instituto Superior Tecnico (IST, Lisboa), Lisbon, Portugal

Poster presentations

- Balampanis, F., Maza, I., & Ollero, A. (2016, April). Area Decomposition and Path Planning in Coastal Regions, Remote Controlled and Autonomous Measurement Platforms Flagship (ReCAMP) Workshop, Northern Research Institute Tromsø AS (NORUT), Tromsø, Norway.
- Balampanis, F., Maza, I., & Ollero, A. (2017, January). Distributed approaches for coverage and tracking missions with multiple heterogeneous UAVs for coastal areas, MarineUAS Mid-Term Internal Research Review, Honeywell, Brno, Czech Republic.
- Rodríguez, L., Balampanis, F., Cobano, J.A., Maza, I., & Ollero, A. (2016, May) MarineUAS Innovative Training Network on Autonomous Unmanned Aerial Systems for Marine and Costal Monitoring, Spanish RPAS Association Conference UNVEX 2016, Madrid, Spain.

2 Decomposition of complex coastal areas

*Let me see you stripped
down to the bone*

DEPECHE MODE

Two main steps are required for a heterogeneous team of robots to be able to perform a coverage task in a given area: (i) a fair partitioning of the area according to the robotic capabilities and (ii) a feasible path planning method. In this Chapter, we present a decomposition method of the area in order to reduce the complexity of those steps and facilitate an online implementation of the solution. In particular, the identified properties of coastal areas for aerial vehicles show that an exact decomposition is needed to ensure complete coverage. Furthermore, a proper decomposition schema is a crucial consideration in order to mitigate the risks of flying unmanned robots near populated areas. Finally, by decreasing the complexity of the area by the means of a *CDT*, the partition and path planning problems can be reduced to graph search problems.

2.1 Introduction

Area decomposition in robotics for coverage, is a widely used method which is performed by a discretisation of a previously known area. Decomposing an area into a collection of sensor-sized cells and treating it like a complete, sensor specific graph, some common tasks

are: obtaining a complete coverage path, increase the information gain and decrease mission time and energy needs. Moreover this decomposition reduces the overall computational complexity by simplifying the area and thus, it can be used to assign the task to a number of robotic vehicles. Then, by formulating a path formation strategy in the decomposed parts, various algorithms are used for the aforementioned tasks. In literature, several approaches can be identified for area decomposition, with *Grid* based methods and *Cellular Decomposition* be the most common. As will be described in the following sections, several of these approaches do not account for complex shapes or they simplify the problem, by not considering the heterogeneity of the robotic vehicles and their on-board sensors.

Regarding coastal areas, the complex geographical attributes, the vast populated coasts and the increasing interest for activities in even remote off-shore locations, open questions regarding marine environment protection and sustainable management. The attributes of these areas, as well as the increased complexity of airspace management raise challenges in the integration of autonomous or remotely piloted aerial vehicles. More specifically, the integration of *UASs* into future airspace is one of the greatest challenges in *ATM* of the *Single European Sky ATM Research (SESAR)*¹ and the *National Aeronautics and Space Administration (NASA)* [1]. The use of *UAS* to cover wide areas implies the consideration of airspace restrictions and static and dynamic obstacle avoidance. This results in complex shapes that need to be decomposed adequately to ensure coverage by using the variety of heterogeneous *UAVs* in use today.

Having in mind the aforementioned properties of coastal areas, the method used in this Thesis is a *CDT* [2]. This method manages to decompose complex areas by having non-convex polygons with holes as an input and performs an exact decomposition regardless of the area complexity. In that manner an area is decomposed in a grid-like manner, having each of the triangular cells as an on-board sensor *Field of View (FoV)* sized triangle. Treating the resulting grid as a graph, the coverage problem can be reduced in a graph search problem, where every edge must be visited at least once. These algorithms are described in detail in the next Chapters.

2.2 Related work

Most current decomposition approaches reduce or simplify the complexity of the area to be visited. In that manner, the vehicle has discrete states that can be reached by choosing different actions. Different decisions are required for tasks such as coverage or tracking,

¹ <http://www.sesarju.eu/>

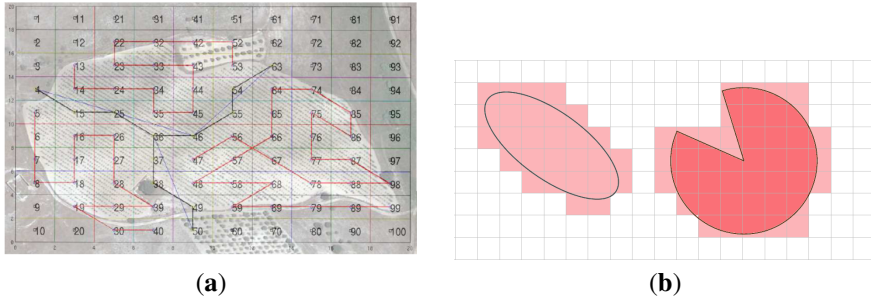


Figure 2.1 A problematic found in grid decomposition methods. In (a), Fig.6 from [5] is presented, where in order to achieve complete coverage, the path passes over cells which are partially not in the region of interest, like cells 4, 22, 32 etc. In (b), a case as adopted from [4]. Big regions are considered as part of the obstacles, even though they are not.

and cost-based path planning algorithms can be applied to reach each state [3]. Numerous studies have extended these algorithms to multiple robots, swarms or distributed systems.

Regarding the decomposition process, the following approaches can be identified in literature: Grid based methods, Cellular decomposition and Boustrophedon (or Morse) decomposition.

In grid based methods, which are also referred as “resolution complete methods” [4], the area partitioning is performed by applying a grid overlay on top of the area. In that way, there is a discrete simplicial complex where if all the cells are visited, then complete coverage is assumed. The decomposition problem is somehow bypassed, as for areas that are partially covered by the cells, the vehicle might not cover them at all in order to avoid collisions, or will go over them, ignoring the fact that they are obstacles or no-fly zones [5, 6] (see Fig. 2.1).

In [7] even though triangles are used for the decomposition of the area, the same problem is still present. By changing the shape of the cells, while arranging them regardless of the area to be decomposed, the problem of having cells which are partially outside the domain of interest still remains. In general, all grid based methods tend to “pixelate” the area, a strategy that does not properly solve real world scenarios; areas do not have rectangular borders, neither do their obstacles and no-fly zones. Galceran et al. states in [4] that this problem can be solved by increasing, locally or globally, the resolution of the grid, creating smaller cells. This actually does not solve the problem in our case, since the dynamics of a fixed wing UAV have to be considered. Increasing the resolution does not guarantee a successful passage over the smaller cells since this operation implies sharper turns in a smaller area. Moreover, we have not identified in literature an algorithm that manages to

visit all cells of varying resolution and to avoid repeated coverage. This Thesis considers the size of the projected *FoV* of the *UAV* sensor for the decomposition method, where the resulting graph is also part of the path planning algorithms, as will be described in the following sections and chapters. As noted before, an exact decomposition in cells which the vehicle can visit, increases computational efficiency and ensures full coverage. If the cells are to be smaller, there is no guarantee that a feasible path for complete coverage can be produced.

In cellular decomposition, a non-convex polygonal area is decomposed into a collection of convex polygons. Even though it manages to decompose complex areas, it does not always consider obstacles [8] within the area or it assumes that the resulting cells will be covered by simple and coherent back and forth lawnmower motions. In Boustrophedon decomposition [9] the problem of only producing many convex polygons without holes has already been addressed. In this strategy and its derivatives, critical points in the area are found and used in order to divide the area in cells. Even though this method is widely used, it still does not address the problem of covering the whole area by respecting the boundaries, as simple boustrophedon movements of the vehicles might not be able to cover complex spaces [10], or the resulting path planning algorithm produces repeated coverage paths [11]. Furthermore, some studies consider movements outside the region of interest in order to take turns and thus provide a full coverage pattern (see Fig.2 in [8] and Fig.2, Fig.5 in [12]). In [13], the authors use the aforementioned methods in order to perform complete coverage of an area. By using a variation of Morse functions, the area is segregated by performing a sweeping motion. In that manner, a set of trapezoidal cells is created and treated as a flow network. Nevertheless the results show that more complex areas and obstacles might not be treated properly by the coverage paths produced.

In almost all of the above cases, it can be noticed that the surrounding area is rectangular and fairly simple. Moreover, most consider a coverage by lawnmower or boustrophedon paths, assuming that the vehicles will always be able to compensate uncertainties in motion or external factors like wind drifts for aerial vehicles, or surface anomalies for ground vehicles. As it can be seen in Fig. 2.2, coastal areas are usually far from being simple, convex polygons.

In [14], the authors provide an optimal decomposition and path planning solution using an Integer Linear Programming (ILP) solver, by taking into account camera-sized grid decomposition. Their solution manages to obtain the desired optical samples, although they do not provide the computational time needed for the ILP to find a solution. In the same context, the authors in [8] use an enhanced exact cellular decomposition method for an area and provide a coverage path consistent with the on-board camera of the *UAV*. Although



Figure 2.2 A coastal area complexity example, Salamina island and Piraeus port in Greece. Numerous residential and commercial areas in a 12Km^2 area, define narrow passages and complex shore shapes.

their solution manages to produce smooth paths with minimal turns, their algorithms are tested only over convex polygon areas. The work presented in [15] deals with the same problem of area coverage for photogrammetric sensing. The authors include energy, speed and image resolution constraints in their proposed algorithms, such as an energy fail-safe mechanism for the safe return to the landing point. However, the provided solution and experiments do not account for complex, non-convex polygonal areas.

Literature also provides a number of bio-inspired approaches that manage to apply strategies that have been observed in nature. In these studies like [16], the behaviour of insects leaving pheromones on their path is used to draw the boundaries of the covered area. In [17] the authors provide a particle filter strategy for a team of robots using bio-inspired swarm tactics in order to communicate the shape of the region of interest or anomaly as they name it in their study. In both cases, the areas are simple environments and a complete coverage is not guaranteed. The work presented in [11] decomposes an area by using convex decomposition and produces parallel lines in the decomposed parts, which are used as straight line paths. The algorithm presented tries to minimize the total amount of turns and provides a complete coverage plan. This strategy produces promising results, but complete coverage is not always achieved. Moreover, in some cases, repeated coverage is performed in order for the vehicle to visit the initial position of the next decomposed

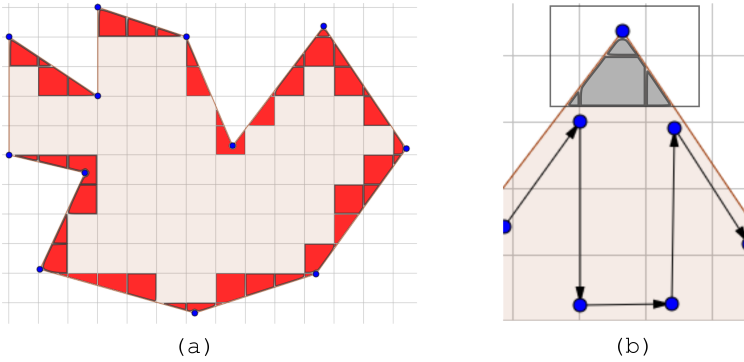


Figure 2.3 In square grid decomposition, some problematic cases can be identified: in (a) there is a border cells ambiguity of visiting or not, having part of them outside of the area of interest. In (b) a boustrophedon motion for coverage, where sharp characteristics or complex shapes create areas that can not be visited.

region.

In most of the aforementioned studies, two strategies can be identified for explicit area division into regions of parallel coverage by multiple UAVs like in [18]. The first is the exact decomposition of the unblocked area of interest into sub-regions. This exact cellular decomposition is performed by either using a trapezoidal grid overlay or by using Morse functions, finding critical points of obstacles or area edges, like in [8] or [19]. The second strategy uses an approximate decomposition, usually by applying a grid overlay of equally sized shapes on top of the area like in [6] or [5]. In this strategy, some grid cells might be partially blocked by the terrain, creating an approximate partitioning while some of them use a wavefront algorithm [20] such as in [21] in order to produce paths in known terrain. In that manner, complex shapes are not considered or the issue of covering or not the border cells is bypassed, as it can be seen in Fig. 2.3a. In case a boustrophedon motion is chosen [9], some areas might be left uncovered, as it can be seen in Fig. 2.3b.

This Thesis combines the two aforementioned families of approaches as it performs an exact triangulation of the non convex polygonal area and computes several cell attributes for multi-UAV area partition and coverage, by using a variation of a wavefront-like propagation algorithm, as we shall describe in the following chapters.

2.3 Complex coastal areas

Applying algorithms in real world situations often means that the application will highlight problems that have to be globally addressed by the method, although they were not initially

considered by the researcher. Instead of trying to generalise a simple shape decomposition algorithm, like a square grid method, for complex shape situations, this work assumes a priori that coastal areas are complex. Their borders have few straight lines, in many cases no-fly zones exist in the middle of the area (e.g. residential or commercial zones) and, as it becomes clear in Fig. 2.2, they require dealing with non-convex polygons with numerous no-fly zones.

Coastal areas have distinguishable properties that permit the use of additional constraints in the algorithmic formulation. As a starting point, in scenarios involving flights over the sea, it is safe to assume that except obstacles and no-fly zones, the remaining area is in flat, sea level altitude. Moreover, flying over the sea means that maintaining a constant altitude, the field of view of a camera or in general the footprint of a sensor, is always relatively stable. This allows the segmentation of the area in a minimal sum of footprint-sized cells to have a complete, on-board sensor specific coverage. Nevertheless, the increased interest of using *UAVs* for commercial purposes on coastal areas in the context of the MarineUAS project, reflects on several mission scenarios which surpass the use of specific waypoint airways. As described in the Introduction, the safe separation constraints in the context of *ATM* management introduce further restrictions, making the decomposition schema even more complex.

2.3.1 A safe separation strategy

IVAN: this section indeed is probably better to be moved in the introduction, as the new last sentence of the previous section proposes. If the latter is a good introduction for this section, then it could remain here. Let's talk in person for that.

As noted before, this Thesis strongly focuses on respecting the aerial restrictions of a flight in the context of the MarineUAS project. This requirement is crucial since the safe integration of *UAVs* in non-segregated airspace is a key requirement of the Single Sky European Research (SESAR)²; an exact decomposition of an area and the avoidance of flight over residential, commercial or restricted areas are ways to mitigate critical damage in case of system failures.

Potential research and commercial *UAS* applications including goods delivery, search and rescue, among others, require a precise set of rules to ensure safety and reliability of the involved actors. In the context of applications which require area coverage in a non-segregated airspace, there are many aspects which need to be considered. Smart path planning is a key area that needs to be studied in order to ensure that any given task does not compromise the safety of the airspace in which it is being performed. Current

² <http://www.sesarju.eu/newsroom/all-news/sesar-takes-next-steps-rpas>

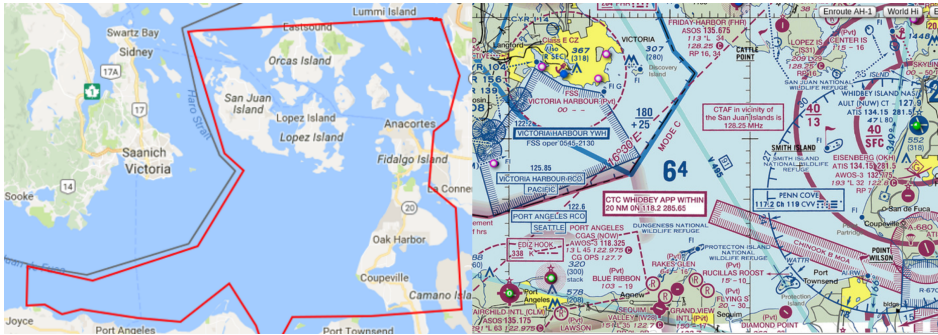


Figure 2.4 On the left: a test area in Seattle(WA,USA) denoted by the red polygon. On the right, several low altitude restrictions apply in the area, like those on the bottom right corner, as part of a local training center. The proposed system should also respond to any online and unscheduled restriction. The image on the right is a courtesy of SkyVector.com.

and future *ATM* imply complex and dynamic areas which represent numerous challenges in mission planning. Regarding light and small *UAVs*, the National Aeronautics and Space Administration (NASA) has proposed the development of the (*Unmanned Air Traffic Management (UTM)*) [1] system for Low-Altitude *UAVs* as a response to safely manage *UAS* in airspace not regulated by the (*Federal Aviation Administration (FAA)*). This effort involves the participation of very important partners such as Amazon, Google, Lockheed Martin Corporation, Honeywell, etc. In addition to this set of rules, one important concern of the airspace that is managed by the civil authority is the need to provide a reliable Detect-and-Avoid capability as exposed by Haessig et al. [22], in which technologies such as (*Autonomous Dependent System-Broadcast (ADS-B)*) are explored as a means to resolve potential conflicts with cooperative obstacles. Research on the topic is extensive, including the studies presented by Cordon et al. [23] and Paczan et al. [24] which provide an insight, from a systems perspective, of different aspects of the integration of *UAS* in both NextGen and SESAR contexts. In [23] the authors present the requirements of the interfaces that are needed for the different phases of flight of a *UAV* including the mission/flight preparation, as defined by the SESAR Concept of Operations (CONOPS)³; the Business or Mission Development Trajectory (BDT/MDT) is a key element on the future *ATM* and *UTM* operations, since mission planning, especially in areas in which the air traffic is dense, represents a considerable challenge (see Fig. 2.4).

³ http://www.sesarju.eu/sites/default/files/documents/highlight/SESAR_ConOps_Document_Step_1.pdf

2.3.2 A Decomposition proposal

Having in mind the aforementioned complexity of coastal areas as also as the strict nature of the present and future legislation as mentioned in the Introduction, a grid decomposition method seems as suitable strategy for area characterisation of no-fly zones. As mentioned before, a common method in literature is grid overlay over an area. As such, a rectangular mesh is produced and by visiting all the allowed squares, complete coverage is considered. A mesh is a partition of an area into shapes which satisfy several criteria. The mesh can also be defined as a graph of vertices that can either be disjoint or share an edge. These vertices can either be part of the boundary of an area or internal constraints of the meshed region. However, a mesh can also be produced by a *Delaunay Triangulation* of an area, as described in the next section. More specifically, in the *Computational Geometry Algorithms Library (CGAL)* [25] used in the implementation of this Thesis, the user can define edge restrictions and seed points that either mark regions to be meshed or regions not to be meshed. In the latter case, the whole constrained region is initially triangulated but not meshed. This meshing algorithm constantly inserts new vertices to the *Delaunay Triangulation* as far as possible from the other vertices, and stops when the refinement criteria are met. The algorithm is guaranteed to terminate satisfying the requested criteria “if all angles between incident segments of the input planar straight line graph are greater than 60 degrees and if the bound on the circumradius/edge ratio is greater than $\sqrt{2}$ ” [26].

As such, the method used in this Thesis produces a mesh which includes a collection of all the edges and vertices of the triangulation as well as user defined attributes. The Delaunay mesher of the CGAL library is responsible for the refinement of the triangulation. Moreover, each produced facet (cell) also has specific attributes, e.g its neighbours, while it is possible to include additional properties such as weights related to the information gain, distance costs, etc. discussed in the next Chapters.

2.4 A Constrained Delaunay Triangulation method

Let us consider a cooperative team of autonomous and heterogeneous *UAVs* and the following scenario: a group of buoys is expected to be submerged from the sea at an approximate region R of a coastal area, with particular aerial restrictions due to reserved airspace, nearby airports and domestic regions. Consider that the expected locations of the buoys are approximate and the location probability is decreasing uniformly in all directions on the sea from the initial belief. A team of *UAVs* can be used for remote sensing purposes in order to localise those buoys around the expected locations. In a cooperative framework,

it is assumed that the task is *divisible* and *parallelisable* in such a way that the task can be partitioned and every subtask is assigned to an UAV in the team.

Let R be the total coastal area to monitor and each sub-area is a subset of R . Let U_1, U_2, \dots, U_n be the team of n UAVs with initial locations $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$. These locations are relevant to the whole procedure since they are the places of where the buoys are expected to be found.

As the UAVs are heterogeneous, each UAV may have a different FoV projection to the surface of the sea. Each sub-area is decomposed into v_1, v_2, \dots, v_{m_i} FoV-sized cells as close as possible to the size of the footprint of the onboard sensor of the respective UAV to the sea. As such, area R_i has a total of m_i cells depending on that FoV of the respective U_i UAV.

The goal of this Chapter is the development of a common framework to solve area decomposition for a multi-UAV team. In particular the following requirements are considered:

- The area should be exactly decomposed regardless of its complexity or the existence of no-fly areas, without cells being outside or partially inside the areas of interest. Then, each sub-area should be the exact sum of its decomposition into cells such that

$$R_i = \sum_{k=1}^{m_i} v_k \quad (2.1)$$

- Every simplicial complex R_i of each UAV cannot be disjointed or intersected by another complex or obstacle. Resulting partitioned areas have to prevent overlapping coverage paths.

By performing a refined *CDT*, there is no segmentation outside the area of interest. Moreover, by visiting all triangles, it could be greedily assumed that the whole area has been covered. From the implementation point of view, the *CDT* and mesh generation of the *CGAL* library [25] algorithms have been used, introducing a top limit of edge side for each triangle: the *FoV* of the UAV as will be described in Section 2.4.1. The same area as in Fig. 2.2 can be seen as a *CDT* mesh in Fig. 2.5. Furthermore, the triangles should be as much homogeneous as possible in order to minimize overlap in sample acquisition we will see in the following sections; a lesser angle of 35 degrees is used for every vertex of the triangular cells.

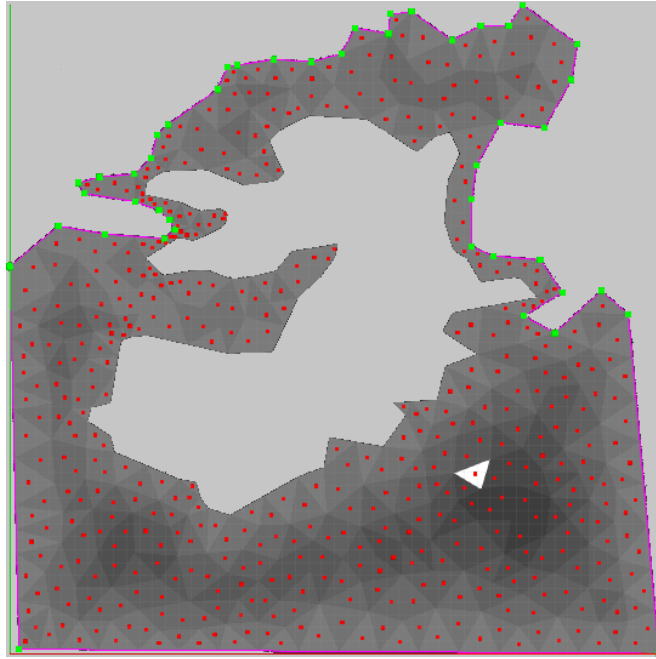


Figure 2.5 Salamina area as a CDT mesh derived by the proposed method. Gray areas are the no-fly zones, while the red dots represent the centroids of each triangular cell.

Basic triangulation has no control on the resulting shape and produces skinny triangles having extreme angles. In contrast, a Delaunay Triangulation $DT(H)$ is a collection of triangles in which all points $H = \{h_1, h_2, \dots, h_m\}$ that belong to H are not inside the circumcircle of any triangle in this set. A Constrained Delaunay Triangulation $CDT(H)$ [2] introduces forced edge constraints which are part of the input, and are useful in this study as they actually define the boundaries of the polygonal area and the holes inside the polygon. This method, tends to maximize the minimum angle of all the triangles in the triangulation, thus providing more homogeneous triangles. Moreover, by connecting the centres of the circumcircles, a geometric dual of a Voronoi diagram is produced. By using this method, we create a mesh or grid of regions of interest. The authors in [27] provide a detailed description on the algorithm's complexity and propose yet another solution for a fully Dynamical CDT with $O(n \log n)$ complexity, where n is the number of vertices. In order to obtain even more homogeneous triangles, a Lloyd optimisation [28] is also applied on the resulting triangulation, as will be described in Section 2.4.3. The following section describes the calculation of the aforementioned lesser side size.

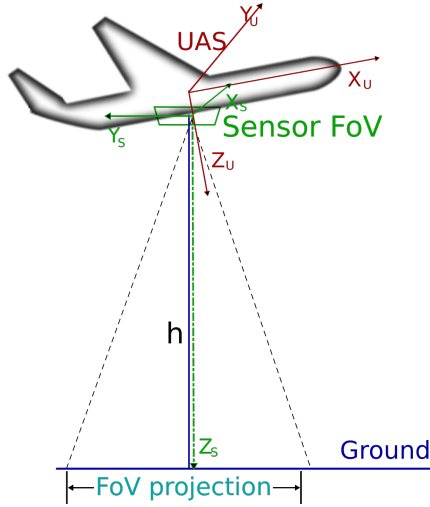


Figure 2.6 The coordinate frames of the system. In presence of a gimbal which compensates for the movements of the UAS, the sensor always points down, parallel to the ground.

2.4.1 FoV based decomposition

Let us consider that function $P(t)$ calculates the area projection of the on-board sensor Field of View (FoV) to the ground at a specific moment t during the flight. This function is dependent on the relative rotation matrices among the UAS frame F_U , sensor frame F_S and ground F_G coordinate frames, as it can be seen in Fig. 2.6.

The usual convention in aviation is used for the UAV movements; roll (γ), pitch (β) and yaw (α). The UAV coordinate frame has its x -axis pointing forwards in relation with the movement, the y -axis is given by the right-hand rule, while the z -axis points downwards. There are also cases where a gimbal is used in order for the UAV attitude to become irrelevant and the gimbal to compensate for the UAV movements, or its specific orientation to translate the sensing angle.

Simple sensor scenario

In a simple scenario where a UAV is equipped with a camera, mounted in a gimbal in order to compensate for the movements of the platform and facing downwards in a zero angle 0° like in Fig. 2.6, the FoV projection will be similar to the one shown in Fig. 2.7.

As such, the FoV angles of θ_x and θ_y , draw the projected rectangle $P(x_p, y_p)$ in relation with the flight altitude h :

$$x_p = 2h \tan\left(\frac{\theta_x}{2}\right) \quad (2.2)$$

and

$$y_p = 2h \tan\left(\frac{\theta_y}{2}\right) \quad (2.3)$$

Then, the largest inscribed circle C_P of P is the one that has a radius r equal to the smallest side of P : P_{min} . Hence, the largest triangle W that can be inscribed in this circle is the equilateral triangle W_P and its side W_d is

$$W_d = \sqrt{3} \frac{P_{min}}{2} \quad (2.4)$$

Thus, as it will be described in the next section, the desired triangulation for the mesh generation is a CDT with maximum side equal to

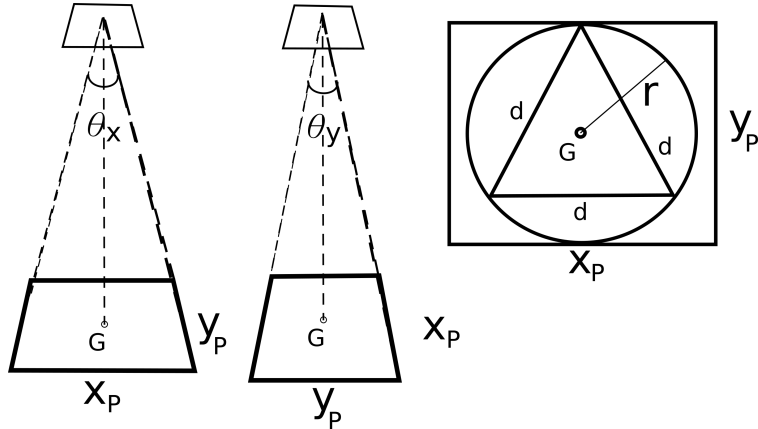


Figure 2.7 Calculating the CDT side constraint in a simple scenario. In the existence of a gimbal, the FoV projection is in a zero angle 0° , parallel to the ground. Thus, the maximum triangle that can be inscribed, is the inscribed equilateral triangle of the trapezoid's inscribed circle.

$$W_d = \frac{2\sqrt{3}h \tan\left(\frac{\theta_{min}}{2}\right)}{2} \quad (2.5)$$

If the camera takes a picture in a $d\omega$ interval, which depends on its frame rate, in the area constrained by the previously mentioned triangle, then by the time Ω it will have finished or stopped its task and the covered area will be

$$A = \int_0^\Omega P(\omega) d\omega = \int_0^\Omega \left[\frac{\sqrt{3}}{4} W_d^2 \right] d\omega = \int_0^\Omega \left[\frac{2\sqrt{3}h \tan\left(\frac{\theta_{min}}{2}\right)}{2} \right] d\omega \quad (2.6)$$

in relation with the FoV smaller angle and altitude. This coverage metric only measures the area covered by the triangles and not by the UAS or the complete FoV projection.

More complex scenarios

In case the sensor is tilted in one or two axes, with the yaw axis $z = 0$ and the gimbal compensates for the platform movements in order to keep the sensor stable, then the resulting FoV projection shape is either a trapezoid or an oblique projection.

In the first case, where the sensor is only tilted in the Y_S axis in relation with the ground as seen in Fig. 2.8, a pitch angle β of the platform along with the bisector of the FoV angle θ , compose the front angle ψ , as it has been adopted by [8].

The ground FoV projection forms a trapezoid $T = a, b, c, d$, having the bottom side a equal to

$$a = \frac{2h \tan\left(\frac{\theta_y}{2}\right)}{\sin(\psi - \beta + \frac{\theta_x}{2})} \quad (2.7)$$

and upper side b equal to

$$b = \frac{2h \tan\left(\frac{\theta_y}{2}\right)}{\sin(\psi + \beta + \frac{\theta_x}{2})} \quad (2.8)$$

where θ_y and θ_x are the FoV angles of the camera on the Y_S and X_S axe respectively, of Fig. 2.6. Then, the largest triangle W that can be inscribed in the trapezoid circle is the equilateral triangle W_p and its side W_d is

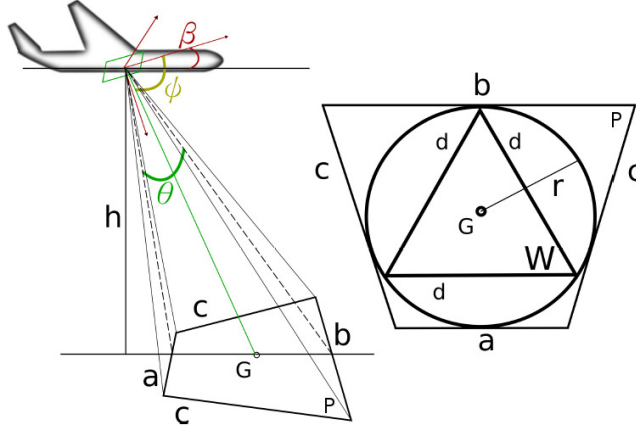


Figure 2.8 Calculating the CDT side constraint in case the sensor is tilted in one or more axis [8]. The angle θ is the FoV angle as in the previous examples. Pitch angle β along with the bisector of θ compose the front mounted angle ψ . Then, the ground projection is a trapezoid and the maximum triangle that can be inscribed, is the inscribed equilateral triangle of the trapezoid's inscribed circle.

$$W_d = \frac{\sqrt{3ab}}{2} = \sqrt{3 \frac{2h \tan(\frac{\theta_y}{2})}{\sin(\psi - \beta + \frac{\theta_x}{2})} \frac{2h \tan(\frac{\theta_y}{2})}{\sin(\psi + \beta + \frac{\theta_x}{2})}} \quad (2.9)$$

Once again, as it will be described in the next section, the desired triangulation for mesh generation is a CDT with maximum side equal to 2.9. The calculation of the coverage is done in the same way as in Eq.(2.6), replacing the size of W_d .

Even more complicated scenarios are tackled by the authors in [29] and [10], while [8] provides a detailed calculation method. In this Thesis, we use a downward facing, zero angle, parallel to the ground scenario, where a gimbal compensates for the UAV movements.

2.4.2 Sensor specific - orientation agnostic segregation cells

The aforementioned calculations manage to provide complete coverage formalisation in the simplicial complex. Having specific information about the sensor, namely its FoV and its sampling rate, the problem can be reduced to a discrete sampling scenario, by creating a grid where every cell is a sample. By performing a discretisation of the area according to the sensing capabilities and characteristics of the sensor, the CDT is applied by using a maximum triangle side constraint which can be calculated from the FoV of the

sensor as described in the previous section. Since the calculated triangle is inscribed in the projection's inscribed circle, if every centre of each circle is considered as a waypoint, the configuration manages to produce segregation cells that are always covered by the sensor FoV if all waypoints are visited, no matter the flight direction; thus, this strategy is identified as *orientation agnostic*.

Then, the integral function in Eq.(2.6) for coverage calculation in continuous space can be simplified as the sum of m cells, for every segregated cell v_k of the triangulation; since we take a sample in every discretised cell, the sum of all cells provides a complete coverage metric $\sum_{k=1}^m P(v_k)$, where the area of each cell v_k can be calculated as

$$v_k = \sqrt{s(s - v_a)(s - v_b)(s - v_c)} \quad (2.10)$$

having s as the semiperimeter of the triangle and a, b, c as the sides; hence the decomposition cells are platform and sensor specific. Please note that this calculation rule also applies in case the resulting FoV projection is not a square or a trapezoid, but a tangential quadrilateral. It is always possible to obtain the inscribed circle and the respective inscribed triangle as a function of the FoV sides. In case the FoV projection is a square, the inscribed triangle calculation is trivial.

Considered model for the on-board sensors

Regarding the use of on-board sensors, the literature mostly refers to cameras [30]. In those cases, by knowing the length, width and focal length of the camera, as well as the altitude from the sea, the shape of the projection of the FoV of the camera can be calculated based on the attitude of the UAV. This is not the case for point or side scan beam sensors, which have a wide width, but a really narrow length scanning profile. As explained before, in this Thesis the term FoV is used in order to refer to the projection of the FoV of a generic camera on the sea. As such, the on-board sensor along with a gimbal, point downwards at a zero angle and parallel to the ground. Thus, pitch and roll angles of the UAV with respect to the horizontal plane are not relevant to the FoV projection, since the gimbal compensates for these angles, as in Fig. 2.6.

This is also true in other scenarios which involve communication sensors. Interesting oceanographic data is continuously collected and stored by buoys. Since a continuous operation of the communication devices of the buoys would waste considerable amounts of energy, a more conservative method is applied. Typically, data traffic in such scenarios is sparse and it is only needed for data acquisition or extraction. Hence, an interrogation method is applied where the communication circuitry is enabled either in intervals or operating in low energy mode until it is waken-up by a required signal. The latter should

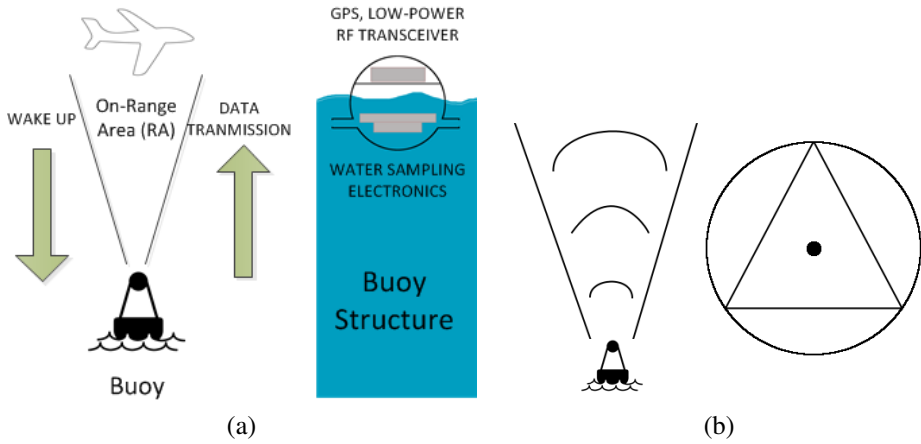


Figure 2.9 In (a), the data extraction structure scenario where in (b), the communication range cone area RA of the buoy on a side (left) and a top (right) view. The top view of this range at a specific altitude which makes the transmission feasible, creates a circle in which an inscribed triangle can be used as the maximum resolution size and constraint of a triangulated mesh. Then, each centroid of this collection of triangles, serves as a waypoint for the coverage task.

have an appropriate frequency, while the communication range of the wake-up sensor is limited. Figure 2.9a shows the on-range area RA to enable transmission that is determined by the transceiver specifications.

Since the location of the buoys is not always known due to sea drifts, the proposed decomposition method will perform the aforementioned CDT and each triangular cell will have the size of the expected buoy communication range cone, as can be seen in Fig. 2.9b.

Sample rate is another sensor characteristic to consider for the cell decomposition of a region. It is necessary to decompose the area in a manner that a sensor can obtain at least one sample of each of the resulting cells in a unit of time (see Figure 2.10). Then, the projected footprint area $P(t)$ must guarantee that its size is proportional to the sample rate T and the UAV speed V . This can be achieved by either reducing speed or increasing altitude in order to grow the projection of the FoV. For most of the sensors, the sample rate is not an issue; however, this aspect has been pointed out for the sake of generality.

2.4.3 Lloyd optimisation grid enhancement

The CDT decomposition strategy discussed in the previous sections, does not guarantee that all the segregation cells are going to be equilateral triangles, thus providing smaller triangles and larger sample overlapping. Its constraints are limited to either maximum

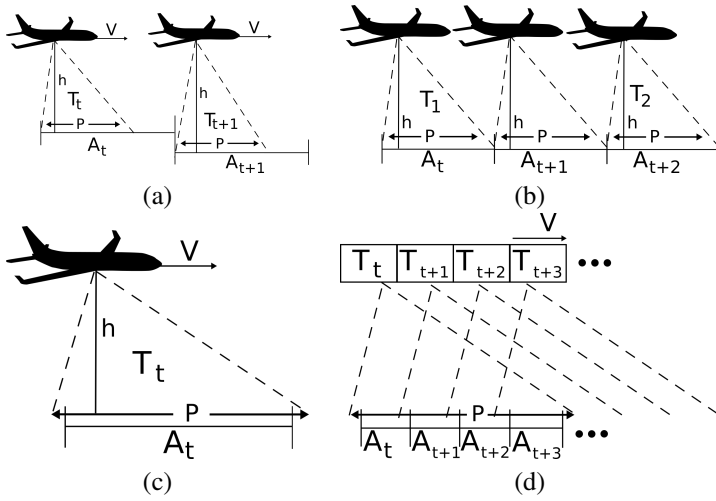


Figure 2.10 The appropriate cell decomposition is proportional to the velocity V , the sample rate T and the FoV projection footprint P . In (a), in every time step t_0, t_1, \dots, t_n , F is not large enough for the sensor to take a complete sample, whereas in (b), T is not fast enough to obtain a sample from each area. In both cases, the problem could be solved by either reducing the speed, while increasing the sample rate if possible or increasing the altitude to increase the projection of the FoV. (c) shows the ideal solution in the limit, whereas (d) depicts the most usual case of the same portion of the sea being present in many samples.

triangle side or minimum angle and the area itself. Hence the choice of optimising the resulting grid is merely a task and equipment defined choice: if a varying overlapping schema is not critical, then the initial grid can be maintained. On the other hand, if a normalised schema is preferred, a series of Lloyd optimization [28] iterations can be performed in the resulting grid. This algorithm improves the homogeneity of the mesh, by altering the angles of each cell, making them as close as possible to 60 degrees. The method is dependent on the iterations selected and since it moves all the points H of the $CDT(H)$ triangulation in each iteration, it is computationally expensive [31]. Therefore, it is recommended mainly for offline computation; more information on the actual complexity of the Lloyd optimization can be found in [32]. Nevertheless, having the size of each cell of the triangulation as well as the FoV of the sensor, the magnitude of sample overlapping can be calculated in the initial coverage planning. It is clear that the more regular the triangles are, the less overlapping will occur. Since by increasing the Lloyd iterations the distribution of triangles closer to equilateral is also increased, this operation can be added to the path generation cost functions. Finally, there is another positive effect of the Lloyd

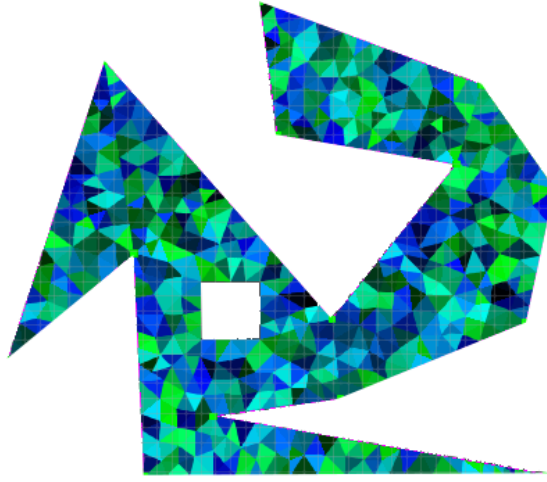


Figure 2.11 A CDT of a non-convex polygon with an internal hole. The simulated shape is a 5km x 5km area, scaled down by a 1/10 factor for visualization and notation purposes .

optimisation on the resulting grid, explained in detail in the following Chapters: since the overall angle is tending to reach 60° , the paths created by connecting every centroid of the triangular mesh have smoother angles, thus resulting in smoother turns; easier to be followed by non-holonomic, fixed wing UAVs.

2.5 Results

In every case, the CGAL implementation of the triangulation has managed to produce a mesh consisting of homogeneous triangles. The triangulation and meshing functions were tested by giving several input constrain criteria for the lesser angle bound constraint of the produced triangles, as described in Section 2.4.1. The proposed method has been tested on various scenarios in order to obtain qualitative and quantitative results, iterating through angle bound constraints for the produced cells. Resulting meshes have been examined, obtaining information about triangle homogeneity. Initially, a relatively simple non-convex polygon with an internal no fly zone was used, as shown in Fig. 2.11.

Produced vertices, total cells, minimum resulting edge and minimum resulting angle are presented in Table 2.1. In all trials, the input criteria for the edge size have remained the same; a predefined size for the *FoV*, as defined in Fig. 2.6, of 25 metres (250 metres in the actual area). It is noted that reaching the upper angle constraint limits of the Delaunay

Table 2.1 CDT performance based on lower angle bound constraint.

Angle bound (deg)	Vertices	Cells	Min edge length (m)	Min angle (deg)
0.011-8.88	304	476	10	25
9.16-10.02	305	478	10	25
12.31	313	488	10	29
15.29	321	500	9.43	31
16.15	325	508	8.24	32
17.01	357	568	6.32	32
18.16	377	597	6.32	34

Table 2.2 Square grid and Constrained Delaunay Triangulation decomposition methods comparison.

Footprint Size (m)	Total cells (square mesh)	Total cells (CDT)	Partial coverage by square mesh
15	331	1395	125
20	211	784	99
25	138	476	73
30	104	352	64

triangulation, an increased number of sub-optimal cells were produced near the no fly zone. Moreover, the algorithm could not proceed further than the angle bound constraint of 18.9 degrees, a size which is consistent to the one being referred to in CGAL notes. Nevertheless, the FoV criteria are always met, as in all produced triangles at least one side has the edge size input constraint.

Comparison with a square grid method

The same shape was decomposed following a simple square grid decomposition strategy as in the studies presented in [4]. A footprint size of 15, 20, 25 and 30 metres has been tested in order to observe the amount of cells that partially cover the region of interest by that method. Note that using the strategy presented herein, there are no cells that partially cover an area, and the centroids of the triangular cells produced are always inside the domain. Results in Fig. 2.12 show that even though this method manages to produce a lesser amount of cells, its performance is highly dependent on the shape of the area, the restricted zones, as well as the resolution of the FoV. Comparison of the two methods in the 25 metres sized footprint as presented in Table 2.2, shows that half of the produced cells by the square grid method share a region outside the domain of interest, raising questions on the restrictions that have to be applied in order to either visit or not those specific cells.

The same comparison has been performed in a more realistic scenario in an actual area,

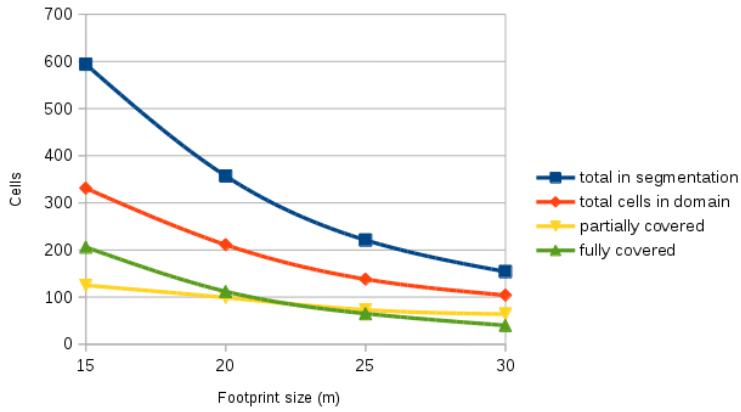


Figure 2.12 Grid decomposition performance of the area in Fig 2.11. In the 25 metres footprint case, out of the 138 total cells that cover the area, 73 cells are partially inside the domain of interest.

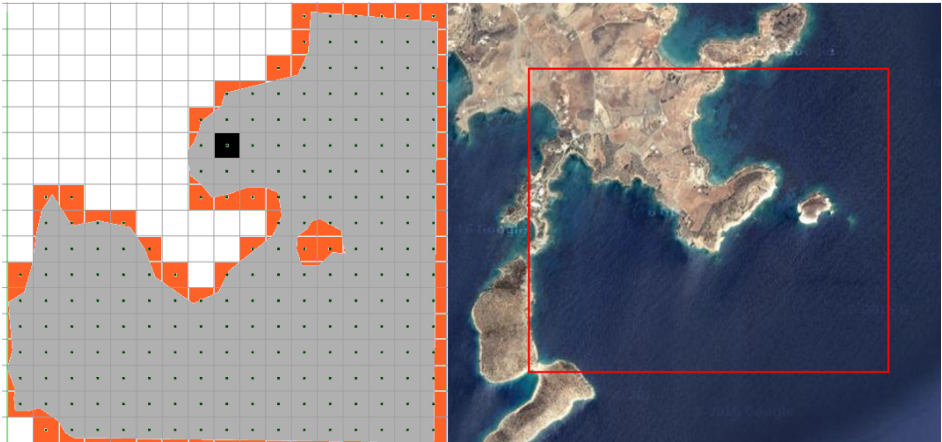


Figure 2.13 A coastal area in Schinoussa island on the right. On the left, the same area when a grid overlay is applied. Every orange shaded cell represents an ambiguous cell.

as shown in Fig. 2.13. In a total of 218 square cells, 78 had partially a no-fly zone in them. As we will see in the next Chapters, these ambiguous areas raise critical questions in path generation. These preliminary results demonstrate the feasibility of the method which provides an initial tool for exact area decomposition for the next stages of area partition and the extraction of waypoint plans.

2.6 Conclusions

The results presented here are a mere demonstration that the proposed method manages to treat the aforementioned complex areas while respecting the aerial and ground restrictions. By segmenting an area into cells that are consistent with the capabilities of a UAV, the segregation of the state space can manage to decrease the complexity for the following proposed methods of path planning and partition among several heterogeneous UAVs. In this architecture the decomposition schema serves the role of a two dimensional information cloud, where each cell can contain different attributes relevant for the mission. The produced mesh and associated graph created by treating each triangular cell as a node, manages to be consistent with the area properties, the capabilities of the UAVs and their on-board sensors. In the next Chapter, this procedure is applied for every UAV and the graph is used for partitioning the whole area with respect to the relative autonomy capabilities of a UAV team.

3 Area partitioning for multiple vehicles

Solidarity makes us greater than the sum of ourselves

UNKNOWN

In the context of multi-robot coverage of a previously known area, a fair division of this area has to be performed. This division is performed in order to properly separate the robots and assign a strictly defined region for each of them. The notion of fairness describes a partitioning strategy which takes into consideration the relative capabilities of the robots, like their energy autonomy in terms of flight time, or their sensing capabilities. This method also has to include some constraints which are met in real world situations, like the restrictions imposed when the initial positions of the robots are part of the input. Considering the way we have decomposed an area in the previous steps, the partitioning methods described in this Chapter take advantage of the triangulated schema as follows: each of the centroids of the FoV-sized, orientation agnostic triangular cells, as described in the previous Chapter, can be treated as a node in a graph. Then, in order to partition this graph for an arbitrary number of robots, we have to consider some geometrical, task defined and vehicle specific properties which are not evident in a graph form. Initially, the configuration spaces of each sub-area cannot be disjoint. Moreover, the initial positions of the robots, or the positions of greater interest in a task have to be taken into consideration. Finally, the vehicles' relative total flight duration autonomy capabilities, will define the percentage that each UAV can cover in an area. These properties define the metrics

described in this Chapter and a collection of algorithms which manage to satisfy those criteria is presented.

3.1 Related work

In the same context of area coverage and tracking tasks, several efforts have been made to extend the solutions for multiple vehicles. The increment of computational capabilities and simultaneous decrement of cost have allowed the use of multi-vehicle systems to distribute the computational burden and to allow the creation of distributed sensor networks. The fundamental work of Hert et al. in [33] manages to successfully decompose a complex polygon with holes in an arbitrary number of equal or unequal sized partitions. Nevertheless, especially in the unequal sized case, narrow passages can be identified and there is no control on the resulting shape. Moreover, the initial positions or regions of increased interest are not taken into consideration. Regarding the discrete configuration spaces and the partition process, using a Voronoi tessellation or a Delaunay triangulation is common; in [34] a Voronoi partitioning of an area to be painted is performed in order for the involved robotic arms not to overlap or collide. The process is optimized by minimizing the completion time for the task and the sum of torques of the arms in order to expand their lifespan. In a recent work presented in [35], the authors provide an algorithm for fair area division and partition for a team of robots, with respect to their initial positions. Their solution produces promising results regarding the algorithm's computational complexity and guarantees full area coverage without backtracking paths. While their approach is feasible and manages to successfully tackle the problems of fair partition and coverage path planning, the strategy accounts only for the fair division case and fixed cell sizes, while in some cases, the sub-areas produced do not have a uniform geometric distribution. Acevedo et al. in [36, 37, 38] have introduced a decentralized algorithm for surveillance tasks which manages to propagate all task related information between robots, considering the communication constraints between them. Kassir et al. in [39] also takes into account the communication resources that each of the vehicles has in order to increase the information gain of the whole system. In [40], the authors tackle the problem of complete coverage by trying to minimize the completion time for the robots. In their strategy, turns imply the decrease in speed and eventually the acceleration of the vehicles. In that manner, their algorithm tries also to minimize the number of turns. This work also uses a grid-like decomposition strategy, based on disks. The vehicles are considered to have equal capabilities while their initial positions are optimized for the optimization problem. Once again, the areas considered for the experimental setups are

convex rectilinear polygons. The work presented in [41] treats the problem in a similar way as this Thesis. Nevertheless the authors present an equally partitioned area, whereas the growing region algorithm for creating this partition is problematic, since it can lead to disjoint areas for the robots. Finally, the authors in [42] construct spanning trees for all the robots for exploration, without treating collisions or repeated coverage. Moreover, the considered area for the experiments is somehow simplistic.

In the framework proposed in this Thesis, a partitioning method is presented which manages to partition any complex area for an arbitrary number of robots, by satisfying a number of criteria:

- The algorithms must account for any complex, non-convex area with holes, as described in the previous Chapter.
- The solution is complete for any number of aerial vehicles.
- The resulting sub-areas cannot be disjoint.
- The partitioning has to be performed in such a manner that each sub-area size is a given percentage of the whole area.
- The initial positions of the vehicles, since the probability of obtaining information in a tracking task for a given target, decreases uniformly from that starting location.
- The algorithm can be implemented in a centralised or decentralised manner.

3.2 Partitioning

Following the example of Section 2.4, consider the same buoy searching test case, where from the initial belief of the buoys location there is a uniformly location probability in every direction, with the same notation:

$$U_1, U_2, \dots, U_n \quad (3.1)$$

is a team of n UAVs with initial locations

$$\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n \quad (3.2)$$

Please note that since we consider a team of heterogeneous UAVs, each of them may be in charge of a different coverage percentage of the whole area R , the *capability coefficient*

C_j . This coefficient is expressed as a percentage, showing the relative capability with respect to the other vehicles of the team. As such, Z_i is the surface in square meters to be covered by U_i . Region R is decomposed into a set:

$$\mathbf{S} = \{ v_j \mid j \in \mathbb{N}, 1 \geq j \text{ and } j \leq m \} \quad (3.3)$$

of triangular cells, as explained in the previous chapter, without any cells being outside or partially inside R . Then, the goal is to compute the geometry of a set of sub-areas $R = R_i, i = 1, \dots, n$ based on the cells v_j taking into account the following metrics:

- The closeness of the cells within R_i to the initial location \mathbf{p}_i of the UAV in charge of searching that sub-area should be minimized. This can be achieved by minimizing the sum of distances between each center of cell \mathbf{v}_{ij} from the set \mathbf{S} and the initial locations \mathbf{p}_i :

$$\min_S F(S) = \min_S \sum_{i=1}^n \sum_{j=1}^{m_i(S)} \|\mathbf{p}_i - \mathbf{v}_{ij}\|, \quad (3.4)$$

where $m_i(S)$ is the number of cells of S inside R_i .

- The size of R_i should be as close as possible to Z_i for all of the UAVs. This can be achieved by minimizing the sum of differences:

$$\min_S G(S) = \min_S \sum_{i=1}^n \left| \sum_{j=1}^{m_i(S)} \text{area}(v_j) - Z_i \right|, \quad (3.5)$$

where $\text{area}(v_j)$ represents the area of the cells inside R_i .

The former metric that takes into consideration the probability of localization motivated the design of the following algorithms where each sub-area is generated by a uniform growth from the starting locations $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$. This process has another positive side effect: since this growing region process is performed in every direction, it creates “symmetric” areas that are suitable to be covered by energy-efficient spiral-like patterns [43]. In addition, R_i by construction cannot be disjointed or intersected by another sub-area, neither by a no-fly zone. This restriction guarantees that the resulting sub-areas prevent the existence of overlapping coverage paths or collisions. In case additional safety requirements were present, it would be possible to define different flight altitudes for the UAVs in adjacent sub-areas. Figure 3.1 shows an example of a region R partitioned among three UAVs.

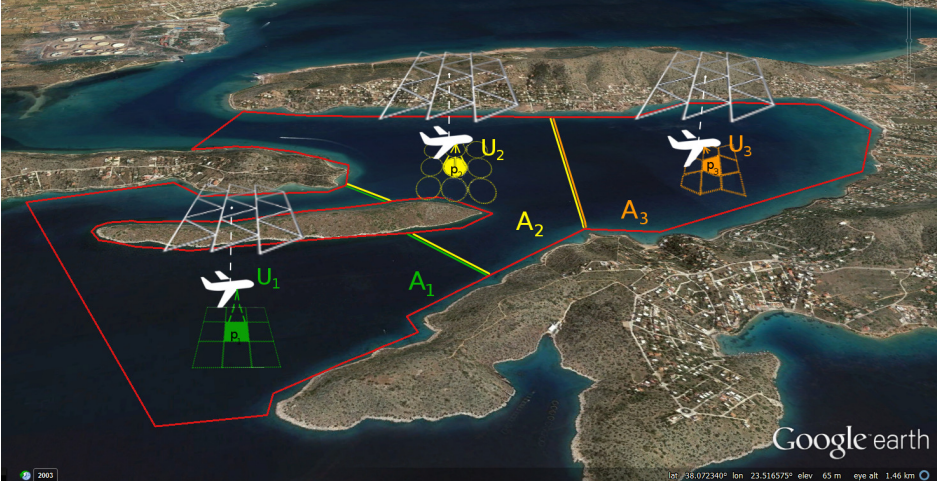


Figure 3.1 An example with three UAVs, each one with its allocated sub-area. The scheme is composed by two levels: the bottom layer shows the different on-board sensors' field of view projection on the sea, whereas the upper shows the cell decomposition denoted as a triangular grid on top of each UAV. U_1 , U_2 and U_3 denote the UAVs, and A_1 , A_2 and A_3 denote the sub-areas of the total region R , which is constrained by the red borders. The initial positions of the UAVs are \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 .

3.2.1 Baseline Area Partitioning Algorithm

Let us consider an undirected graph $G = (V, E)$, where the set V of vertices represents the triangular cells of the CDT and E is the set of edges such that there is an edge from v_i to v_j if the corresponding triangles are neighbors. Two triangular cells are neighbors if a UAV can move freely between them. This graph is also intended to be used in order to compute roadmaps for coverage path planning after the area partition is obtained. It should be mentioned that the CDT is computed based on the largest FoV among the available UAVs. Later, once the sub-areas R_i are computed, another CDT is performed inside to fit the particular FoV of each UAV.

By treating the CDT as a graph, a baseline area partition algorithm can be designed based on two attributes for each vertex v_i of the graph: $C(v_i)$ as a unitary transition cost; and $A(v_i)$ is the identifier of the UAV that will visit v_i . These attributes are computed as an isotropic cost attribution function by a step transition algorithm, starting from the initial position of each UAV, propagating towards the other UAVs or the borders of the area. Due to the fact that this algorithm expands in waves from each of the UAV and since each vehicle cannot overtake triangular cells of another vehicle and it progresses

Algorithm 1: Antagonizing wavefront propagation algorithm that computes the baseline area partition. Q is a queue list managed as a FIFO by functions *insert* and *getFirst*

```

1  $v_{Ik}$ : initial vertices/triangular cells for each UAV  $U_k$ ,
2  $S_v$ : area size of triangular cells  $v$ ,
3  $N(v)$ : the set of neighbours of vertex  $v$ ,
4  $A(v)$ : the UAV identifier allocated to triangular cell  $v$ ,
5  $Z_k$ : area coverage capability of UAV  $U_k$  in square meters,
6  $S_{vMin}$ : area size of the smallest triangular cell in CDT,
7  $C(v_i)$ : is the unitary transition cost of vertex  $v_i$ ,
8  $C_p$ : is a unitary transition cost place-holder
9  $C_p \leftarrow 1$ 
10 foreach  $U_k \in CDT$  do
11    $Q.insert(v_{Ik})$  and mark  $v_{Ik}$  as visited;
12    $Z_k \leftarrow Z_k - S_{v_{Ik}}$ ;
13    $C(v_{Ik}) \leftarrow C_p$ ;
14 end
15 while  $Q$  not empty do
16    $v \leftarrow Q.getFirst()$ ;
17   if  $C(v) \neq C_p$  then
18      $C_p \leftarrow C_p + 1$ ;
19   end
20   foreach  $v_i \in N(v)$  do
21      $k \leftarrow A(v)$ ;
22     if  $v_i$  not visited AND  $Z_k > S_{vMin}$  then
23        $C(v_i) \leftarrow C_p + 1$ ;
24        $A(v_i) \leftarrow A(v)$ ;
25       mark  $v_i$  as visited;
26        $Q.insert(v_i)$ ;
27        $Z_k \leftarrow Z_k - S_{v_i}$ ;
28     end
29   end
30 end

```

in a breadth-first manner [3], the strategy is called Antagonizing Wavefront Propagation (AWP).

This strategy is presented in Algorithm 1 and works as follows. Let us consider each of the initial positions of the UAVs as the root node of a tree; each root node is given an initial step cost of one. In every recursion step, each vertex that has an edge connected to the parent vertex is given that cost plus one. In addition, vertex v_i gets the same $A(v_j)$ attribute of its parent vertex v_j , propagating the identifier of the UAV in that way. In case

the number of vertices for the U_k UAV meets its autonomy limit, denoted by the total area Z_k it should cover, the algorithm for that UAV stops. Please note that these steps are not performed if a triangular cell already has any of these attributes.

Since queue Q is accessed once for every i -th cell, the *while* iteration has a complexity of $\mathcal{O}(m)$, where m is the number of vertices. The complexity of getting the first element is $\mathcal{O}(1)$; then, it inserts new elements according to the restrictions. The insertion in a stack has a complexity of $\mathcal{O}(1)$. Hence, the complexity of Algorithm 1 is $\mathcal{O}(m)$. The area partition computed is not sufficient for complex cases where deadlock situation occur after applying Algorithm 1. A further adjustment step is needed in order to assign regions where a deadlock has happened, as will be described in Section 3.3.

3.2.2 Reverse Watershed Schema

By performing the previous step, each configuration space is either adjacent to another configuration space or to the borders of the whole region. In that manner, a second algorithm (see Algorithm 2) assigns to each vertex that already has an UAV identifier a unitary border-to-center cost attribute $D(v_i)$ of proximity from the borders to the center of the configuration space. The triangular cells that are adjacent to a border with another configuration space or to the whole area are given a high $D(v_i)$ cost and are considered as the root nodes of a tree. In each step of the algorithm, this cost is decreased and propagated to the adjacent triangular cells of these nodes. This function manages to create a border-to-center pattern resembling a watershed algorithm, and as such it is called the Reverse Watershed Schema (RWS) algorithm.

Please note that this cost attribution will further facilitate the coverage plans to be produced, as will be described in the following Chapter. Finally, we have to note that the complexity of this algorithm is similar to the previous one. The first loop has a complexity of $\mathcal{O}(m)$, $\mathcal{O}(m)$ for the initial *foreach* loop and $\mathcal{O}(1)$ for each insertion to the stack, since the second inner *foreach* has a maximum of three iterations. For the same reasons, the second *while* loop has also a $\mathcal{O}(m)$ complexity.

3.3 Deadlock handling

The previous baseline partitioning algorithm is able to perform well in most cases where the area is simple or where the initial positions of the UAVs are evenly distributed in the area. Nevertheless, it may lead to several deadlock scenarios as the growing sub-areas meet each other, as can be seen in the example shown in Figure 3.2. Hence, a Deadlock Handling (DLH) algorithm that adjusts the initial partitioning in the non-allocated areas

Algorithm 2: RWS algorithm for the generation of the border-to-center cost $D(v_i)$ attribute. Q is a queue list managed as an FIFO by functions *insert* and *getFirst*.

```

1  $N(v)$ : the set of neighbors of vertex/triangular cell  $v$ ,
2  $A(v)$ : the UAV identifier for triangular cell  $v$ 
3 foreach  $v \in CDT$  do
4   foreach  $v_i \in N(v)$  do
5     if  $A(v) \neq A(v_i)$  then
6       mark  $v$  as visited;
7        $D(v) \leftarrow \infty$ ;
8        $Q.insert(v)$ ;
9     end
10  end
11 end
12 while  $Q$  not empty do
13    $v \leftarrow Q.getFirst()$ ;
14   foreach  $v_i \in N(v)$  do
15     if  $v_i$  not visited then
16        $D(v_i) \leftarrow D(v) - 1$ ;
17       mark  $v_i$  as visited;
18        $Q.insert(v_i)$ ;
19     end
20   end
21 end

```

is needed, by exchanging UAV identifiers or assigning UAVs to the empty areas. Two different approaches have been tested, by applying the two algorithms presented before.

As was stated before, each UAV U_k should cover an area of Z_k . In a test case, after the initial partitioning, let us consider that $Y_k \neq Z_k$ space has been allocated to U_k . In the deadlock scenarios (see Figure 3.2), there are areas that do not belong to any UAV. These areas are allocated to a virtual UAV U_{-1} with area $Z_{-1} = 0$. Let us consider a list L_U that contains the results of $Z_k - Y_k$ for each UAV and the area size of the smallest triangle in the CDT S_{vMin} . Thus, each UAV can have an area surplus if $Z_k - Y_k > S_{vMin}$ or a shortfall if $Z_k - Y_k < S_{vMin}$. The latter case always happens in the deadlock scenarios for U_{-1} . In each recursion of Algorithm 3, a pair of UAVs, for instance U_i and U_j , one having an area surplus and another with a shortfall, is chosen from the list in order to gradually exchange triangular cells between them to reach the desired area size, one area at a time. In order to do so, a feasible transition sequence P_{ij} must be found, as shown in Figure 3.3.

The complexity of this algorithm is calculated as $\mathcal{O}(nm^2)$ due to the *findSequence* function, which is actually a tree sort; n is the number of UAVs and m the number of

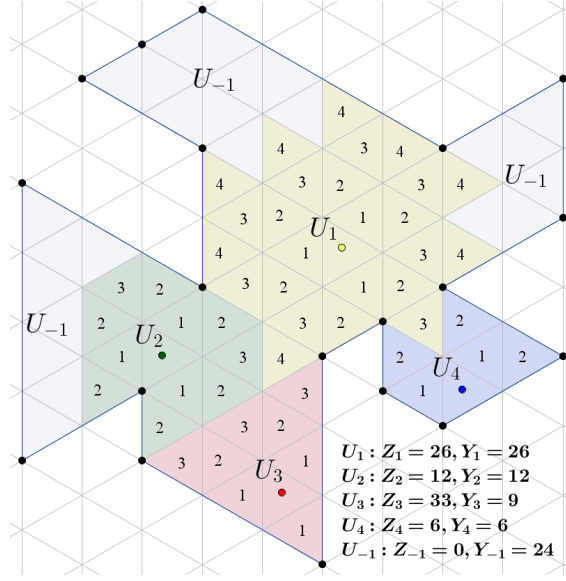


Figure 3.2 A deadlock scenario. Four UAVs U_1 , U_2 , U_3 and U_4 after the baseline partition Algorithm 1. UAVs U_1 , U_2 and U_4 have met their autonomy capability of Z_k by covering Y_k area. Nevertheless, U_3 was not able to overtake any more area, being “blocked” by the other UAVs and the borders of the whole region. Different colors/shades of areas indicate the configuration space of each UAV, while the numbers inside the cells indicate the isotropic cost, as has been assigned by Algorithm 1. The free or non-allocated areas belong to virtual UAV U_{-1} .

cells. The complexity of the *move* functions is displayed below, in each of the following transposition algorithms.

Two algorithms called *moveAWP* and *moveRWS* have been implemented for the move function, which is used in Algorithm 3. In the former, the farthest vertex in the R_i sub-area of U_i UAV in sequence P_{ij} is chosen, by using the unitary transition cost information from Algorithm 1. This vertex has also to be adjacent to the second area in the transition sequence P_{ij} . Starting from that vertex, Algorithm 1 is applied again, overtaking the requested area size in the means of exchanging UAV identifiers between those triangular cells. Recursively, this operation is performed for every item of the sequence (see Algorithm 4). The complexity is $\mathcal{O}(k)$, where k is the number of areas that are in the transition sequence. Then, since Algorithm 1 is used for the transposition function, the whole complexity is $\mathcal{O}(l * m^2)$, where l is the number of cells to be transposed and w is the *findSequence* algorithm complexity.

In the second approach, we apply the RWS algorithm in order to get a depth schema of

Algorithm 3: Multi-UAV partitioning Deadlock Handling (DLH) algorithm. Base-line partitioning is performed by Algorithm 1, whereas this method is for the sub-area size adjustment (if needed). Function *getSurplusUAV*(L) gets a UAV identifier from list L that has an area surplus, whereas function *getShortfallUAV*(L) gets the identifier of a UAV that has an area shortfall after Algorithm 1. Function *findSequence* finds a feasible transition sequence P_{ij} between UAV U_i and U_j , whereas the *move* function performs the transfer between triangular cells

```

1  $S_{vMin}$ : area size of the smallest triangular cells in CDT
2 while  $\exists U \in L_U < S_{vMin}$  do
3    $i \leftarrow \text{getSurplusUAV}(L)$ ;
4    $j \leftarrow \text{getShortfallUAV}(L)$ ;
5    $P_{ij} = \text{findSequence}(U_i, U_j)$ ;
6   if  $Z_i - Y_i > Z_j - Y_j$  then
7      $\text{move}(Z_j - Y_j, P_{ij})$ ;
8   end
9   else
10     $\text{move}(Z_i - Y_i, P_{ij})$ ;
11  end
12 end

```

Algorithm 4: MoveAWP algorithm. $C(v)$ is the transition cost from the AWP algorithm (see Algorithm 1). Then, function *findBiggestC*(v)(U_i, P_{ij}) finds the triangular cell with the largest transition cost value of UAV U_i that is adjacent to UAV U_{i+1} in the sequence. Then, function *Awp* takes as variables an initial cell v , the area size that needs to be exchanged and the UAV identifier that needs to be exchanged from. The growing function is similar to Algorithm 1

```

1  $P_{ij}$ : the transition sequence between  $U_i$  and  $U_j$  for triangular cells exchange, treated as a list
2  $v_{init}$ : initial triangular cell for identifier exchange
3  $S$ : area size to be moved
4 foreach  $U_i \in P_{ij}$  do
5    $v_{init} = \text{findBiggestC}(v)(U_i, P_{ij})$ ;
6    $\text{Awp}(v_{init}, S, U_i)$ ;
7 end

```

the adjacent areas, as was described in Section 3.2.2. In each recursion of the algorithm (see Algorithm 5), the amount of triangular cells that are in the borders of the first pair of the transition sequence exchange their UAV identifiers. If the area of these border triangular cells sum up less than the requested area, then the area size and the total amount of border triangular cells exchange their UAV identifiers. If not, then only the triangular cells in the

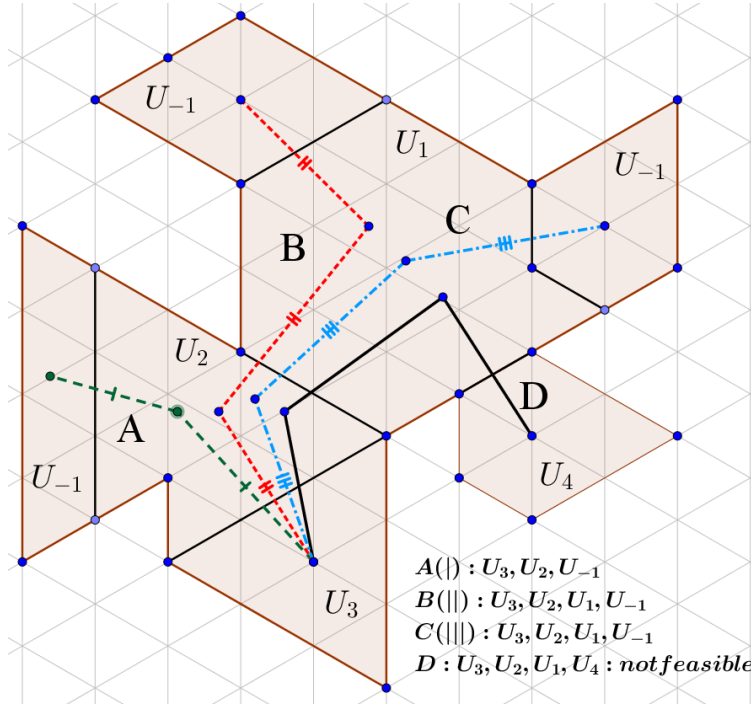


Figure 3.3 Transition sequence selection. After the initial partition process, U_1 , U_2 and U_4 have met their sub-area size constraint and blocked the growth of U_3 . As a result, three areas are not allocated (U_{-1}). The feasible transition sequences $A(\text{green}(|))$, $B(\text{red}(||))$ and $C(\text{blue}(|||))$ are used in order for U_3 to obtain the requested total area, by gradually exchanging cells in every pair of the sequence. Sequence $D(\text{black})$ does not lead to a partition that has an area shortfall, and thus, it is a not feasible sequence.

front (in the borders) exchange their UAV identifiers. This amount of triangular cells is then exchanged to the next UAV in the sequence and so on, maintaining the aforementioned restriction, until all of the requested area and associated triangular cells are transposed from the initial UAV in the sequence to the last.

This algorithm's complexity is $\mathcal{O}(nw^2)$ due to the use of the *findSequence* function, as has been described in Algorithm 3.

There are two main differences in these approaches. In the first approach, we have a wavefront pattern from a single triangular cell, whereas in the second, the exchange progresses as a kind of width sweep Morse function [4]. The second difference is that in the first approach, all of the triangular cells to be exchanged are the transposed UAV first, and in the second approach, only the amount of triangular cells that are in the adjacent

Algorithm 5: MoveRWS algorithm. Function *findSequence* finds a valid transition sequence, as can be seen in Figure 3.3. This function is also called before the initial recursion of the MoveRWS algorithm. Function *exchangeIdentifiers* makes use of the information of the RWS algorithm (see Algorithm 2), and it exchanges robot identifiers on two adjacent configuration spaces, by exchanging the amount of triangular cells that have the lowest coverage cost, but are adjacent. It also propagates and extends this cost. Function *restOfSequence* returns the remaining sequence for the specific $P[i] \rightarrow P[i+1]$ transition, in order to initially transfer only the amount of triangular cells that are adjacent between i and $i+1$ until the final U_j UAV. In case this happens, the requested area has not been exchanged yet, so the algorithm runs recursively, and the last line takes a step back in sequence traversal

```

1  $S$ : area size to be moved
2  $S_{adj(kl)}$ : the area size of adjacent triangular cells between UAV  $k$  and  $l$ 
3  $P_{ij}$  the transition sequence between  $U_i$  and  $U_j$  for triangular cell exchange, treated
   as a list
4 foreach  $U_i \in P_{ij}$  do
5    $P_{ij} \leftarrow \text{findSequence}(U_i, U_j)$ ;
6   if  $S_{adj(P[i], P[i+1])} > S$  then
7      $\text{exchangeIdentifiers}(P[i], P[i+1], S)$ ;
8   end
9   else
10     $P_{rest} = \text{restOfSequence}(U_i)$ ;
11     $\text{MoveRWS}(S_{adj(P[i], P[i+1])}, P_{rest})$ ;
12     $S = S - S_{adj(P[i], P[i+1])}$ ;
13     $i = i - 1$ ;
14  end
15 end

```

borders are transposed in each step. In that manner, the triangular cells of the area are propagated respecting the total amount of cells that each UAV has each time, resolving overlapping UAV issues, as will be discussed in Section 3.4.

3.4 Results

Two coastal areas in Greece have been selected for the experiments (Figure 3.4). The first is a broad and populated shore near the harbor of Piraeus, Salamina, the same as presented in Figures 2.2 and 2.5. The second is a remote island in the Aegean archipelago, Astipalea. The first area was used for evaluating and comparing the partitioning algorithms, whereas the second for evaluating the proposed strategy in various setups.

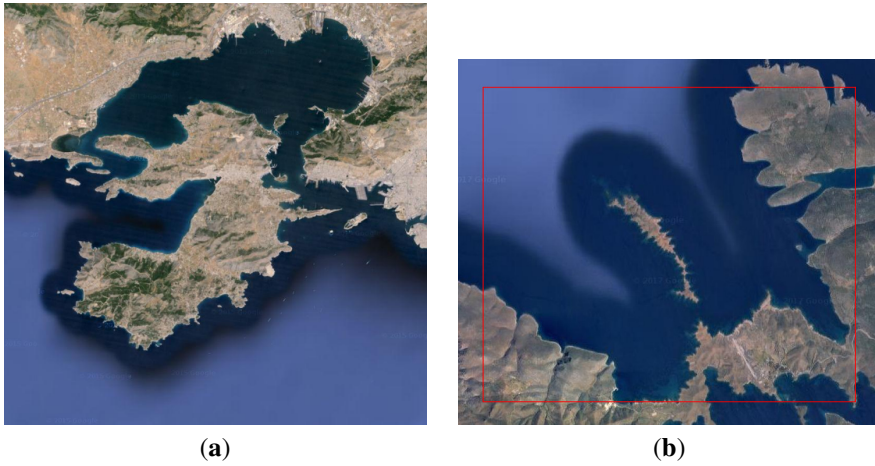


Figure 3.4 Selected areas for testing: (a) Salamina area having narrow passages and complex shapes in shores; (b) the Astipalea area is used for testing the suitability of the proposed algorithm.

In the first area, two square shaped FoV constraints with side sizes of 100 meters and 900 meters have been used, as has been defined in Section 2.4.1, in order to compare the behavior of the algorithms. Three test cases of different relative UAV capabilities have been tested: one has an equal relative percentage for the UAVs of 33,3%, another has a distribution of 10%,60% and 30% whereas the last test case has a distribution of 80%, 10% and 10% for the respective UAS (see Fig. 3.5). The partitioning strategies called MoveAWP and MoveRWS described in Section 3.3 have been compared. The former uses the transition cost of the AWP algorithm, and the latter applies the RWS algorithm for adjusting the baseline partition computed by Algorithm 1.

The complexity of the area has managed to highlight some issues that were not evident for the majority of simple areas. The main problem occurs during cell exchange when the initial position of the UAV is close to the borders, because a sub-area could overtake the initial position of the UAV (see Figure 3.5a).

Additional simulations have been performed to measure the performance of the different algorithms with respect to the metrics F and G explained in Section 3.2. In particular, the simulation environment shown in the second area of Figure 3.4 has been used for the metric F . Some results are detailed in Figure 3.6 for three and five UAVs and a FoV size of 30 m. In general, simulations have been executed for three and five UAVs, with initial locations evenly or randomly distributed in the area and different FoV sizes. The results for the sum of distances between each center of the triangular cell inside a sub-area and

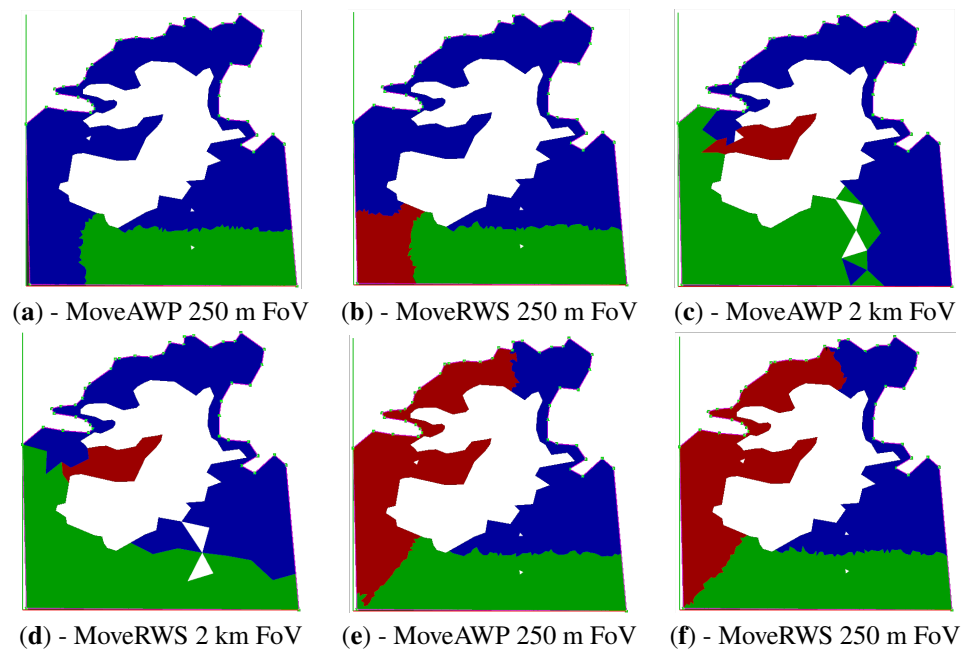


Figure 3.5 Partitioned area for three UAVs (indicated by the white cells) and visualized by using the ROS rviz node. Each four cases represent the results for different relative capabilities: cases *a-d* is the 10% (red), 60% (blue), 30% (green) case; cases *e-h* depict the 33% (red), 33% (blue), 34% (green) case; whereas the cases show the 80% (red), 10% (blue), 10% (green) case. Each consecutive pair of images indicate the comparison of the two algorithms. Images (a,b) show how the MoveAWP and MoveRWS algorithms have performed with the small (250 m) FoV, whereas (c,d) show the results for the large (2 km) FoV case.

Table 3.1 An even distribution of initial locations for three and five UAVs, with different relative capabilities.

	#UAVs	FoV (15 m)		FoV (30 m)	
		moveRWS	moveAWP	moveRWS	moveAWP
Metric <i>F</i> (m)	3	333.861,84	333.909,67	82.768,44	84.979,76
Metric <i>F</i> (m)	5	437.988,74	439.642,85	129.879,24	131.516,96

the initial location of the UAV inside that sub-area (metric *F*) are shown in Tables 3.1 and 3.2. In both cases, it can be seen that the moveRWS algorithm has a better performance than moveAWP, since the metric is lower.

Regarding the other metric *G* considered in Section 3.2, simulations have been performed

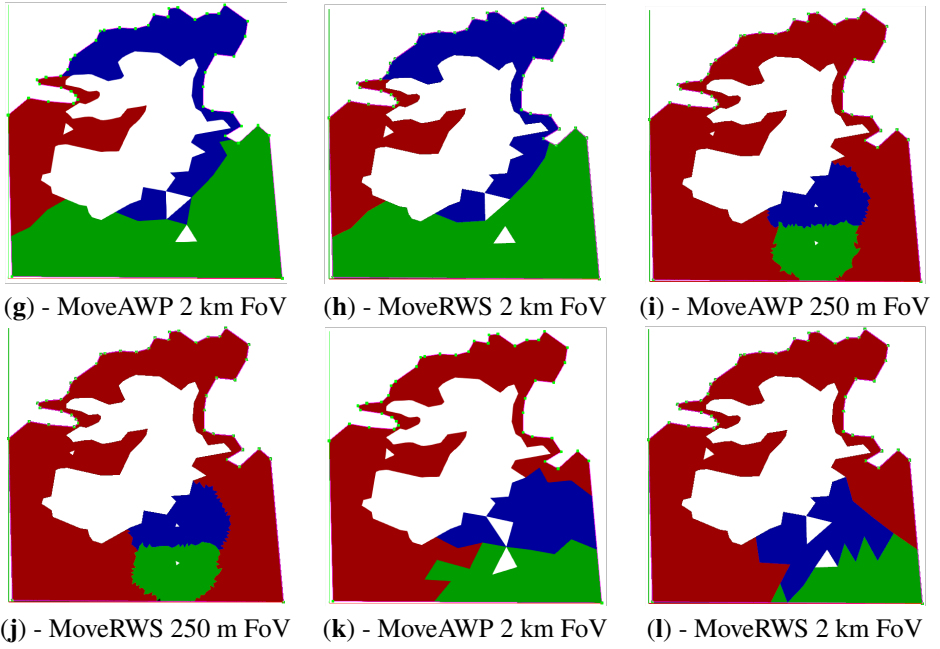
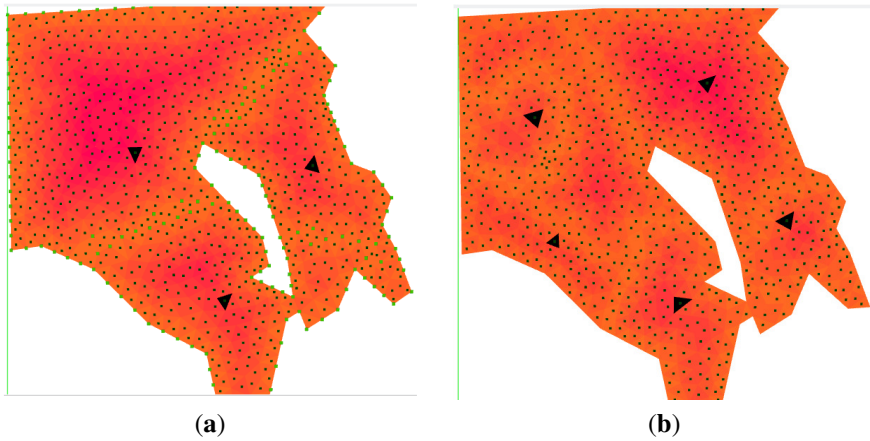
Figure 3.5 *Cont.*

Figure 3.6 Area of Figure 3.4b selected for the comparison of the two partitioning algorithms. (a) is partitioned for three UAVs, whereas (b) for five UAVs. The depicted FoV size is 30 m in both cases. Both figures are computed with the deadlock moveRWS handling of Algorithm 5. Lighter areas show the borders of each sub-area.

Table 3.2 Random initial position distribution for three and five UAVs, with different relative capabilities. Like before, Algorithm 5 has performed better than Algorithm 4.

	#UAVs	FoV (15 m)		FoV (30 m)	
		moveRWS	moveAWP	moveRWS	moveAWP
Metric F (m)	3	508.801,74	508.751,513	211.395	214.945,82
Metric F (m)	5	566.971,55	568.819,45	151.389,621	155.269,99

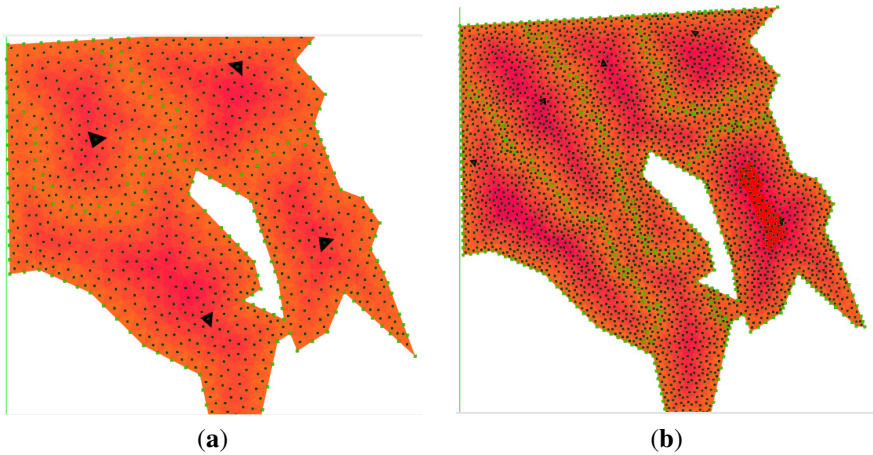


Figure 3.7 Area partition after applying the baseline and the deadlock moveRWS handling algorithm. (a) shows the area partitioned for four UAVs evenly distributed in the area and a FoV projection of 30 m. (b) shows the results for a FoV projection of 15 m and four UAVs randomly located. The black triangles depict the initial positions in all of the cases, while lighter areas and green dots represent the sub-region borders.

also in the second scenario of Figure 3.4 with three and six UAVs with different relative capabilities and initial locations evenly and randomly distributed (see Tables 3.4 and 3.5, respectively). The goal is to compare the results computed with the baseline algorithm and the improvement achieved with the moveRWS deadlock handling algorithm, which had the better performance in the previous scenarios. Figure 3.7 shows the results for two particular setups with four UAVs.

Moreover, in order to obtain a simple metric for the computational cost of the method, the first six cases (a to f) of Figure 3.5 have been selected to be timed . The results shown in Table 3.3 indicate that the Lloyd iteration computation in the whole procedure is decisive as to categorize this framework as a real-time solution. As the size of the cells

Table 3.3 Computational time in seconds for the small (100 meters) and large (900 meters) FoV projection test cases of each part of the partitioning procedure of the simulations (a-f) shown in Fig. 3.5. The Lloyd optimization has been applied in the initial mesh as well as in every sub-area, after the partitioning process. Notice that the remaining time of the total calculation belongs to various visualization processes which are not part of the proposed framework.

Test case Lloyd iterations	1st		2nd		3rd	
	20	40	20	40	20	40
Small FoV (100 meters)						
Lloyd optimization (s)	2,888	5,288	2,968	5,152	2,72	5,217
Partition (s)	0,52	0,587	0,71	0,68	0,49	0,513
Total time (s)	3,71	6,99	4,036	7,19	3,3	6,04
Large FoV (900 meters)						
Lloyd optimization (s)	0,516	0,712	0,472	0,76	0,488	0,88
Partition (s)	0,2	0,23	0,212	0,22	0,179	0,2
Total time (s)	0,953	1,245	0,93	1,364	0,925	1,42

becomes smaller (smaller FoV), producing larger number of vertices, the time of the Lloyd optimization process increases. Hence, the number of Lloyd iterations can be adjusted considering the time to calculate a new waypoint plan versus the importance of having a more optimized mesh. By decreasing the Lloyd optimization iterations, a lesser amount of equilateral cells occurs, resulting in larger oversampling by the on-board sensor; the number of waypoints remains the same. Then, depending on a temporal constraint, e.g. the time a UAV during flight needs to pass from one waypoint to another, a constraint can be formulated regarding the computational time limit. Also note that the size of the cells are relevant to the whole procedure as they define the distances between the waypoints. Finally, the dynamics of the platform used, like its speed, are crucial to the optimisation decision and the number of iterations. Nevertheless, the rest of the proposed strategies and the partitioning algorithms manage to provide a solution relatively fast in relation with the number of vertex increase. Please note that the operations have been executed in a single system (Intel Core i5-5200U@2.20GHz CPU, 8GB of RAM, kUbuntu 14.04) and the software implementation included the CGAL library version 4.8.1 [44].

Finally, different numbers of Lloyd iterations have also been tested on the resulting mesh for all the cases, ranging between 20 and 60 iterations. In the simulations, different numbers of UAVs with even and random distributions for the initial locations (see Tables 3.4 and 3.5 respectively), different relative capabilities and FoV projections have been used. The results show the suitability of the proposed solution, as the average difference from the targeted relative capability of the UAVs has not exceeded a value of 1% on average for the

even distribution and 1,33% for the random distribution of the initial locations; Figure 3.8 shows this comparison of the average difference in the even and random distribution scenarios. Moreover, the algorithm manages to properly overcome deadlock scenarios as expected, as can be seen in the various setups of Table 3.5, where the initial baseline algorithm has up to 30% difference from the targeted relative capabilities of the UAVs.

Table 3.4 In each pair of values, the left one indicates the difference between the initial capability of a UAV (on the left column) and the actual assigned area after the initial baseline algorithm. The right value indicates the same difference after the moveRWS deadlock treatment algorithm. An average for all UAV's, as well as the total difference is shown below each experimental setup, where G is the metric defined in Equation 3.4 and $\text{area}(R)$ is the area in m^2 of the whole region R . The UAV's have evenly distributed initial positions. Setups for 3, 4, 5 and 6 UAV's have been tested, with different relative capabilities and FoV values. Different Lloyd iterations on the mesh have been tested, ranging between 20 and 60.

UAV Capability %	FoV (15 m)			FoV (30 m)		
	Lloyd Iterations			Lloyd Iterations		
	20	30	60	20	30	60
50%	0,5/0,92	0/0,62	0/0,92	0,35/0,3	0,35/0,18	0,05/0,3
30%	6,74/0,14	6,74/0,89	6,88/1,21	6,95/0,21	7,19/0,65	6,86/0,25
20%	0,01/1,04	0,01/0,26	0,01/0,28	0/0,49	0,03/0,82	0,05/0,54
Average%	2,41/0,69	2,25/0,59	2,29/0,80	2,43/0,32	2,52/0,55	2,32/0,36
$G/\text{area}(R)\%$	7,25/2,1	6,75/1,78	6,89/2,42	7,3/0,99	7,57/1,66	6,96/1,09
20%	0,02/0,02	0,02/0,08	0,02/0,06	0/0,41	0,05/0,12	0,05/0,45
40%	0/0,06	0/0,11	0/0,05	3,89/0,28	3,91/0,59	3,6/0,49
20%	0,96/0	1,11/0	0,78/0,06	0,48/0,17	0,48/0	0,62/0,12
20%	0/0,02	0/0,17	0/0,03	0/0,29	0/0,48	0/0,07
Average%	0,25/0,03	0,28/0,09	0,2/0,05	1,09/0,37	1,11/0,3	1,07/0,28
$G/\text{area}(R)\%$	0,98/0,12	1,13/0,35	0,8/0,2	4,37/1,11	4,44/1,18	4,27/1,13

Table 3.4 *Cont.*

UAV Capability %	FoV (15 m)			FoV (30 m)		
	Lloyd Iterations			Lloyd Iterations		
	20	30	60	20	30	60
20%	0,15/0,38	0,15/0,6	0,18/0,01	0,15/0,11	0,11/0,59	0,11/1,36
30%	13,73/0,34	13,75/0,31	13,72/0,15	13,1/0,24	12,98/0,76	12,63/1,64
20%	0/0,31	0/0,42	0/0,08	0/1,6	0,3/0,81	0/0,66
10%	0/0,41	0/0,86	0/0,41	0/0,71	0/0,3	0/1,21
20%	0,21/0,65	0,05/0,14	0,21/0,36	0/1,01	0/0,36	0/1,61
Average %	2,81/0,42	2,79/0,47	2,28/0,35	2,65/0,92	2,68/0,56	2,55/0,9
G/area(R) %	14,09/2,1	13,95/2,33	14,11/1,01	13,25/3,69	13,39/2,81	12,74/4,5
10%	0,15/0,21	0,15/0,73	0,15/0,41	0,11/0,72	0/0,18	0,11/0,29
20%	0,3/1,04	0,3/1,37	0,3/1,35	0/1,83	0/1,37	0/0,43
10%	0,15/0,23	0,15/0,16	0,15/0,23	0,11/0,14	0/0,97	0,11/0,39
30%	11,26/0,38	11,45/0,69	11,2/0,53	11,94/0,84	12,64/0,83	10,75/0,96
20%	0,3/0,51	0,3/0,36	0,3/0,33	0,24/0,8	0/0,98	0,24/0,55
10%	0,15/0,19	0,15/0,47	0,15/0,15	0,11/1,04	0,11/1,25	0,11/0,98
Average %	2,05/0,41	2,08/0,63	2,04/0,5	2,09/0,9	2,13/0,93	1,89/0,6
G/area(R) %	12,32/2,47	12,5/3,78	12,25/3,01	12,51/5,36	12,75/5,58	11,32/3,59

Table 3.5 In each pair of values, the left one indicates the difference between the initial capability of a UAV (on the left column) and the actual assigned area after the initial baseline algorithm. The right value indicates the same difference after the moveRWS deadlock treatment algorithm. An average for all UAV's, as well as the total difference is shown below each experimental setup, where G is the metric defined in Equation 3.5 and $\text{area}(R)$ is the area in m^2 of the whole region R . The UAV's have randomly distributed initial positions. Setups for 3, 4, 5 and 6 UAV's have been tested, with different relative capabilities and FoV values. Different Lloyd iterations on the mesh have been tested, ranging between 20 and 60.

UAV Capability %	FoV (15 m)			FoV (30 m)		
	Lloyd Iterations			Lloyd Iterations		
	20	30	60	20	30	60
50%	0/0,1	0/0	30,8/0,27	0/0	0/1,49	29,81/0,5
30%	1,19/0,02	0,08/0,02	0,01/0,16	0,024/0,024	20,9/0,65	0,01/0,57
20%	0,01/0,09	0,01/0,01	0,01/0,11	0,024/0,024	0,024/0,84	0,05/1,1
Average%	0,4/0,07	0,03/0,01	10,27/0,18	0,02/0,02	6,97/0,99	9,95/0,72
$G/\text{area}(R)$ %	1,2/0,21	0,09/0,03	30,82/0,54	0,05/0,05	20,92/2,98	29,87/2,17
20%	0/0,43	15,37/0,22	2,19/0,49	0/0,730	14,2/1,66	0,62/0,05
40%	13,1/0,33	0/0,53	0/0,22	14/0,43	0/1,89	0,07/0,14
20%	6,48/0,31	0/0,06	2,64/0,35	6,24/0,56	0/0,12	1,8/0,16
20%	0/0,42	0/0,69	0/0,35	0/0,26	0/0,12	0,02/0,07
Average%	4,9/0,37	3,84/0,38	1,21/0,36	5,06/0,5	3,55/0,95	0,67/0,11
$G/\text{area}(R)$ %	19,58/1,48	15,37/1,5	4,83/1,43	20,24/1,98	14,2/3,79	2,67/0,42

Table 3.5 *Cont.*

UAV Capability %	FoV (15 m)			FoV (30 m)		
	Lloyd Iterations			Lloyd Iterations		
	20	30	60	20	30	60
20%	0,01/0,03	0,01/0,21	11,23/0,77	0/1,8	0,02/1,43	10,78/0,4
30%	0,53/0,02	12,85/0,24	0,02/0,68	0/0,27	13,45/0,42	0,02/1,4
20%	0,01/0,03	0,01/0,11	11,59/2,65	6,24/0,57	6,88/0,44	10,78/0,64
10%	0,01/0,07	0,01/0,19	3,382/1,19	0/0,46	0,04/0,21	3,97/0,04
20%	0,01/0,14	0,01/0,14	0,01/1,36	0/1,03	0,02/0,37	0,02/0,4
Average %	0,11/0,05	2,58/0,18	5,25/1,33	1,25/0,82	4,12/0,57	5,11/0,58
G/area(R) %	0,57/0,27	12,89/0,89	26,24/6,64	6,24/4,12	20,59/2,87	25,57/2,88
10%	0,02/0,46	4,36/0,05	4,2/0,26	0,04/0,36	0,04/0,18	0,04/0,09
20%	0,02/1,02	4,23/0,26	0,01/0,06	0,05/0,3	0,05/0,36	0,05/1,49
10%	0,02/0,42	0,01/0,47	0,01/0,01	0,04/1,2	0/0,41	0,04/0,25
30%	19,04/0,5	0,01/1,22	0,01/0,01	20,32/0,74	20/1,48	19,85/0,31
20%	0,01/0,9	0,02/0,04	0,01/0,02	10,32/0,49	10,32/0,24	10,08/0,65
10%	0,02/0,66	0,01/0,58	0,01/0,2	0,04/0,028	0/1,01	0,04/0,22
Average %	3,18/0,66	1,44/0,43	0,71/0,09	5,13/0,52	5,07/0,61	5,02/0,5
G/area(R) %	19,13/3,96	8,64/2,62	4,25/0,56	30,81/3,12	30,41/3,68	30,1/3,01

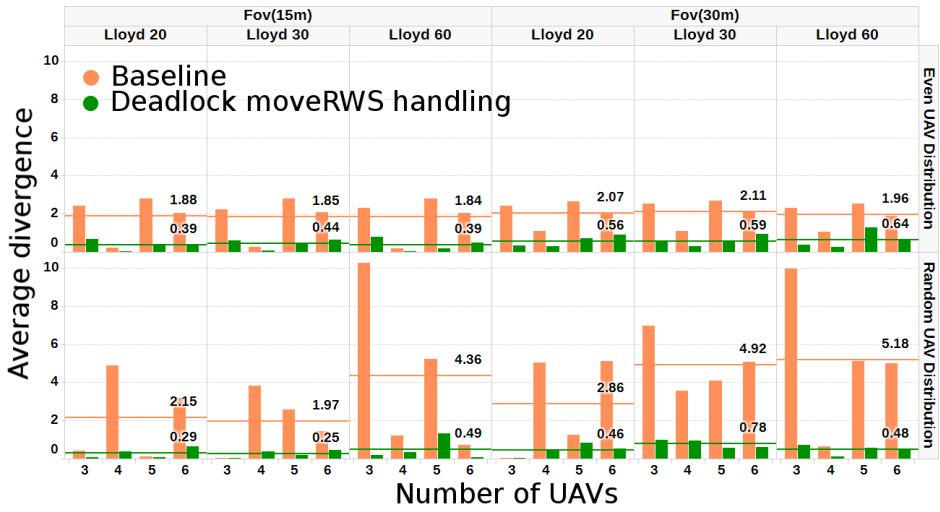


Figure 3.8 A graphical representation of Tables 3.4 and 3.5. Average difference of ideal allocation for every UAV versus final area allocation, after the baseline algorithm (orange) and after the deadlock moveRWS treatment algorithm (green). As expected, random initial positioning of UAVs creates more often deadlock scenarios for the baseline algorithm. The algorithm has been tested for 3–6 UAVs, evenly or randomly distributed in the area. FoV projections of 15 and 30 m have been tested, and in each case, a different Lloyd iteration setting (20, 30 and 60) has been set. The horizontal lines show the average difference.

3.5 Conclusions

The proposed method shows that the produced mesh and associated graphs manage to be consistent with the area properties and the capabilities of the UAVs and their on-board sensors. The defined metrics and the results show that the approach manages to provide an adequate solution without sacrificing the precision which a continuous solution would achieve. The presented partitioning algorithms are capable of treating any UAV setup since they solve deadlock scenarios regardless of the initial positions or the complexity of the area. The latter, as will be described in the following Chapters, along with the attribute of a uniformly expanding partitioning schema, show that the proposed framework provides smooth complete coverage waypoint plans in a decentralized manner. The framework presented so far is actually a generic waypoint planner by treating every centroid of the triangular grid as a waypoint. However, it does not take into consideration the UAV platform dynamics regarding path tracking, even though the pitch and roll upper boundary is used calculate the maximum triangle cell. Nevertheless, a platform might, in the case

of a holonomic vehicle like a multicopter, or might not, in the case of a non-holonomic fast moving fixed wing, be able to follow sharp turns that are produced. Hence, so far this solution does not account for waypoint to waypoint flight trajectories. These issues are usually addressed by the trajectory generator and flight controller, for example, by assuming that a waypoint has been visited if the UAV passes close by. This metric usually is task specific and, in real-world applications, user defined. However, an optimization of the coverage waypoint plans and an enhancement of a state of the art trajectory tracker will be presented in the following Chapters, thus leading to a complete top-to-bottom architecture.

4 Coverage plans

I can go anywhere

THE WHO

Having acquired the subregions for each UAV, we can once more utilise the grid-as-graph approach used in the previous Chapters, in order to produce coverage waypoint flight plans. By treating each triangular cell centroid as a waypoint of a flight plan, complete coverage can be assumed in case the vehicles pass over all of these waypoints. In that manner, the vertices of the area graph should be *prioritised* according to some optimisation criteria, in order to define the order in which they are visited. There are several restrictions to be considered in this prioritization; initially the paths must have limited or no repeated coverage. Several studies show that repeated coverage cannot be avoided, since this issue resembles the *Seven Bridges of Königsberg* or the *multiple Travelling Salesman (mTSP)* problems. Moreover, since this study is focused mainly on non-holonomic vehicles, the resulting coverage plans should have paths as smooth as possible, in order to provide feasible trajectories for the vehicle to follow. As it is already stated, the latter is a critical consideration, since the main goal is to respect the *Region Of Interest (ROI)* of every UAV; as we note in the next Section, current research provides paths which pass over no-fly zones in order to proceed in the next boustrophedon parallel path. Finally, as also presented in the previous Chapters, area characteristics impose several restrictions on the design of such a flight plan. In this Chapter, a proper identification of these area properties as well as a novel algorithm is presented in order to provide smooth, complete coverage flight plans, with minimal repeated coverage.

4.1 Related work

Two main coverage path planning approaches can be identified in literature: continuous and discrete. In both of them, usually a back and forth motion strategy is applied, commonly known as “lawnmower path”, “zigzag” or boustrophedon motions. Several derivatives exist, like the Zamboni flight pattern [45], but in general the simplicity of calculating these paths is preferred over more complex trajectories. There are also efforts for sensor-driven paths[10] in order for the robot to increase the information gain from its sensors, instead of trying to preplan a solution, proposing like so an online strategy for coverage.

In the discrete, grid-like methods, every cell of a segmented space is considered as a node in a, usually, undirected graph[5]. The vertices between these nodes might or might not have weights, which describe a transition cost. In literature, cost or weights are applied depending on e.g. the distance from the target or from the information gain that might have [39].

Regarding repeated coverage of a known area by using a team of robots, the work presented in [46] tries to tackle the problem by evaluating the results by the metrics of total path lengths, total average visiting period, total worst visiting period and balance of workload distribution. In order to also tackle the *art gallery problem* of maximizing the visibility of the robots over the whole area, that algorithmic approach generates a number of points that serve as a kind of guards which maximize complete visibility. These points along with an application of a *CDT* in the whole area, produce a set of nodes which serve as a graph for creating coverage waypoint lists, by using cluster-based or cyclic coverage methods. Coverage with aerial vehicles is also a relevant topic of the previously mentioned survey of Galceran et al. [4], which includes several related papers. In many cases, grid decomposition strategies have been used [5], [47], whereas the enhanced exact cellular decomposition of Li et al. [8] manages to decompose an area in convex cells and by using boustrophedon motions taking into account the *FoV* of the vehicle, provide coverage trajectories with minimum turns. In the latter case of convex decomposition, Maza et al. in [48] also used a convex decomposition method to partition the area.

However, complete coverage is not always achieved and usually, as we have shown in the previous chapters, simple non convex areas are chosen as a test case. In case of non convex areas, the usual strategy is decomposing that region into a sum of convex polygons. Unfortunately sometimes this leads to repeated coverage paths as seen in [11] when the vehicle traverses to the next polygon. Like so, spiral paths might give a better coverage in comparison with boustrophedon movements [8]. In this Thesis we chose a spiral path strategy which can be either outward, from the initial position of the aerial robot to the

borders of the area, or inward, starting from the closest border cell to the initial position. This is achieved by prioritizing each of the triangular cells centroids according to the current position of the vehicle, based on the weights which have been attributed in the decomposition phase. As we will show in the next Sections, several geometrical properties of the area impose specific restrictions for these paths, and as such a novel algorithm is proposed in order to treat those properly.

4.2 Waypoint prioritization

Consider a list of waypoints $W = \{w_1, w_2, \dots, w_i\}$ of a *ROI* as an undirected graph $G = (V, E)$, where the vertices V are the waypoints, and the edges E are the transitions between the waypoints. Also consider the resulting theoretical straight line path P , which connects each of the waypoints, by not visiting a waypoint twice. Let us define as $\theta_{w_i} \angle w_{i-1} w_i w_{i+1}$ the angle between two consecutive edges defined by three consecutive waypoints of P . As discussed before and shown in several studies with aerial vehicles [49], the transitions and turns between the waypoints or between the lines of a lawnmower path, are outside the *ROI*. This is particularly true in fixed wing vehicles, where a constant forward motion cannot allow sharp maneuvers. Some control frameworks treat this issue by making a turn before the waypoint, heading towards the next one, or by passing outside the *ROI* [12] [10]. The proposed method of grid construction guarantees full coverage if the aerial robot passes exactly over the waypoints. Then, the coverage flight plan problem is defined by the following three constraints:

- Construct a spiral path by using each of W just once, avoiding passing over a triangular cell twice.
- Find a minimal P total path length and reduce the repeated coverage which might occur in complex areas.
- Increase the average value of the θ_{w_i} angles of P :

$$\overline{\theta_w} = \sum_{i=2}^{N-1} \theta_{w_i} / (N - 2) , \quad (4.1)$$

where N is the number of waypoints.

By combining these constraints, smoother trajectories are expected [50].

4.2.1 Spiral coverage plans

The framework presented in the previous Chapter, and in particular the weight attribution schema of Algorithm 2, can be applied to generate waypoint lists for the UAV to achieve complete coverage of a complex coastal sub-area. By using the border-to-center cost described in Section 3.2.2, inward spiral-like waypoint lists W can be generated. Algorithm 6 performs a selection of vertices by initiating from the vertex that has the highest *Reverse Watershed Schema (RWS)* cost $D(v)$ and is closer to the starting position of the UAV. In every recursion, the closest adjacent cell v_j that has the same cost ($D(v_j) = D(v)$) is inserted in the waypoint list. In case all of the vertices of the same D_v cost have already been inserted in the list, the algorithm reduces the cost D_v and chooses the cell that is closer to the previous step. It should be mentioned that the complexity of this algorithm is $\mathcal{O}(n^2)$; as such, Figure 4.1 presents the relative time cost of decomposition, partition, Lloyd optimization and waypoint plan calculation of Table 3.3, of the previous Chapter results.

Algorithm 6: Waypoint list computation for coverage. D_c is an auxiliary variable with the current border-to-center cost in each step, whereas v_{I_k} is the starting position of the UAV U_k . Function *findClosest* finds the closest vertex to the current one that has its same border-to-center cost. CDT_k is the sub-CDT for UAV U_k . W is the produced waypoint list

```

1  $D_c \leftarrow \infty$ ;
2  $v \leftarrow \text{findClosest}(v_{I_k}, D_c)$ ;
3  $W.\text{insert}(v)$ ;
4 foreach  $v \in CDT_k$  do
5   if  $\exists v, D(v) = D_c$  then
6      $v_j \leftarrow \text{findClosest}(v, D_c)$ ;
7      $W.\text{insert}(v_j)$ ;
8      $v \leftarrow v_j$ ;
9   end
10  else
11     $D_c \leftarrow D_c - 1$ ;
12  end
13 end

```

In order to simulate how an actual coverage trajectory would be by the proposed algorithm, a scenario has been chosen in the same area of Figure 2.13. The area was partitioned into sub-areas respecting the different UAVs' capabilities as described in the previous Chapters (see Fig. 4.2a) and their respective coverage paths connecting the waypoints can

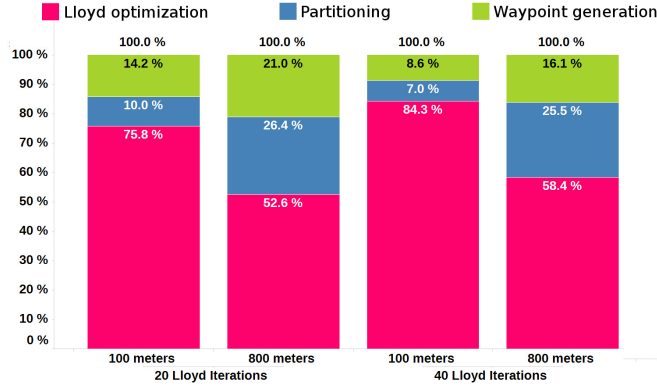


Figure 4.1 A percentage bar chart depicting the average time needed for each of the operations of all the simulations of Table 3.3. This plot also includes the waypoint generation strategy. A comparison between a large FoV projection side size (800 metres) and small FoV side size (100 metres) is presented, while also comparing the difference between different Lloyd iteration settings.

be seen in Fig. 4.2b. Finally, Fig. 4.2c shows the simulated trajectory for an UAV, in the simulation framework which will be described in Section 4.3.

This method offers complete coverage in simple environments. Nevertheless, geometric properties in complex areas indicate that this method increases repeated coverage as it introduces back and forth motions in narrow passages, as described in the following section.

4.2.2 Spiral plans and area characteristics

By using the aforementioned method, spiral paths create repeated coverage patterns in areas like (C) of Figure 4.3. We use the term *valleys* to refer to this kind of areas, while areas like (A) and (B) of the same figure are referred as *inlets*. As it is also described in [51], the "valley" areas are produced in non-simply connected environments and a boustrophedon or, in our case, an inward or outward spiral path will cover them at least twice.

Like so, the algorithm of the previous section is extended and improved by introducing an *inlet sensitivity* coefficient I . The new algorithm (see Algorithm 7) identifies the existence of *inlets* in the decomposition schema by comparing the total path length for different I values. In that manner, the resulting path visits isolated areas first before continuing to the rest of the area. However, since our goal is also to produce smoother trajectories, $\overline{\theta}_w$ is also considered in the evaluation of I .

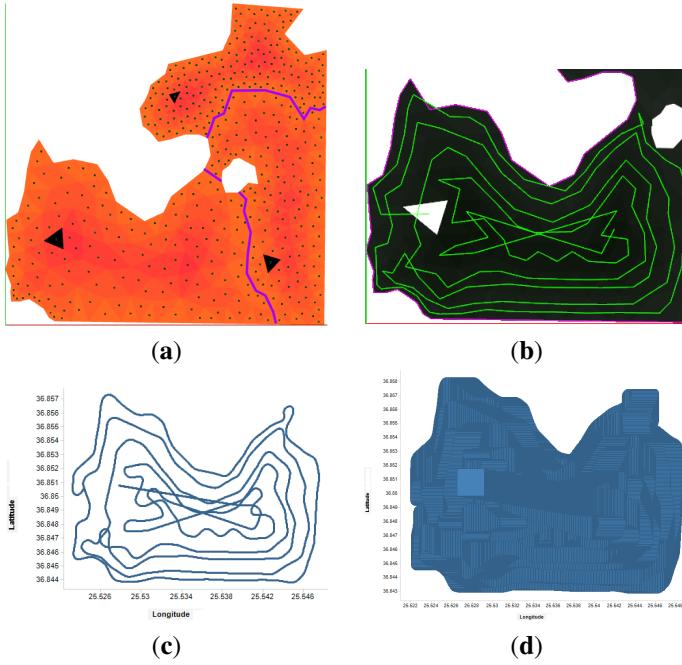


Figure 4.2 In (a), area partitioning for three UAVs on the same location as in Figure 2.13. White areas indicate the no-fly zones, whereas the black triangles show the initial positions of the UAVs. The FoV sized cell distribution is shown along with the centres of the triangles. In (b), a detailed view of the UAV 3 waypoint prioritization, while in (c) latitude and longitude information received during the simulated flight of that UAV. Finally, (d) shows the the total coverage of a squared FoV simulated sensor, working at a very slow rate of 1 Hz.

Let G_i be the centroid of cell v_i and G_{i_j} the centroid of its j neighbouring cell. Consider the maximum euclidean distance L among all neighbouring centroids of the CDT:

$$\|L\| = \max_{i \in N} d(G_i, G_{i_j}), 1 \leq j \leq 3 \quad (4.2)$$

where N is the number of waypoints.

This distance is chosen as the initial and minimal value of I . Please note that I is actually a multiplier for L . As such, the initial value $I = 1$ equals $I = 1 * L$. Then, different I values are actually multiplier values for L .

Let us consider two sets

$$\overline{\theta}_w(\alpha, I) \Big|_{i=I}^k = \overline{\theta}_{w_i}, \overline{\theta}_{w_{i+1}} \dots \overline{\theta}_{w_k} = \Theta \quad (4.3)$$

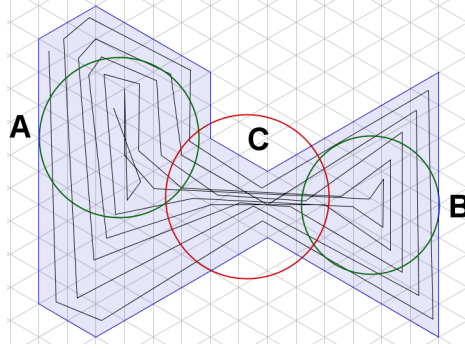


Figure 4.3 A waypoint list of a triangulated area computed with the coverage Algorithm 6. The regions in A and B create a valley in C in the borders-to-center cost attribution schema as their surrounding region has higher costs. Since the algorithm initially visits cells according to their distance from the borders, a lot of back and forth paths are created in region C.

and

$$P(\alpha, I) \Big|_{I=i}^k = P_i, P_{i+1} \dots P_k = \mathbf{P}, \quad (4.4)$$

where α is the area to be covered, k is an upper limit for the I value and Θ, \mathbf{P} are the subsets of the original sets, for the evaluated I 's.

Let us also consider the functions

$$f_1 = \left| \frac{\Theta}{\text{norm}\Theta} \right| \quad (4.5)$$

and

$$f_2 = \left| \frac{\mathbf{P}(-I)}{\text{norm}\mathbf{P}} \right| \quad (4.6)$$

where $\mathbf{P}(-I)$ is the mirror function of P , and their difference

$$\varepsilon = f_1 - f_2 \quad (4.7)$$

Then, the path enhancement problem is to find the optimal I , which occurs in the intersection of f_1 and f_2 , where $\varepsilon = 0$. This problem is a multi-objective optimization problem, where the optimal I value is the one which minimizes the total path P and maximizes $\overline{\theta}_w$.

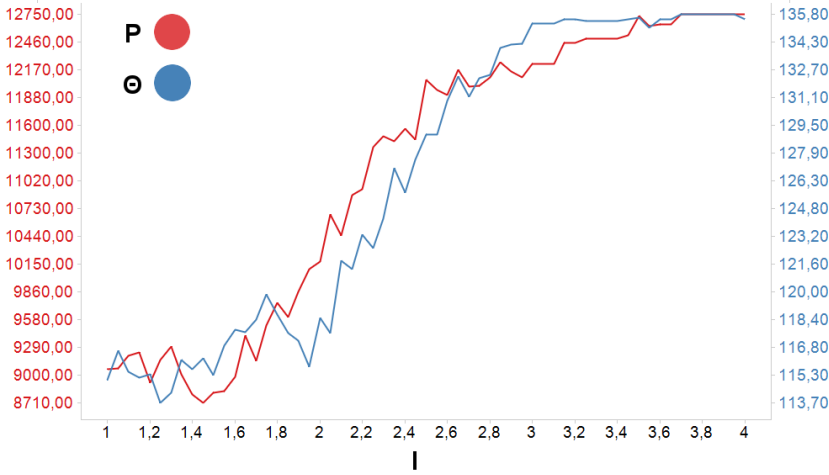


Figure 4.4 As I values increase, so does the path length and the average angles in a linear correlation, where $I \in [1, 2, 4]$, with a step of 0,05. Path P is measured in metres while angle Θ is in degrees.

As seen in Fig. 4.4, there is a linear correlation of I , P and Θ , where $I \in [1, 2, 4]$. As expected, different optimal values of I are found in different scenarios, since the algorithm is dependent on the area or cell size, as it can be seen in Fig. 4.5.

It can also be seen that in the top sub-figure of Fig. 4.5, some areas and configurations will not provide an intersection of f_1 and f_2 . In order to reduce the computational complexity of the regression analysis which is needed for finding an optimal I , an approximation is chosen:

$$I = \min \varepsilon : \varepsilon \leq T \quad (4.8)$$

where T is a given threshold. As such, an I value is found where f_1 and f_2 have the minimum distance, as in Fig. 4.6. Here we have to note that this solution is strictly platform dependent, since the constraint of maximizing $\overline{\theta}_w$ is a consideration which provides smoother trajectories on non-holonomic vehicles, whereas is not crucial in holonomic aerial vehicles like quad-rotors. In those cases, minimizing P alone is enough for choosing I and identifying *inlets* and *valleys*.

Please note that, as also shown in several cases of Fig. 4.6, the value of I has limits of $1 \leq I \leq 4$ and has been extracted empirically, since higher values produce higher path lengths without providing better results regarding $\overline{\theta}_w$.

The algorithm works as follows: as before, consider a *CDT* grid H which is an aggregation of its triangular cells $H = \{h_1, h_2, \dots, h_n\}$. This is treated as an undirected graph

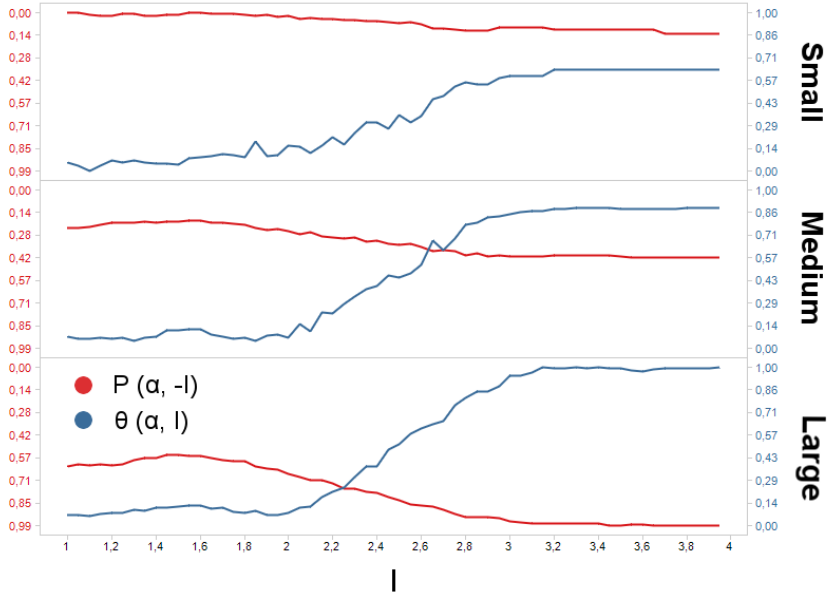


Figure 4.5 The intersection of P and Θ for different area sizes, in relation with I . Please note that the vertical axes are normalised in $[0,1]$ and P axis is inverted. The evaluation step for I is 0,05.

$G = (V, E)$. Then, the RWS cost for every triangular cell/vertex of the graph, as introduced in Algorithm 2, is used as a *border proximity cost* $D(v)$. As such, every cell gets a border proximity cost which indicates how far is that waypoint from the borders. Then, consider the aforementioned *inlet* (as defined in [51]) *sensitivity coefficient* I , which initially has the value of the largest edge E of G : $I = \max(E)_{G(V,E)} = \|L\|$. Initially, path P is constructed by prioritizing the waypoints as follows: a border cell is chosen and one of its neighbours with the same $D(v)$ is considered as the next step. If there are no neighbours with the same $D(v)$, the closest cell which has that $D(v)$ value is chosen; if it is further than the absolute value of I , then the algorithm searches for a neighbour with $D(v) = D(v) - 1$, until no neighbours are found. In that case, the closest cell with the highest $D(v)$ is chosen, and a full P path is constructed. By comparing P length and $\bar{\theta}_w$ for different values of I by using the aforementioned threshold T , an optimal I is found as described before.

4.3 Results

In order to validate the proposed algorithms and compare with methods found in literature, both simulations and actual flights have been performed. In order to proceed to

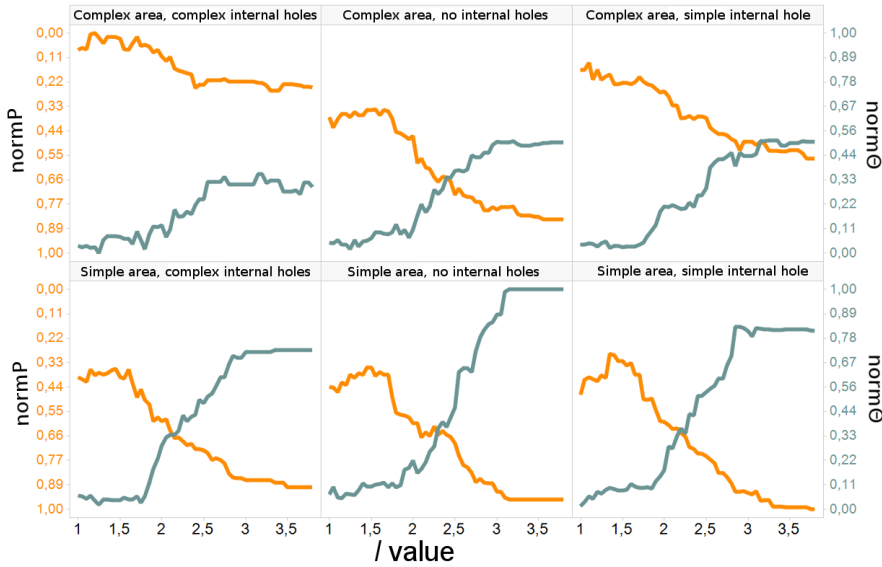


Figure 4.6 Different values of I are required in different scenarios of relative same size areas in order to find an adequate shorter path/wider angle equilibrium. Please note that the vertical axes are normalised in $[0,1]$ and P is inverted. The evaluation step for I is 0,05.

a complete experimental framework as will be discussed in the following Chapters, a modular architecture is chosen having as core the *Robotic Operating System (ROS)* [52] framework. Moreover, an application has been developed which manages to perform full parametrization of the conducted experiments while on the same time providing valuable visualization tools and quantitative data. These tools remain the same for both simulated and actual experiments, demonstrating the feasibility of the solution in an embedded, onboard companion computer.

4.3.1 Simulation framework

The simulations have been performed on computers with an Intel Core i5-5200U@2.20-GHz CPU with 8 GB of RAM and the kUbuntu 14.04 distribution of the Linux OS. The framework adopted is shown in Figure 4.7.

The main application is based on the Qt(<https://www.qt.io/>) cross-platform software development environment. The setup consists of a configuration window (Figure 4.8) where the number of the UAVs along with their attributes can be set. These attributes are the sensor type, the *FoV* size referring to the maximum triangular side size, as had been defined in Section 2.4.1, a percentage of the whole region to be used in the partition step,

Algorithm 7: Coverage waypoint list with inlet sensitivity. D_{v_c} is the border-to-center cost of cell v as it has been assigned by the method in Algorithm 2. D_c is an auxiliary variable with the current border-to-center cost. I is the inlet sensitivity coefficient which is compared for every next step. Cell v_{I_k} is the starting position of UAS U_k . P is the produced waypoint list path of vertices. CDT_k is the sub CDT for UAS U_k . Function *findClosest* finds the closest vertex to the current one that has its same border-to-center cost. Function *calculateDistance* measures the distance between two vertices. Vector *Aux* is the placeholder for inlet vertices that are to be visited in the future.

```

1   $D_c \leftarrow \infty$ ;
2   $v \leftarrow \text{findClosest}(v_{I_k}, D_c)$ ;
3   $P.\text{insert}(v)$ ;
4  foreach  $v_i \in CDT_k$  do
5      if  $\exists v, D_{v_c} = D_c$  then
6           $v_j \leftarrow \text{findClosest}(v, D_c)$ ;
7          if  $\text{CalculateDistance}(v, v_j) > I$  then
8               $Aux \leftarrow v_j$ ;
9               $D_c \leftarrow D_c - 1$ ;
10         end
11         else
12              $P.\text{insert}(v_j)$ ;
13              $v \leftarrow v_j$ ;
14         end
15     end
16     else if  $Aux = \emptyset$  then
17          $D_c \leftarrow D_c - 1$ ;
18     end
19     else
20          $v_j \leftarrow \text{findClosest}(v, Aux)$ ;
21          $P.\text{insert}(v_j)$ ;
22          $Aux.\text{remove}(v_j)$ ;
23          $D_c \leftarrow D(v_{j_c})$ ;
24     end
25 end

```

initial positions and tasks. The configuration application sets the type of visualization that will be conducted by the rviz visualization package [53]: showing the borders of each sub-area, colouring it depending on different parameters and showing the produced waypoints for coverage. Regarding the CDT, its constraints of minimum angle and initial triangulation maximum edge can be also defined, and the user can define the area of interest by uploading a KML file, including obstacles. Finally, each step of the simulation

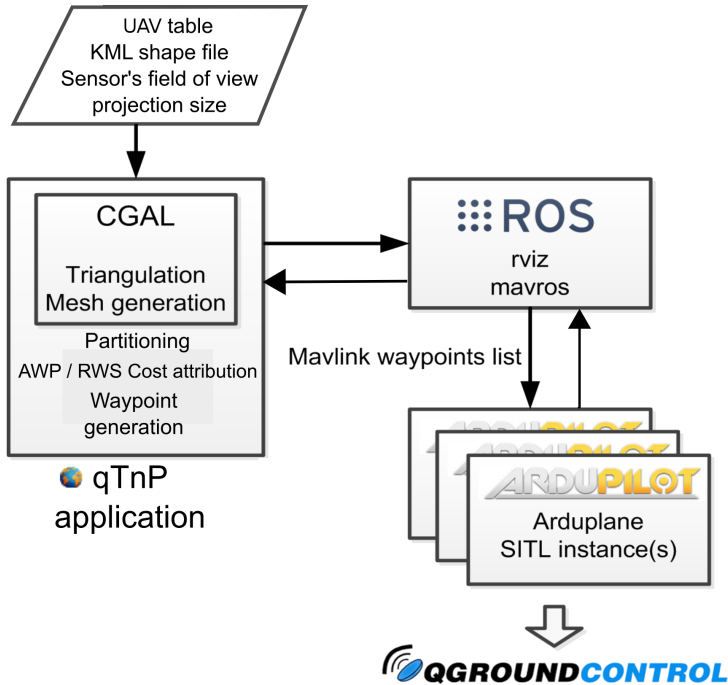


Figure 4.7 Experimental framework with different libraries and components: the latest *CGAL* library (4.8.1) [44], *ROS* Indigo [52] components (the *rviz* package [53] for visualization and the *mavros* node [54] for the mavlink interface with the simulated UAV), an Arduplane instance [55] of the Ardupilot SITL [56], which uses the JSBSim flight dynamics model [57], and the *qgroundcontrol* control station [58].

can be performed separately; performing the triangulation, extracting the partition for each UAV based on its percentage of the total region and computing coverage waypoint plans for each UAV.

The implemented algorithms are part of a ROS node named *qTnP* (Qt Triangulation and Planning, Figure 4.8). This node performs all calculations and manages the communication with the rest of the ROS nodes of the configuration. Visualization of the mesh of the area, partitioned areas, cost attribution, waypoints and produced paths is handled by the *rviz* node, whereas the produced waypoint stacks are sent to the *mavros* node. This node has a dual purpose. It maintains the connection with the simulated vehicles, sending waypoint list plans when the main application produces them. It also listens to the simulated UAVs, which report the *mavros* node on each cycle for their current position and telemetry data.

Regarding the UAV model used in the simulations and its on-board controller, the open source autopilot Ardupilot has been used. Its arduplane instance for fixed wing model

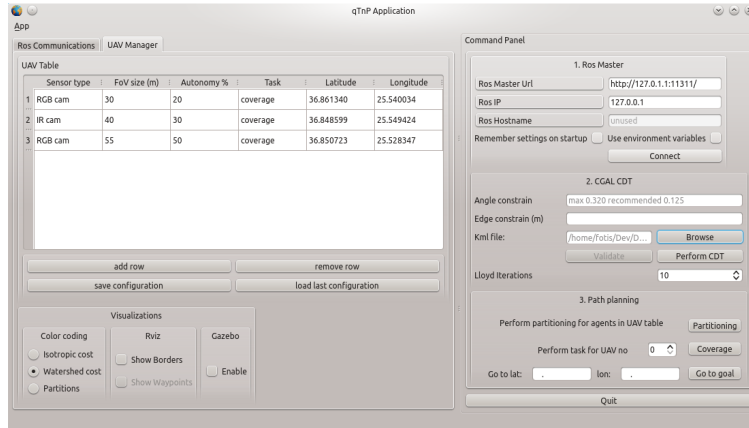


Figure 4.8 The qTnP main application. The first tab echoes the ROS communication messages and logs. The main "UAV Manager" tab of the application includes the UAV management table, indicating the sensor type, the cell *FoV* size, autonomy percentages and initial positions. It also includes the visualization options for rviz, showing the cost values of each of the proposed algorithms, visualizing the partitioned configuration space, showing the borders of each UAV and the produced waypoints for coverage. Finally, the command panel on the right includes connection settings, CDT-specific configuration, the KML file of the area, as well as several command buttons for the different stages of the experiments.

aircraft has been combined with the JSBSim flight dynamics model simulator. In our setup, the system simulates the dynamics of the Rascal110 model airplane. The Arduplane controller used is the Pixhawk Flight Management System [59]. The behavior of the vehicle during the simulated flight, as well as the produced trajectories were monitored live using the open source ground station qgroundcontrol [58].

4.3.2 Comparison with a grid decomposition and simulations

The whole area of the initial simulated experiment of Figure 4.2 has been selected in order to compare the initial strategy of Algorithm 6 with a grid decomposition. Although the actual path is longer, as expected due to repeated coverage, and the number of turns is higher, the area was fully covered (see Fig. 4.9) respecting the aerial restrictions.

These waypoint path extraction results have been compared with a series of other configurations of a classical grid decomposition strategy as shown in Fig. 4.10. Table 4.1 presents various metrics regarding the path length, the minimum and maximum angle values of the path, and the percentage of flight over no-fly zones for the grid cases simulation. Although the suggested path by the waypoint lists of the proposed algorithms is longer,

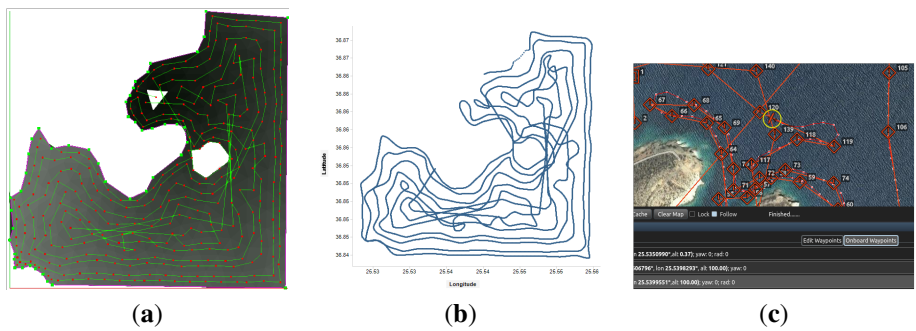


Figure 4.9 An area segregated in triangular cells by a FoV projection size of 50 metres. Figure (a) shows the produced mesh of 519 cells and the waypoint list with each vertex connected to the following with a line segment, according to Algorithm 6. The triangular white cell indicates the initial location of the UAV, while the white areas represent the obstacles or no-fly zones. Sub-figure (b) shows the simulated trajectory results for the Rascal110 model aeroplane from the latitude and longitude information received by the mavros node during flight. In (c), an in-flight screenshot of the ground station during the simulation, where the behavior of the simulated autopilot can be observed; narrow turns produce paths outside of the expected trajectories.

Table 4.1 Path metrics comparison when a classical grid decomposition strategy has been used for the areas shown in Fig. 4.10.

	Waypoint generation case				
	(a)	(b)	(d)	(e)	(f)
Path length (m)	8132,26	10580,86	6741,16	6988,05	10315,48
Min angle (deg)	0,94	26,98	18,43	0,01	0,01
Max angle (deg)	179,91	179,94	151,97	173,66	139,4
Avg angle (deg)	120,22	147,14	84,87	87,75	108,98
% of path over no fly zone	0,032%	0,026%	11,467%	9,521%	8,39%

the results suggest that: (a) the algorithms provide smoother trajectories since the average angles of the paths are wider and (b) there are no paths produced over no-fly zones or outside the areas of interest.

Inlet coefficient method

In order to identify the geometrical properties and spot the qualitative differences in different inlet sensitivity coefficient settings of Algorithm 7, two areas have been chosen. The first is a sample area with several “valley” areas. The second is a scenario in an actual area presented in Fig. 4.11. On the latter, a simulated flight has been performed in order

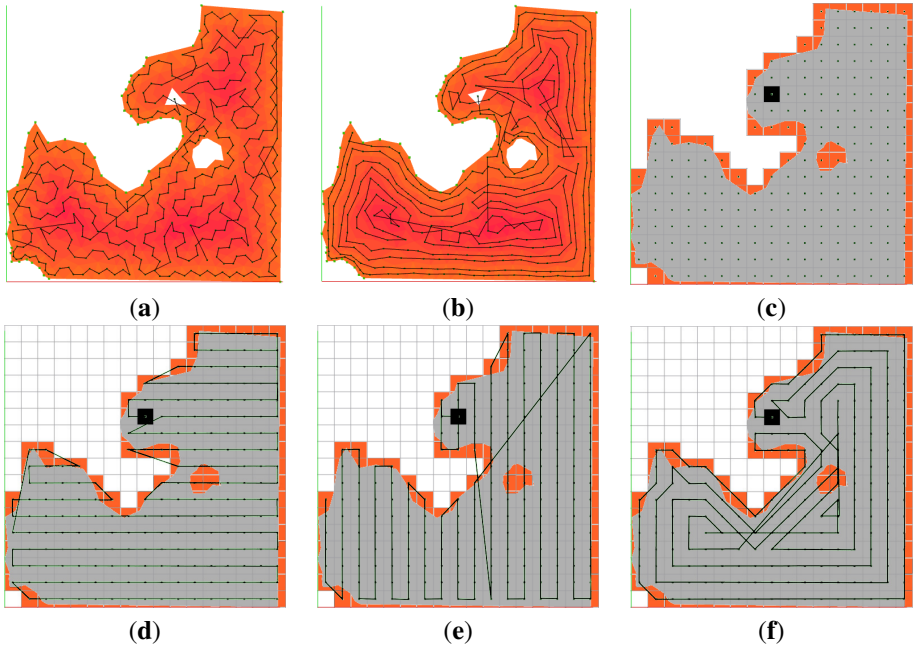


Figure 4.10 A comparison between the proposed method and a classic grid decomposition with boustrophedon coverage motions. The first two figures use the CDT schema as described in the previous Chapters, resulting in a mesh of 391 triangular cells. Sub-figures (a) and (b) present a different inlet sensitivity coefficient in order to highlight the resulting paths. The rest of the sub-figures show a classic grid overlay on top of the area, having a total of 210 square cells of which 78, in orange color, partially having a no-fly zone in them. The black square is the respective initial position. Figures (d) and (e) show two boustrophedon strategies which have been tested in order to compare them with the strategies of (a) and (b) whereas Fig. (f) uses the same border-to-center cost strategy of Section 3.2.2 in order to obtain a waypoint list plan by Algorithm 6, properly adjusted for a grid decomposition.

to obtain results of the real trajectory followed. Please note that in order to show the difference of different sensitivity settings, the I values were selected implicitly.

Sample area

This area has been segregated and partitioned for two UAVs; one of them visits three out of four “valley” regions, as it can be seen in Fig. 4.12.

The two settings that have been chosen demonstrate the effects that this algorithm might have: on lower inlet coefficient values, some vertices might be identified as inlets while they are not. As a result, repetitive turns are produced and the paths are far from smooth.

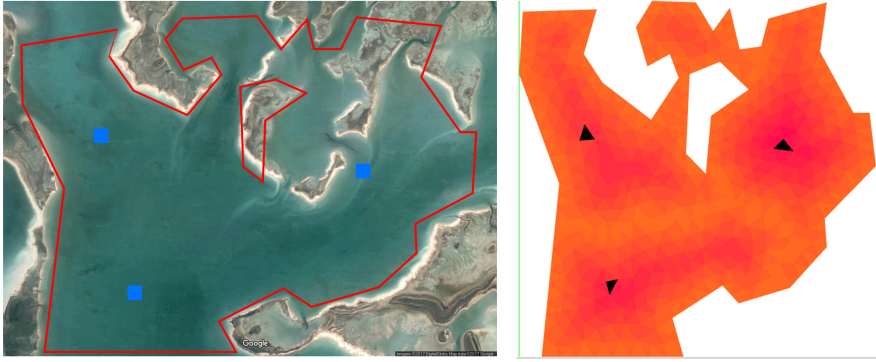


Figure 4.11 On the left, a region in Great Abaco island near Miami. The red polygons define the area constraints. The blue boxes indicate the initial positions of each UAS. On the right, the same region partitioned for the three UAS. Their initial positions are the black cells, whereas each shade of orange is the border-to-center cost. Their coverage percentage in relation to the whole area is 30%, 40% and 30%. The CDT is a result of 10 Lloyd iterations.

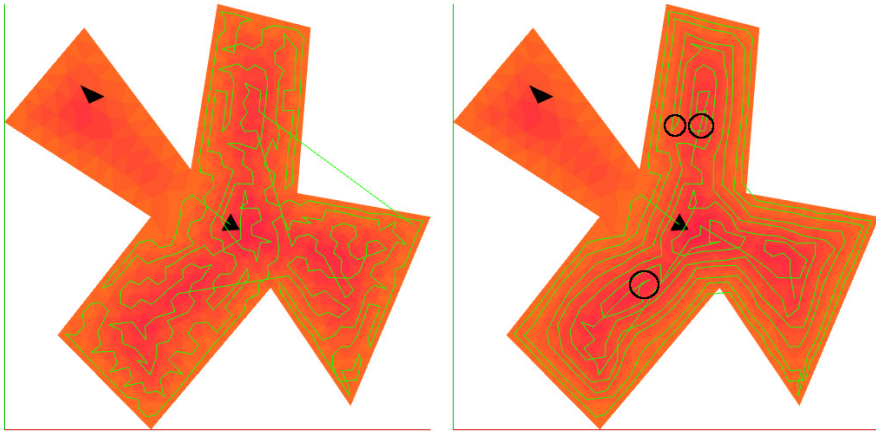


Figure 4.12 Spiral paths for different inlet coefficient values. In the left case, the algorithm identifies a lot of next steps which are considered inlets, making a lot of turns. On the right, the paths are smoother. In the black circles, some points where the algorithm identified a far next movement and chose the closest cell instead.

On higher inlet coefficient values, the paths are smoother and the inlets are identified correctly, but the total length of the path is larger, as shown in Table 4.2. In order to compare these results with the initial Algorithm 6, tests have been conducted for three different Lloyd smoothing settings: 10, 30 and 60 iterations. The algorithm has been tested with a inlet coefficient of 4 times the $\|L\|$ value. The upper panel of Fig. 4.13 shows that

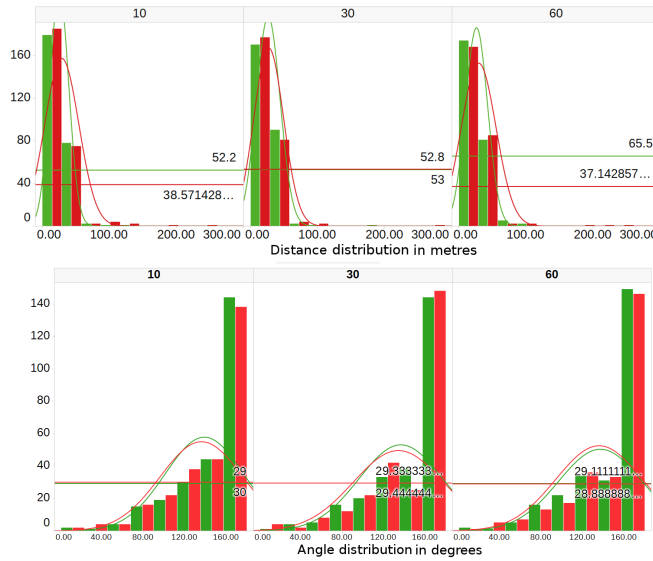


Figure 4.13 A comparison between Algorithm 6 (depicted in red color) and Algorithm 7. In the upper panel the average distance between each cell might be larger but as the distribution curve shows, also clear in Table 4.3, the total path length is smaller. In the lower panel, the angle distribution is shown. While there is no big difference, the distribution curves show that with the new algorithm, there is a shift towards larger angles, thus smoother paths. Both panels are in columns which represent the different Lloyd smoothing settings (10, 30 and 60).

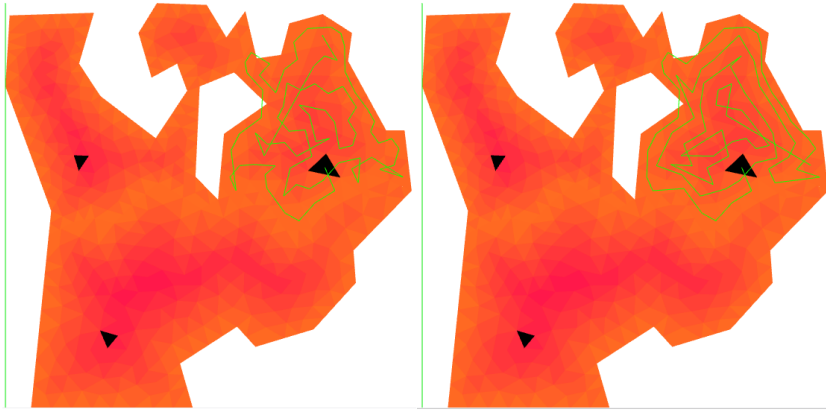
Table 4.2 Metrics on inlet coefficient corresponding to Fig. 4.12.

	Inlet coefficient	
	$2* L $	$4* L $
Total Path(m)	7139,51	8515,79
Min distance(m)	9,87	10,63
Max distance(m)	226,51	191,72
Min angle(degrees)	20,60	5,09
Max angle(degrees)	179,889	179,99

the distribution dispersion of distances between each cell is wider. This results in longer total path lengths, as it is also shown in Table 4.3. In the lower panel, the distribution curve shows that in average the new algorithm also manages to produce smoother paths, by increasing the angles between each three waypoints.

Table 4.3 Comparison between Algorithm 6 and Algorithm 7, corresponding to Fig. 4.13.

Algorithm	Lloyd	Total path (m)	Distance Min (m)	Distance Max (m)
Initial	10	7736,95	5,77	255,15
	30	7322,59	6,02	290,29
	60	7502,25	12,99	233,97
Current	10	6958,26	10,41	144,13
	30	7144,34	12,99	174,32
	60	7098,88	12,99	94,11

**Figure 4.14** The difference in coverage path planning can be identified mainly in the number of sharp turns, repeated paths and total path length. The image on the left represents an inlet coefficient value of 2 whereas the one on the right a value of 4. The total metrics of the example can be seen in Table 4.4.

Actual area scenario

The area of the upper right UAV in Figure 4.11 has been selected as an actual area scenario for a mission. In Fig. 4.14, the qualitative difference in the calculated waypoint list path can be identified and Fig. 4.15 shows the angle distribution in the two different cases; it shows that the dispersion of angles is greater in smaller inlet coefficient values but the produced total paths are smaller. On the other hand, bigger I values provides wider angles, thus smoother paths. Then, the total metrics of the produced paths can be seen in Table 4.4.

As shown from Fig. 4.15, both increased Lloyd iterations and inlet sensitivity values manage to increase the average angle metric. The actual simulated trajectory can be seen in Fig. 4.16(a) and a screenshot of the simulated experiment in Fig. 4.16(b).

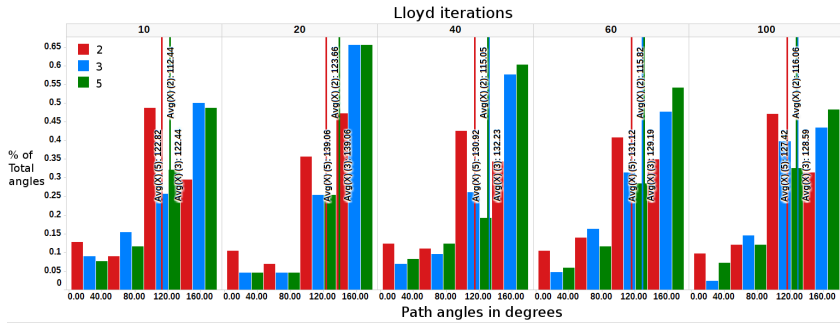


Figure 4.15 The angle distribution in the produced path of the Abaco region. Red, blue and green colors represent the different inlet sensitivity coefficient values of 2,3 and 4 respectively. Each panel shows the different Lloyd iteration settings. The horizontal axis distributes the angles in 5 bins, whereas the vertical axis show the percentile rank of each bin. Vertical lines show the average angle of each case.

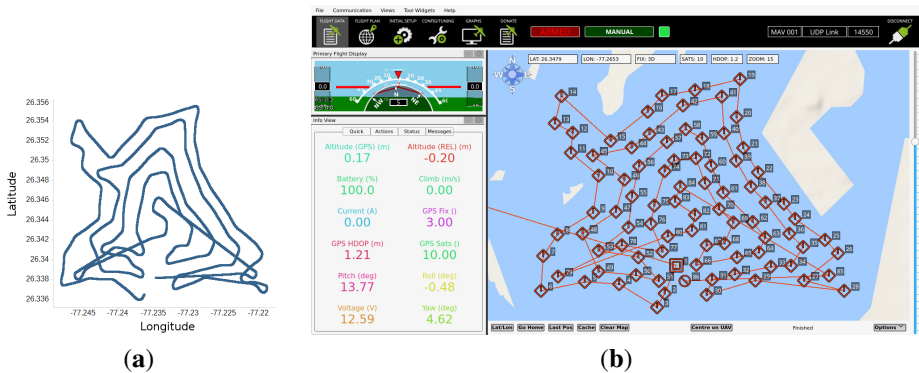


Figure 4.16 In (a), the actual trajectory when the simulated model followed the produced waypoint plan. In (b), the APM Planner 2 ground control station showing the loaded waypoints. The mission had a flight setup of 100 metres altitude.

Table 4.4 Metrics on inlet sensitivity values corresponding to Fig. 4.14.

	Inlet coefficient value		
	$2^* L $	$3^* L $	$4^* L $
Total Path(m)	3807,88	4185,54	4344,05
Min distance(m)	12,90	12,90	12,90
Max distance(m)	216,87	127,52	118,24
Min angle(degrees)	11,5629	11,69	0,8535
Max angle(degrees)	179,667	179,667	179,913

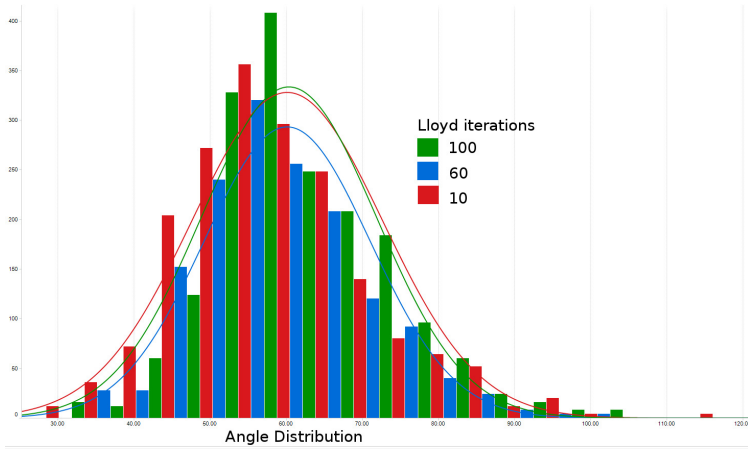


Figure 4.17 Angle distribution in different Lloyd iteration trials. The red Gaussian curve which represents the lesser number of iterations tested, shows that there is a larger dispersion in angles, resulting in more “sharp” cells. While the best performance is achieved by the 100 iterations, where there are more angles reaching the 60° ideal case, the required computation time should be considered. Depending on the application, a solution closer to 50 iterations could be competent enough.

Configuration comparison

Regarding the use of Lloyd optimization for creating a smoother grid, results show that the angle distribution of the grid is more coherent with increased number of Lloyd iterations. Nevertheless, this operation is strictly dependent on the area input as well as the constraints of the CDT. In the distribution shown in Fig. 4.17, in the 100 iterations case, the algorithm did not manage to overcome the 65 iterations thus not been able to smooth the grid any more.

The complexity effect of the algorithm is clear in Table 4.5, where the time each smoothing procedure needs significantly increases, thus exposing the trade off among sample overlapping and smoother or shorter paths.

4.3.3 Field experiments

Two sets of experiments (M1 and M2) were conducted to test the proposed algorithm. The first set is a classical boustrophedon mission (M1) in the same area and it is used in order to obtain a comparative result of the classical method [4] with the proposed method. The second set of flights demonstrates the proposed method while it includes an optimised

Table 4.5 Time in seconds for Lloyd optimization in the whole area and in each UAV configuration space.

CDT	Lloyd Iterations		
	10	60	100
All	0,2s	0,86s	1,008s
UAS1	0,076s	0,36s	0,32s
UAS2	0,064s	0,316s	0,392s
UAS3	0,156s	0,552s	0,6s



(a)



(b)

Figure 4.18 In (a), the area which the experiments took place. The red external and internal borders represent the limit and the internal no-fly zones. In (b), the Skysurfer fixed-wing platform used, operated in a maximum speed of 15m/s.

tuning with a stringent acceptance radius (M2). Please note that the term acceptance radius, refers to the UAV-waypoint distance where the waypoint is considered visited.

The experiments have been conducted in the agricultural area near Brenes, Spain (Figure 4.18a). A COTS fixed wing platform(Figure 4.18b) has been used which had the proposed framework on-board. Moreover, a square theoretical sensor with a *FoV* projection of 100 metres side is considered in order to provide a metric of coverage.

The comparison of the two trajectories (see Figure 4.19) shows that the proposed method has managed to significantly reduce the flight over no-fly zones (Q_{out}) and total flight length (Q), while at the same time performing a slightly better coverage (see Table 4.6).

The missions were executed in different meteorological conditions during August and September in the Spanish province of Seville where the winds are relatively low and the temperatures are high, as they can go above 50°C. These extreme conditions affect



Figure 4.19 In (a), the trajectory of the boustrophedon path and the respective theoretical sensor coverage (in blue shade). The red external and internal borders represent the limit and the internal no-fly zones. As shown, several violations, in green ellipsis, of the no-fly zones have occurred in turns. In (b), actual trajectory if the proposed method, with a calculated inlet sensitivity coefficient value of 2,35. As before, the blue shade represents the theoretical sensor coverage, while the *ROI* violations are shown in green circles. The flight over the no-fly zone is limited, whereas the sensor coverage is extended.

Table 4.6 Comparative experimental results for the boustrophedon and proposed method missions.

Mission ID	Q_{out} (m)	Q (m)	Coverage (%)	Q_{out} (%)
M1 _{avg}	775,09	7204,8	94	10,75
M2 _{avg}	96,65	5925,84	97,2	1,63

the performance of the power sources, therefore, the appropriate tuning is necessary to increase flight endurance in order to finish the missions within appropriate safety levels. This method will be described in the following Chapters.

4.4 Conclusions

The experimental results show that the proposed algorithm aids the process of reducing the overall flight path length and time, while reducing the flight over no-fly zones, in complex areas. The latter also supports the safe separation consideration, in context with the main contribution of this Thesis. Even though initial simulations showed that a boustrophedon *P* path was shorter, the actual experiments demonstrated that this is not actual the case. This occurs due to the aforementioned extra trajectories which occur during turns. Regarding

the average turn radius, our intuition was correct: as smoother paths are provided by creating wider turns, less energy consumption has been achieved.

We have to note that except the non-holonomic nature of the considered vehicles, flight or vehicle dynamics have not yet been considered. The following chapter will describe in detail several considerations and cases on those issues. Nevertheless, it is clear that holonomic vehicles like multi-rotors would not demonstrate this behaviour, as they are able to perform sharp turns and follow the produced waypoint plans with more precision. However, these considerations are somehow application specific and heavily dependent on the task to be performed.

5 Path tracking

'Let's just drive. Maybe we ought to go back, though?'
'No, never-never! Let's go on. I can barely see the road. We'll make it.'

JACK KEROUAC - ON THE ROAD

This chapter introduces a method for path tracking based on a basic *pure pursuit*(PP) algorithmic strategy. In the context of the coverage waypoint lists described in the previous Chapter, a guidance strategy proposal is made, where for each next waypoint of the vehicle, a goal point is chosen in order to reduce a deviation error from the ideal straight line path and the current position of the vehicle. In order to calculate this point, an offline lookup table is created, which correlates the underlying control model and the system dynamics and delays with the turn angles; during flight, the best value is chosen for each next turn. This algorithm has been tested in different scenarios against the simulation framework described in the previous Chapter and also in field experiments. The results show that this strategy manages to provide a computationally inexpensive method for the goal points of the pure pursuit algorithm and reduce the deviation from the ideal straight line trajectories connecting the waypoints.

5.1 Related work

Path tracking in autonomous vehicles is the process which connects the global path planner for a task with the underlying control mechanisms. The on-board controller utilises

information from the sensors and the localisation techniques are responsible for current position tracking in relation with the desired trajectory; in the case of a non-holonomic Unmanned Aerial Vehicle (UAV), the appropriate lateral acceleration steering motions are responsible for the resulting trajectory and are sent to the platform controller at a given rate.

In literature we can identify many approaches for path tracking which use a pure pursuit (PP) algorithm, by taking into consideration different scenarios and applications. In [60], the authors report high performance, best stability and accuracy of the vehicle by using a pure pursuit algorithm, due to the ease of tuning the gain of the lookahead distance and the absence of noisy derivative terms. The same method and reasoning behind the study, has also been the goal of the authors in [61], where a stability analysis has been performed in the followed path. The study in [62] also considers the vehicle stability problem applied in industrial forklifts as an example. Their hybrid approach is also based on a PP algorithm for calculating the next goal point, which is sent to the control module in order to produce continuous smooth curvature trajectories. The authors in [63] state that they use the PP algorithm due to its robustness in large disturbances, a criterion which is crucial in autonomous car design, while in [64], for autonomous cars also, the authors provide a fuzzy controller approach to determine and tune the lookahead distance of the PP method. Regarding path following algorithms for UAVs, the comparative study in [65] presents several strategies used in literature, stating that while methods like vector field following [66] are more precise, their computational expense might make other non-linear methods more appropriate; thus the decision on the path tracking algorithm is heavily application dependent. This is particularly true in small sized fixed wing UAS, since their limited takeoff weight along with an always forward constant motion, dictates for a fast and computationally cheap algorithmic approach.

As shown from all the previous studies, the use of a PP approach for path tracking aims to the minimisation of the disturbances effects in motion, the increase of smoothness of the resulting path and the decrease of the computational load of the system. One of the main similarities in all of these studies is the approximation of a best lookahead distance, either for stability issues or for error minimisation. As also described in the previous Chapter, in order to obtain trajectories more compatible with the dynamics of the fixed wing UAVs, the global path planner must provide straight line waypoint lists with maximized angles between each waypoint-to-waypoint segment of the plan. By omitting sharp turns, the probability of exactly following the planned path is greater since the dynamics of the vehicles are more compatible with paths without sharp turns. In order to further reduce the error in the path following task, this Chapter introduces a method of reducing the fitting

error of the trajectory, always having in mind the reduced computational load of an online framework.

5.2 Path tracking and following

In many modern *COTS* autopilots, the path tracking strategy follows this PP algorithmic strategy: a point P_G forward of the vehicle at distance L in the desired path P is chosen and the controller generates lateral acceleration commands using that reference point as a target (Fig. 5.1a). This acceleration command is given by

$$\alpha = 2 \frac{V^2}{L} \sin \theta \quad (5.1)$$

where V is the speed of the vehicle, L is the lookahead distance and θ is the angle between the vehicles velocity vector and the the vector towards the reference point. This acceleration command is actually equal to the acceleration needed in order to follow a circular segment which crosses P_G , the actual position of the vehicle and is tangent to the vehicle's velocity vector, where its radius R is given by:

$$L = 2R \sin \theta \equiv R = \frac{L}{2 \sin \theta} \quad (5.2)$$

By taking into consideration that with a fixed L at each time step the reference point progresses forward in P as the vehicle approaches, the vehicle slowly converges to P . As also seen in Fig. 5.1b, the direction of L makes a large angle with P when the vehicle is far away from P , while a smaller angles are present when the vehicle approaches P . Also note that if a fixed L distance is used, R becomes shorter when θ increases, which implies a larger circular path and a larger curvature (Fig. 5.2).

This guidance approach follows the work presented by Park et al. in [67]. A linear analysis of this path tracking approach for following straight lines can be done assuming a small angle between the velocity vector and the line which passes through the vehicle being parallel to the desired flight path (Fig. 5.3). Under this assumption, a second order linear model can be derived

$$\ddot{d} + 2\zeta\omega_n\dot{d} + \omega_n^2d = 0, \quad (5.3)$$

where d is the cross track error, with a damping ratio $\zeta = 1/\sqrt{2}$ and a natural frequency of $\omega_n = \sqrt{2}V/L$. Then, the frequency at which the vehicle converges towards P is dependent on the speed and the lookahead distance. In relation with the work presented in this

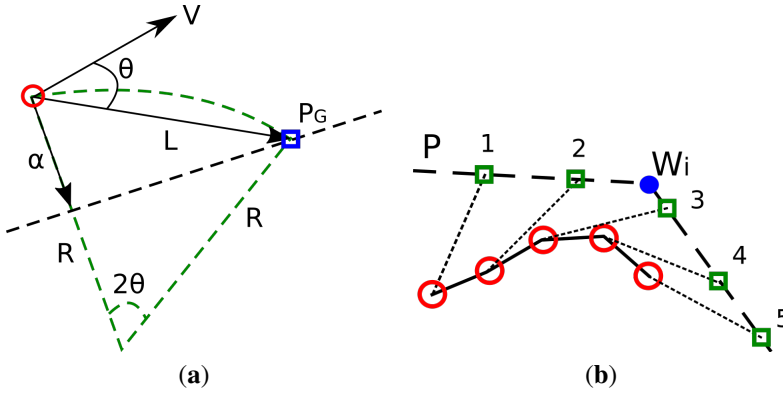


Figure 5.1 In (a), the lateral acceleration command α for a vehicle (red circle), in order to reach a goal point (blue square) in path P to be tracked. Distance L determines the position of the goal point. In (b), the convergence path of the vehicle in different time steps, as approaching waypoint w_i .

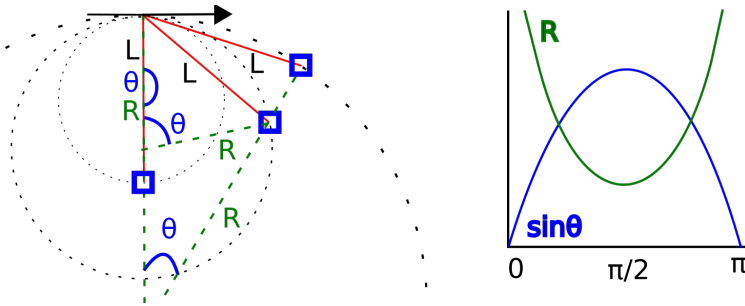


Figure 5.2 If distance L is stable, the circular segment gets bigger providing a sharper turn and bigger curvature.

Thesis, this also implies that the algorithm is designed in a way that does not guarantee an exact passage over the waypoints of a waypoint plan, also seen in Fig. 5.1b.

As Heredia et al. point out in [68], lookahead distance L has a non-dimensional form of $L_s = L/VT$ in order for the trajectory to be stable for straight line and circular paths. If the lookahead distance L is too far then the vehicle might cut corners; if too near, oscillations might occur. By presenting a stability analysis of a fixed L_s value, they state that the system is stable in the straight line path case if

$$L_s > 1, \quad (5.4)$$

where V is the speed and T is the time constant.

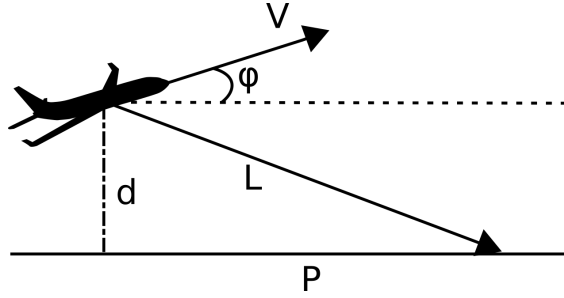


Figure 5.3 The derived linear model as proposed in [67], makes the assumption that angle ϕ is small and negligible. In tight turns after meeting a waypoint, this is not always the case. Distance d is the cross track error and L is the lookahead distance.

Regarding circular paths with γ^{Path} curvature, distance L_s produces stable trajectories if

$$L_s > \sqrt{\frac{2}{1 + \gamma_p^2} + \frac{2}{\gamma_p^2(1 + \gamma_p^2)} - \frac{2}{\gamma_p^2\sqrt{1 + \gamma_p^2}}}, \quad \gamma_p = VT\gamma^{Path} \quad (5.5)$$

where γ_p is the curvature at the goal point, for the circular path case.

This provides a minimum L distance in relation with speed of the vehicle for the straight line path case, and a relation of L and the path curvature γ^{Path} on the goal point, in the circular path case. We can note from Eq. 5.5 that as the curvature gets bigger, the smaller the lower of limit of L_s gets in order for the trajectory to be stable. By providing a minimum L_s in all cases, smaller errors are expected since the convergence to the desired path is faster. However, vehicle dynamics as well as external disturbances and pure delays in the control framework make the calculation of L more complex.

The goal of the proposed method is to apply and extend the aforementioned studies, by providing a framework which will reduce the flight error over the straight line path segments of P . In order to provide a platform and system specific best lookahead distance for an online-onboard path tracking method, an offline-online method is presented in the next Section.

5.3 An offline-online hybrid approach

In the way we have constructed P and treated it as a graph $G = (V, E)$ from the previous Chapters, the path tracking problem resembles cases like the one of Fig. 5.1b. Then, the following constraints are taken into consideration:

- The vehicle has to pass as close as possible over every waypoint of the planned path.
- The overshoot after visiting a waypoint has to be as small as possible, in order to respect the area constraints as described in the previous Chapters.
- The solution has to take into consideration vehicle dynamics and delays of every platform to be used at, especially in *COTS* systems.

In that manner, area A which the UAV will fly over which is not over the desired path, has to be minimised (Fig. 5.4). Consider the total area integral error of the current position $U_c(t)$ of vehicle U to the ideal position $U_r(t)$ on the edge $E_{w,w+1}$ from waypoint w to the next, while T is the duration between the two waypoint visits.

$$\min \int_1^T [(e(t-1) + e(t))h] / 2] dt \quad (5.6)$$

where $e(t) = U_r(t) - U_c(t)$ at time t and h is the height of the trapezoid formed by $U_c(t-1), U_c(t), U_r(t-1), U_r(t)$.

Initially the path tracking algorithm for every straight line segment can utilize a lookahead distance similar to Eq. 5.4. However, preliminary experiments showed that pure delays and latencies make this method not adequate for passing exactly over the waypoints and might produce cut corners and paths like the one in Fig. 5.1b, where the control loop reports goal points 2 and 3 consecutively, even though they belong to a different straight segment. In those areas, the equation of the circular path (Eq. 5.5), even for an instantaneous moment, seem more fit.

The pure pursuit strategies mentioned in the previous Section constantly fit an arc between the vehicles current position and a goal point. These are extended as follows: the algorithm selects a goal point P_G in the path to follow, at a lookahead distance L from the closest point in the desired path with respect to the current position of vehicle U . In order not to proceed to the next straight line segment prematurely, the vector of the straight line path is extended in the same direction until the waypoint is visited.

Then, by combining the two strategies of straight path following and circular path in order to treat delays that might occur as discussed before, for every next turn a different lookahead distance is chosen. Since the size of that angle is correlated with the curvature, lower or higher L_s values can be expected in different turns. This means that L is expected to follow the pattern of R in relation with θ of Eq 5.2 and Fig. 5.2, decreasing on the interval $(0, \pi/2)$ and increasing on the interval $(\pi/2, \pi)$.

The preliminary experiments described in the following Section, validated that indeed a fixed distance does not reduce the previously mentioned area A (Fig. 5.5); the reduction is

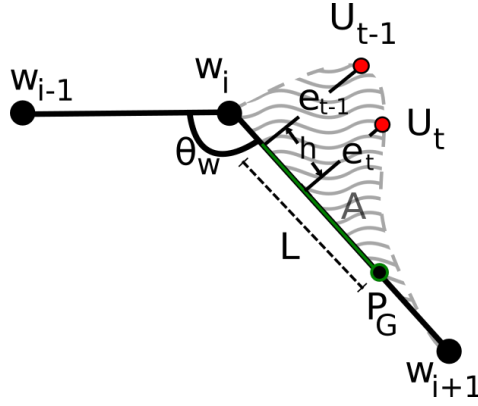


Figure 5.4 A pure pursuit algorithm. In a desired path of a $W = w_1, w_2, \dots, w_i$ set of waypoints, the closest point on the path from the current position of the vehicle U is chosen and a L lookahead distance is used in order to obtain the goal point P_G . The proposed method tries to reduce the shaded area A , using each previous and current position, $e(t-1)$ and $e(t)$ accordingly, for several L values. In the conducted experiments, current vehicle position is obtained 3 times per second.

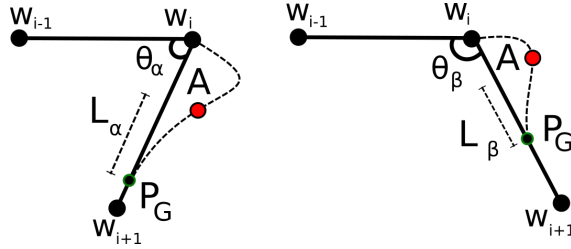


Figure 5.5 The L coefficient which reduces area A is different for different θ angles.

dependent on a dynamic distance coefficient, based on the angle θ_w between two segments of the waypoint plan.

Hence the reduction of area A is dependent on the properties of the underlying UAV model, like its speed and angular velocity, the angle of the turn to be performed and the rate of which the next goal point is given. In that manner, a series of offline simulations are performed, by introducing a method which tests several lookahead distances L against different angles. This procedure produces a lookup table of a distance coefficient L in relation with the angles. Then, during flight, the pure pursuit algorithm uses this table in order to select the appropriate lookahead distance values for every next waypoint. Since this approach is dependent on the underlying vehicle dynamics model, it is characterized as model based - angle dependent.

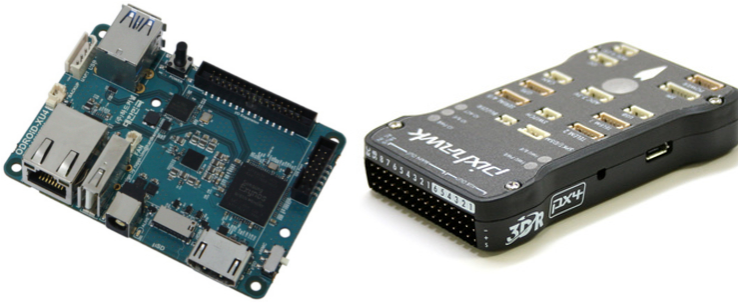


Figure 5.6 On the left, the Odroid XU4 companion computer. On the right the PX4 autopilot.

5.4 Results

As in the previous Chapter, the proposed algorithm has been tested on a simulation framework as also as in field experiments; the simulated setup also permitted the extraction of the aforementioned lookup tables. The same Software In The Loop (SITL) simulation framework of the previous Chapter has been used for simulations, whereas an Odroid-XU4¹ computer has been used as an onboard companion computer for the field experiments, along with a Pixhawk PX4² autopilot (see Fig. 5.6). The onboard computer has the same experimental framework as the simulations, except the visualisation tools. In both cases, the APM Planner ground station software has been used in order to observe the trajectories in real-time (see Fig. 5.7).

5.4.1 Simulations

The UAV model used in the simulated flights is a fixed wing Rascal11 model airplane. Its on-board controller is the open source autopilot Ardupilot and has been combined with the JSBSim flight dynamics model simulator³. The Arduplane controller used is the Pixhawk Flight Management System [59].

Lookahead table extraction

Two sets of simulations have been conducted: the first is intended for checking the validity of the strategy and the extraction of the aforementioned lookup table and has been tested against the onboard autopilot control model, which acts as a trajectory tracker based on

¹ http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825

² <https://pixhawk.org/modules/pixhawk>

³ <http://jsbsim.sourceforge.net/>

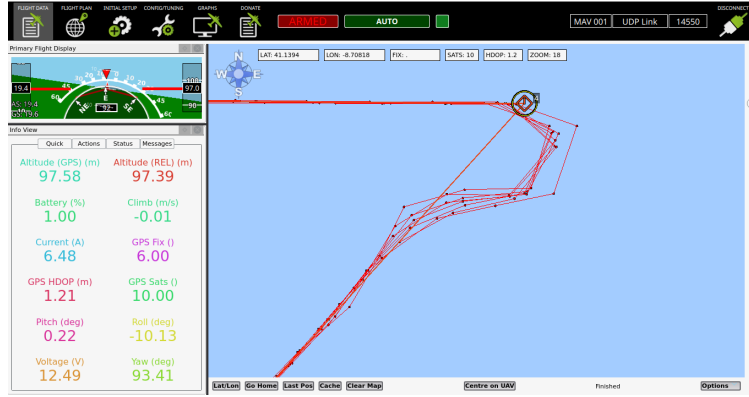


Figure 5.7 An APM Planner 2 screenshot during the simulated flights of the first experiment for the lookahead table extraction, showing different trajectories of the vehicle based on different L values. Note that the outer right trajectory belongs to the control one, without using the algorithm presented in this paper. Also note that the ground control software updates the map every second, hence the rough, orthogonal looking trajectories.

the study presented in [67]. The second is a set of simulated test flights in order to observe the results by using the previously extracted look-up table.

For the calculation of the lookup table, the UAV had a constant platform specific speed of 20m/s. In order to include the expected delays of the experimental framework for the calculation of L , we had to take into consideration the rate at which every new goal point is sent to the controller. By examining the results of Fig.9 in [68], we could expect that a rate of 3Hz (since this is the limit the experimental framework permits us to update the actual position of the vehicle) in giving a new command would result an L_s value of approximately 2,3. Considering that $V = 20m/s$ and $L_s = 2,5$, and since from Eq. 5.4 $L = L_s * VT$ a minimum of 45 meters has been tested as the L value for the lookahead distance.

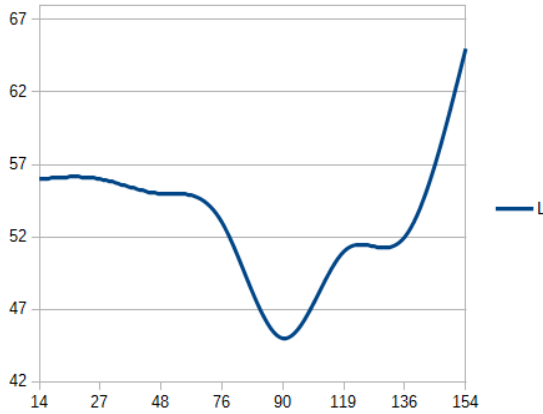
Then, a random incremental set of angles of 14,27,48,76,90,119,136 and 154 degrees has been tested against the proposed method (Table 5.1 and Fig. 5.8). The lookahead distance which reduced the overhead area A for these settings, indicated the lookup distance coefficient value to be used on the respective turns, which was used in the second set of simulated experiments and field tests.

Accordingly, the comparison between using or not the proposed method for the individual turns can be seen in Fig. 5.9

Moreover, the relation between the different values for the lookahead distance coefficient and the tested angles, can be seen in Fig. 5.10.

Table 5.1 Lookup distance coefficient table based on different path angles.

Angle (degrees)	Best lookup distance coefficient L (m)	Area (km^2)
14	55	0,005448
27	56	0,005404
48	55	0,004911
76	53	0,003239
90	45	0,002321
119	51	0,001245
136	52	0,000717
154	65	0,000564

**Figure 5.8** Angle vs L distance.

Simulated test flight

The second simulated experiment is a full scale scenario of an actual area of river Douro in Porto, Portugal, where a UAV flying at 20m/s, performs a coverage task (see Fig. 5.11). Please note that on the indicated path, the average turn angle is of $119,84^\circ$. Also note that there is a specific radius R_v around each waypoint to be considered as visited. A total of nine scenarios have been tested; the first is without the proposed method and by letting the onboard controller to pass by a waypoint on a loose trajectory, meaning that the UAV could pass at a $R_v = 40m$ distance, if this would produce a smoother trajectory. This method is often used in fixed wings in order not to perform intense manoeuvres and it is the default behaviour of the SITL framework. The second case was also without using the proposed method, but restricting the on-board controller that the aforementioned

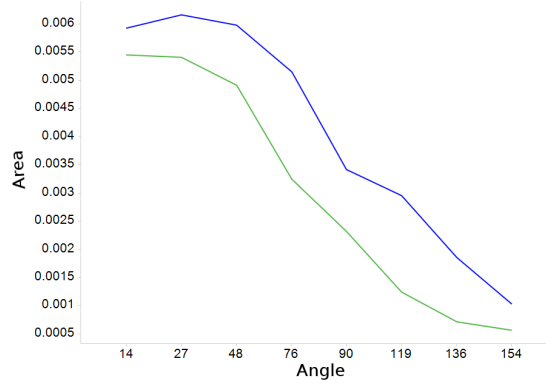


Figure 5.9 The blue line represents the total area outside of the straight line paths by using the autopilot's default path tracking algorithm. The green line is after the proposed method.

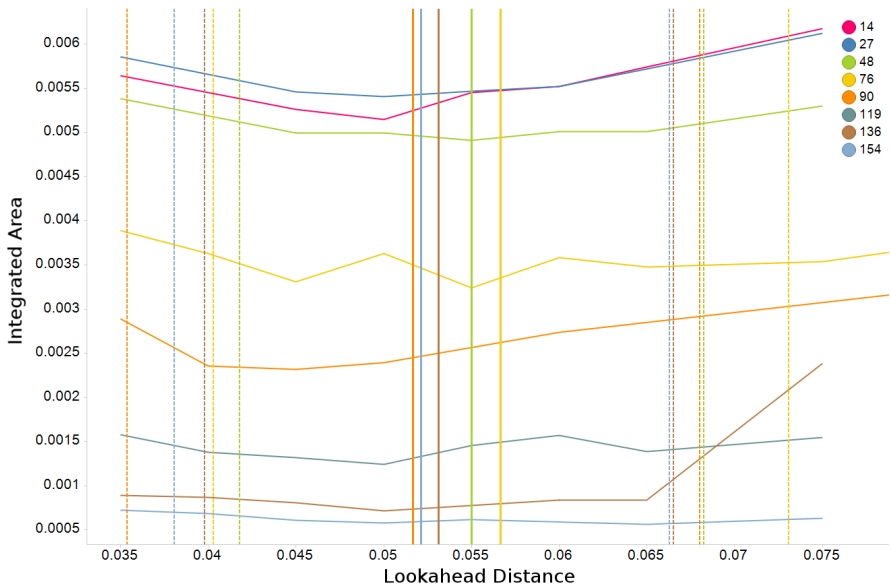


Figure 5.10 The effect of different L lookahead distances (horizontal axis) in relation with the integrated area outside the straight line paths (vertical axis) and the angle of the next segment (individual coloured lines). The vertical lines indicate the average of the lookahead distance, whereas the dotted lines on the left and right indicate the standard deviation.



Figure 5.11 On the left, the estuary of Douro river in Porto; the red lines indicate the borders of the area of interest. On the right, same area as has been decomposed for one UAV, having an initial position on the blue cell. A coverage task is performed by visiting all the waypoints, indicated by the centroids of each cell. The green line indicates this path.

relaxation would have a radius of $R_p = 10m$; a restriction which our proposed framework considers. The next six test cases were performed by using the proposed method, but with a static lookahead distance for all turns, which was extracted from the minimum and maximum best L values of Table 5.1, in a 5 meter step: 40, 45, 50, 55, 60 and 65 meters. Finally, the last scenario is by performing each turn by looking up on the next turn and changing the lookahead distance accordingly, demonstrating the proposed algorithm.

As the results indicate (Figs. 5.12 and 5.13), when imposing the same restrictions to the autopilot, all cases of the proposed method have performed better, while the definitive last case (see case "Adaptive L" in Fig. 5.12) has managed to reduce the overhead area against all other instances.

Another simulated flight has been conducted, this time in the same area and waypoint plan as of Fig. 4.19b of the previous Chapter. The proposed method was tested in two different speed scenarios of 10m/s and 20m/s. Figure 5.14 shows the four trajectories where the default method is tested against the proposed method. The prioritisation of passing over the waypoints with the proposed method is evident, while the area of error was reduced by an average of 18.79%.

5.4.2 Field experiments

The same area of the experiments of the previous Chapter has been used in order to conduct a set of comparative test flights. Two sets of experiments have been conducted, with two different UAVs (see Fig. 5.15). The first set uses the platform already presented in the previous Chapter and extends those experiments, by introducing a tuning control strategy

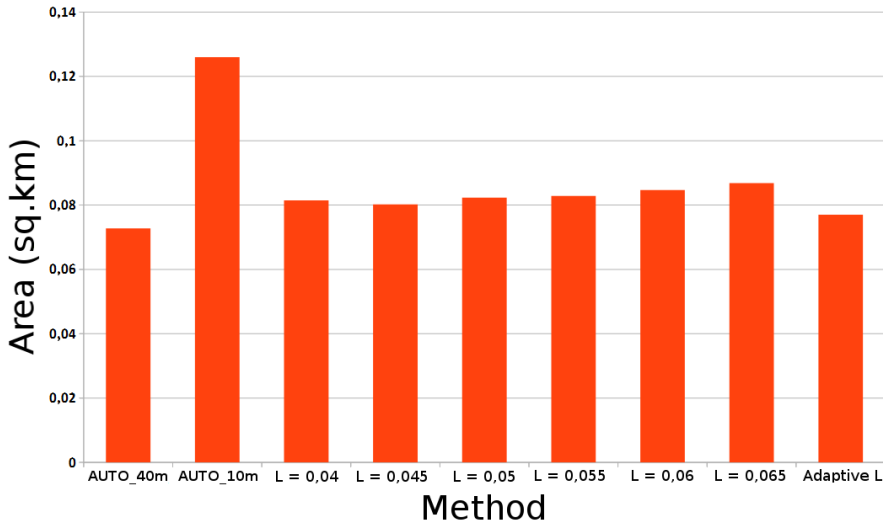


Figure 5.12 The integrated area for each test case of the second simulated experiment. The "AUTO_40m" case refers to the default autopilot behavior, without always passing over the waypoints. On the other hand, the strict "AUTO_10m" case is restricted on passing over them at least 10 metres away. The rightmost "Adaptive L" scenario shows the result by making use of the proposed method, whereas the rest refer to the static L coefficient for all turns.

of the platform. The second set is focused only on the path tracking method in order to compare it with the default one. Table 5.2 shows the characteristics of the field experiment framework

A tuning control strategy

The controller gains of the selected autopilot need to be tuned to meet flight duration and energy consumption requirements. Moreover, the system has to be able to execute its missions in different conditions, both platform-related and ambient-related, e.g. payload or weather. Hence, the performance should be tested in different conditions, with different payloads, and emphasizing the presence of wind disturbances, which may alter the mission performance.

Two more sets of experiments were conducted (M3 and M4) to test the tuning strategy along with the rest of the methods, with M1 (boustrophedon) and M2 (proposed) being from the previous Chapters. The platform tuning was performed by a set of ten flights, by using a sharp turn pattern (M3) of Fig. 5.16.

Then, mission M4 is using the same methods as already proposed, but with a relaxed waypoint acceptance radius, in contrast with strict acceptance radius of mission M2 as

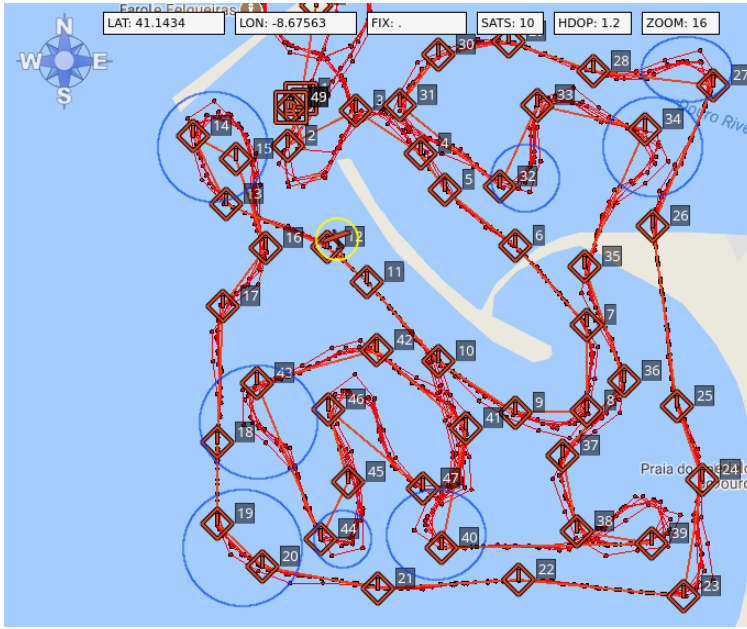


Figure 5.13 As a qualitative result, this groundstation image shows the recorded flight trajectories on different test scenarios. Denoted by the blue circles, some cases where the overshoot of the autopilot is evident.

presented in the previous Chapter.

A tuning baseline is required in order to adjust the control parameters of the platform and is obtained with the aircraft in manual mode, i.e. with no assist, in which a series of manoeuvres are performed to find the desired characteristics with certain throttle $T_i n$, airspeed V_a , climb rate V_h , pitch θ and roll ϕ angular demands. The baseline calibration was obtained in steady conditions with low winds and a temperature of 25°C . The goal is to modify different parameters to achieve the optimal characteristics of the flight. These parameters are: the feed-forward roll gain K_{ff}^ϕ , the proportional (roll-rate) gain K_P^ϕ , the integral (roll trim offset) gain K_I^ϕ , the feed-forward pitch gain K_{ff}^θ , the proportional (pitch rate) gain K_P^θ , the integral (pitch trim offset) gain K_I^θ . To illustrate the attitude control laws that need to be tuned, the roll ϕ control loop is depicted in Fig. 5.17, which is equivalent to the pitch and yaw control laws.

As discussed before, the path tracking strategy of the PX4 autopilot software is the same described in [67]. Its control module has a L_1 controller which is implemented in two stages: one in which a reference waypoint L_{1R} is calculated on the trajectory that has to be followed, and a second one computing the lateral acceleration needed to reach L_{1R} , which

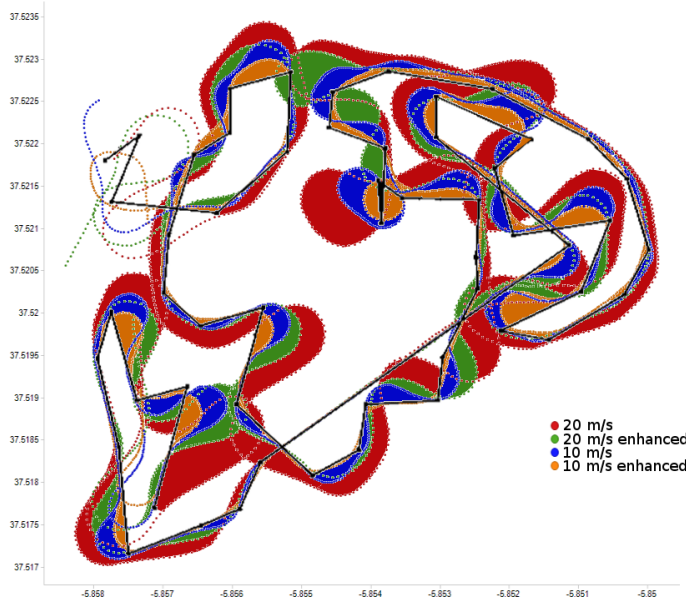


Figure 5.14 Comparison between the onboard tracker trajectory of the autopilot and the proposed method, for two different speed profiles. The error is shaded in each turn. Please note that the overlapped regions show the smaller error first. The black line indicates the waypoint path as sent to the autopilot.



(a)



(b)

Figure 5.15 In (a), the MarineUAS-1 platform used for the coverage and tuning flights. In (b), MarineUAS-2 used for the path tracking experiment.

is equivalent to the roll angle command shown in Fig. 5.17. The main advantage of this method is the reduction of configurable gains, as the Roll PD and cross track errors are replaced with the reference distance L_{1R} . In addition, this function presents improvements in disturbance rejections and circular path tracking in the presence of wind and this also gives the possibility of combining it with a higher level trajectory generation algorithm.

Table 5.2 MarineUAS Platform characteristics.

Component	Characteristic	Value
MarineUAS-1	Material	Expanded Polystyrene (EPO)
MarineUAS-1	Wingspan	2000 mm
MarineUAS-1	Length	1100 mm
MarineUAS-1	Weight	1350 g
MarineUAS-2	Material	Expanded Polystyrene (EPO)
MarineUAS-2	Wingspan	1718 mm
MarineUAS-2	Length	1100 mm
MarineUAS-2	Weight	1050 g
Autopilot	Type	PX4 2.4.8
Autopilot	Processor	32 bit STM32F4V27 (FPU)
Autopilot	Flash Memory	2 MB
Companion Computer	Processor	Samsung Exynos544
Companion Computer	RAM	2GB LPDDR3 SDRAM
Companion Computer	Operating System	Ubuntu 16.04.03
Onboard Software	Framework	ROS Indigo
Onboard Software	Framework	qtnp ROS node
Ground Station	Processor	Intel Core i5
Ground Station	RAM	8GB DDR3L
Ground Station	Operating System	Ubuntu 16.04

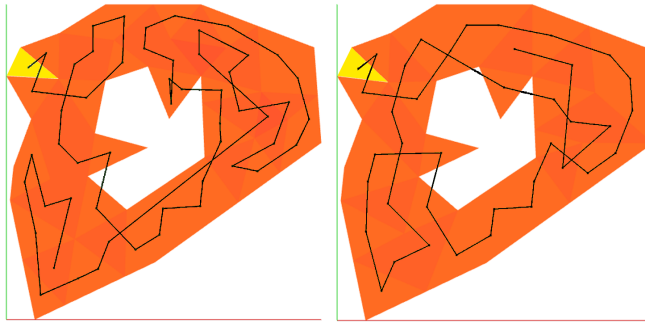


Figure 5.16 On the left, a sharp turn pattern for platform tuning (M3). On the right, the waypoint path (M2) extracted by the proposed methods, as presented in the previous Chapter. This solution was a result of 40 Lloyd iterations and had an inlet sensitivity coefficient I , of 2.35. The yellow cells are the initial positions of the robots.

Therefore, the PX4 L_1 has two parameters that need to be tuned for this process, the L_1 damping, ζ_{L_1} , and the L_1 period ω_{L_1} .

Once the baseline calibration is obtained, the next step is to execute a complex mission,

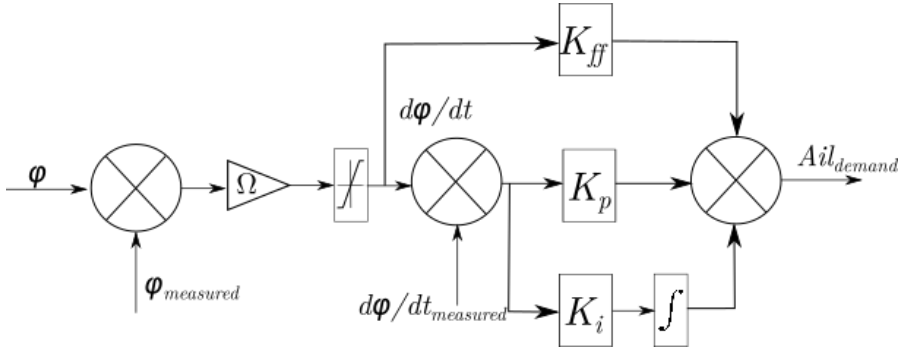


Figure 5.17 The pitch control loop in which $\Omega = 1/\tau$, in which τ is a configurable time constant and $d\phi/dt$ corresponds to the roll rate which is affected by the tuning parameters K_{ff}^ϕ , K_p^ϕ and K_i^ϕ .



Figure 5.18 The waypoint lists used for the tuning experiments is from Fig. 5.16a. The resulting trajectory (M3) can be seen in yellow while the respective sensor coverage in blue shade. Note the sharp turns near the area edges which provide the appropriate conditions for fine-tuning the system.

such as the one presented in Fig. 5.18, to fine-tune the parameters to achieve the adequate performance for sharp turns, disturbance (wind) rejection and power saving.

The tuning mission was executed ten times in different temperature and wind conditions to achieve the desired output. The obtained parameters compared to the baseline calibration are shown in Table 5.3.

Table 5.4 shows the results for every mission, in which a comparison of the conditions

Table 5.3 Tuned Parameters for MarineUAS-1.

Parameter	Baseline Value	Optimized Value
K_{FF}^ϕ	0,20 % rad/s	0,24 % rad/s
K_P^ϕ	0,060 % rad/s	0,055 % rad/s
K_I^ϕ	0,010 % rad	0,024 % rad
K_{FF}^θ	0,32 % rad/s	0,44 % rad/s
K_P^θ	0,032 % rad/s	0,46 % rad/s
Ω_a	2,0 m	0,1 m

Table 5.4 Experimental results in average for the different missions: current discharge rate during mission δ in mAh, the percentage of battery consumption β , wind speed W_s , air temperature T_a in degrees Celsius and flight duration t_f in seconds.

Mission ID	δ (mAh)	β (%)	W_s (m/s)	T_a (°C)	t_f (s)
M1 _{avg}	536,65	0,15	9,23	31,48	529,76
M2 _{avg}	416,717	0,10	8,42	28,50	408,14
M3 _{avg}	552,65	0,21	9,34	33,01	465,40
M4 _{avg}	459,064	0,12	8,49	26,31	414,94

and some important mission parameters are considered. Since M1 has a lower waypoint acceptance radius $\Omega_a = 1,0\text{m}$ there is a slight improvement on energy performance but this may represent a slight impact in the coverage if sharper turns are presented. Temperature also influences on the battery performance, as the average δ is at least 100 mAh higher for those flights performed with an air temperature $T_a > 30^\circ$. On the other hand, wind has a very significant impact of the overall mission performance, in both energy and coverage.

Pure pursuit flight

In the experiments with the second UAV, since only the path tracking method was tested, the flight area was smaller in order to test the behaviour in shorter paths and turns. The two reported trajectories are shown in Fig. 5.19. The vehicle had a constant speed of 15m/s, while the same tables of the second simulated experiment have been used. The error has been reduced in an average of nearly 22% percent in the turns in question. Nevertheless sometimes this error was increased and exaggerated due to the internal programming of the autopilot, which permits visiting nearby waypoints in case they are near before proceeding to the next one on the queue. The actual reduction of error by subtracting these cases dropped to nearly 16%. However we can also notice that the proposed method has managed to also decrease the flight over no fly zones.

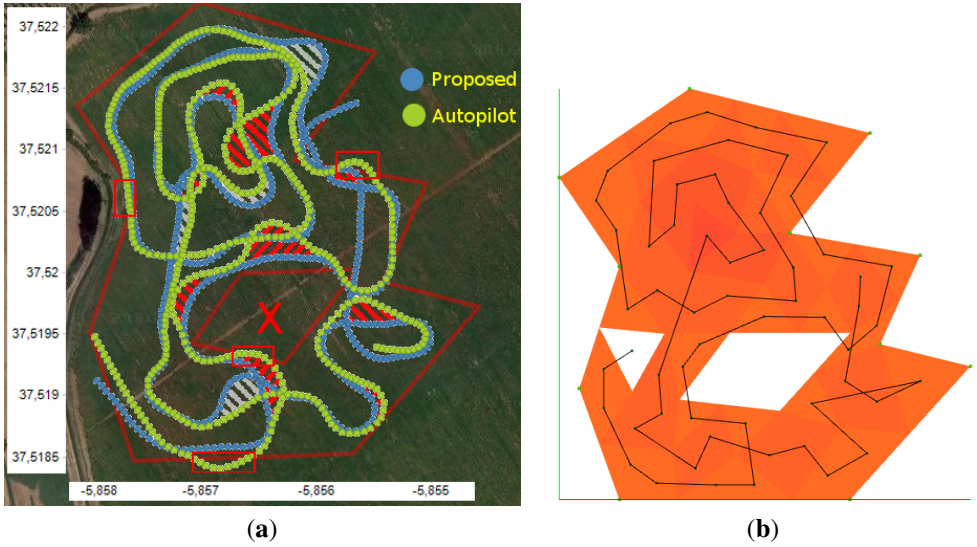


Figure 5.19 In (a), the blue trajectory indicates the proposed method whereas the green one indicates performed trajectory by the onboard PX4 autopilot. The red "X" indicates the no-fly zone and the red rectangles indicate the trajectories outside the ROI by the autopilot. The red shaded areas indicate the error overshoot by the autopilot, while the grey ones indicate the negative error, where the autopilot cuts corners. Please note that in some cases the autopilot created a trajectory in order to visit nearby waypoints. These errors were not calculated in the reduction of flight error with the proposed method, which reached nearly 16%. In (b), the selected area, triangulated with a straight line coverage path according with the previous Chapters.

5.5 Conclusions

The results show that the proposed framework aids the process of reducing the overall flight path length and time, while reducing flight over no-fly zones and supporting the on-line safe separation between the aerial robots, in comparison with classical boustrophedon methods. On the other hand, it manages to reduce the flight error of the planned trajectories in comparison with state of the art on-board controllers in a hybrid on-line/off-line manner, decoupling the control model with the desired trajectories, creating in such a way a uniform method which can be applied to any team of heterogeneous aerial robots.

The obtained results from the experimental tests of the tuning strategy, present consistent measurements, as standard deviations were reduced, e.g. the standard deviation $\sigma(t_f)$ of flight time was $\sigma(t_f) = 20,73s$ for M3 and $\sigma(t_f) = 29,19s$ for M2 between missions, to very low values, e.g. $\sigma(t_f) = 3,96s$ for M4 and $\sigma(t_f) = 7,35s$ for M1. This dispersion

was consistent for every other performance measurements, showing that the disturbance rejection and performance capabilities are sufficient for the coverage missions.

The results show that the proposed method further aids the process of reducing the flight over areas of no interest or of a flight path which intersects restricted zones. Moreover, the creation of lookup tables for different vehicle models and speed profiles might be useful in the opposite direction; by having a trajectory, a specific model can be identified. Also, the computational complexity of obtaining safer trajectories for a predefined path is reduced, as the only operation that the vehicles have to perform is to choose an adaptive lookahead distance for their next turn. The results also support that this selection could not only be chosen dynamically but as a static best distance, as an average of the produced table. Since the difference between the two alternatives does not produce a significant difference, the selection of either is application dependent; a finer and dynamic tuning of the distance coefficient can manage to reduce the integrated error.

6 A decentralized framework for coastal missions

So long, and thanks for all the fish

DOUGLAS ADAMS

By using a *Team Of Robots (TOR)* in coverage tasks, several advantages can be identified. Initially, the system has a critical increase in overall robustness, since it can account for robot losses. This robustness can also be expressed by the information redundancy in information exchange between the robots, either by combining the same kind of information for a large area, or by performing a multi-layered data fusion by using different sensors. A team of aerial vehicles achieves a better overall performance since the time to complete a task is shared between them. Furthermore and since our work is mainly focused on small low cost UAVs, the overall cost is reduced, in comparison with a large expensive vehicle or a manned aircraft; the latter is also a consideration in the context of operator safety. Finally, a TOR has a better spacial distribution in comparison with a single vehicle, especially in cases where the data to be gathered are sparse and distributed in a large area.

In the context of the MarineUAS project, the methods described in the previous Chapters can be organised in a top down framework architecture of path enhancement techniques for each robot. Since the proposed framework is an on-board, computationally non-expensive solution, it can be extended in a decentralised manner in order to provide a robust, fail-prone and flexible low cost *COTS* infrastructure for teams of UAVs. By applying a market-based

architecture for task assignment, each of the UAVs can change roles and serve as data relays or as data acquisition agents. Then, a multi-layered architecture is proposed by providing a test case of a buoy searching team of UAVs, as described in the previous Chapters. This Chapter presents the complete architecture for such a scenario, where the goal is to locate the positions of the buoys while maintaining connectivity. Moreover, the TOR is able to be reconfigured in terms of *ROI* and roles in case it is needed.

The decomposition, partition, waypoint extraction and path tracking enhancement methods presented in the previous Chapters as a top-to-bottom architecture of path enhancement modules is presented in Fig. 6.1. In each level of this organisation, the following modules can be identified:

- Complex area decomposition based on the FoV of the robots.
- Partitioning for an arbitrary number of robots, based on their relative flight time autonomy capabilities, along with feasible Lloyd smoothing procedures on the resulting ROI.
- Path enhancement through an enhanced waypoint list extraction, in terms of path length and overall turn angle.
- A pure pursuit online path tracker which guarantees the passage over the waypoints while reducing the flight error.
- A tuning control strategy which performs a COTS platform parameterization, as described in the field experiments of the previous Chapter.

This framework is platform independent and applied on-board each UAV, providing a team-wide homogeneity in respect with the communication protocols and world representation. As described in the next Section, depending on the team set-up, each robot has a global knowledge of the map partition and local knowledge of its trajectory, position and gathered information. Then, depending on the evolution of the tasks, the team can be reconfigured.

6.1 A market based decentralized architecture

In centralised approaches for multi-robot task assignment, a single robot or a *coordinator* is used in order to assign tasks to the rest of the team. In the UAS case, this role is usually held by a ground station inside the communication range of every vehicle or through a message-hop schema, where information and instructions are propagated through the network of vehicles. This approach can result in optimal solutions since the coordinator has global

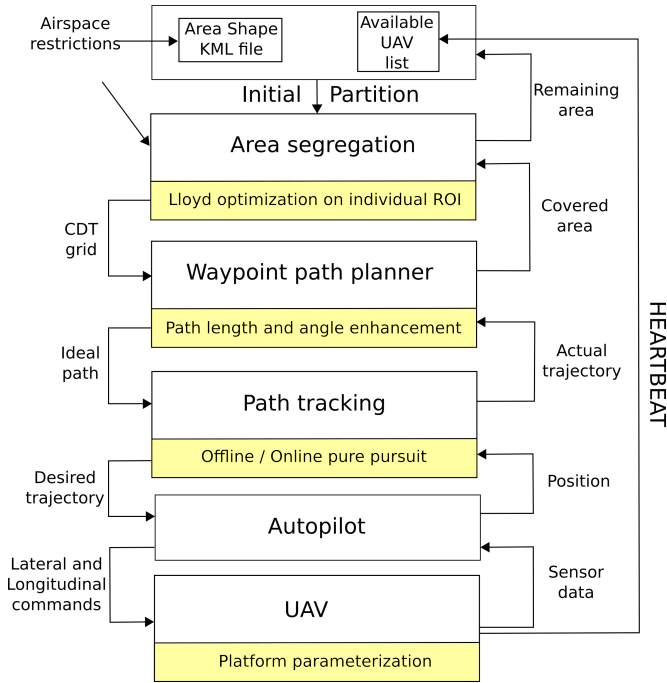


Figure 6.1 The top-to-bottom proposed framework architecture, with every path enhancement module denoted in yellow. Each operation is performed onboard each robot, thus providing an initial step for a decentralized solution. Please note that the airspace restrictions can be communicated to each individual vehicle, which will then reconfigure its ROI.

knowledge of the vehicles' states as also as any other relative information. Moreover, if the coordinator's task is solely the planning of task assignment, the computational burden of each of the vehicles is solely on the mission execution. Nevertheless, centralised architectures are not robust, since they possess the single-point-of-failure disadvantage. Moreover, and especially in teams of aerial robots with limited communication range operating in large areas, the communication constraints imposed are restrictive, leading to critical latencies.

On the other hand, in distributed approaches only local knowledge is available for the vehicles, which is exchanged with other vehicles in specific intervals or depending on their proximity. During those periods, decisions are made based on that information which may alter the task assignment, based on specific local and global utility functions. These approaches are more robust since there is no single-point-of-failure and can be characterised by an increased flexibility since every vehicle can be its own and its neighbours coordinator.

Nevertheless this also implies that the robots have an increased computational burden of planning the feasibility of these complex tasks and since they only possess local information, the overall solution can be sub-optimal.

Considering the taxonomy proposed by Gerkey et al. [69] for task allocation in multi-robot systems, a categorisation is proposed based on three categories:

- Task type: **Single Robot** or **Multi-Robot**, distinguishing the tasks that need one or more robots to be completed.
- Robot type: **Single Task** or **Multi-Task**, distinguishing types of robots which can perform only one or multiple tasks simultaneously.
- Schedule type: **Instantaneous Assignment** or **Time-extended Assignment**, differentiating the ability to provide to the system information only for the current plan or to plan for future task allocations too.

The authors point out that only the SR-ST-IA problem, which is an instance of the optimal assignment problem in combinatorial optimization, can be solved in polynomial time, while the rest of the combinations are strongly NP-hard. In our case and considering the approaches already presented in the previous Chapters, we can identify each UAV task of area coverage as an SR-ST-IA case. Each sub-area is a task which is assigned to a single robot and each robot can be assigned to this task only. Regarding scheduling, current task assignment can be altered only if new information is provided and no future scheduling is provided. Please note that this categorisation is assumed if, as noted, each task is the sub-area coverage. If the notion of task is defined as the visit of every waypoint, then the problem changes to SR-ST-TA, since a serial waypoint list has to be constructed, dictating in that way the future scheduling of single waypoint visit task for every robot.

Following the aforementioned taxonomy of Gerkey et al. [69], Korsah et al. [70] proposed an extension in order to include interrelated utilities and task constraints. This taxonomy named *iTax*, has the following categorisation:

- **No Dependencies**: Problems which each sub-task for a robot is not affected by other tasks or robots.
- **In-schedule Dependencies**: The utility function of a robot for a task is affected by the other tasks this robot has to perform or is performing.
- **Cross-schedule Dependencies**: The utility function of a robot for a task is affected by the other tasks this robot has as also as the schedules of other robots.

- **Complex Dependencies:** The utility function of a robot is affected by the schedules of other robots for that **specific** task decomposition, thus not decoupling the task decomposition and task allocation problems.

As before, only the first category falls in the polynomial time/linear assignment problems. Our case of UAV-area assignment also falls into this category, since each sub-area and task of area coverage is not affected by the schedules of other agents. Once more, if we consider as a task each individual waypoint visit, then the categorisation is that of **CD**, since each partition solution results in different utility values. However, we have already treated this task decomposition and allocation in the previous steps. Nevertheless, this categorisation is ideal and in order to meet the aforementioned decentralized robustness and flexibility, some properties have to be identified, which will provide the system several fail-prone characteristics.

Initially, it is assumed that the robots will not always be in a communication range to provide uninterrupted data exchange. Moreover, UAV loss mitigation is considered; this requires methods which will perform team reconfiguration and area reassignment operations. These methods also aid sub-area alterations, like the appearance of a new no-fly zone. This would change the utility function of a robot since it would have to cover less area, thus affecting the global utility function and optimality of the original partition.

Following the definition of a market based architecture based on Diaz et al. [71], the aforementioned taxonomies, categorisation and pros and cons of the centralised and distributed solutions, a hybrid centralised/decentralised market-based architecture is proposed, which has the following characteristics:

- The TOR has a global objective function of obtaining task related information through complete area coverage. This is achieved by performing a decomposition into sub-area individual tasks, depending on the number of robots and area restrictions.
- The global utility function for the TOR, or for the auctioneer in a centralised market-based architecture, is the maximisation of the information retrieval by utilising the resources of the vehicles. This is achieved by the capability aware partitioning algorithms as described in the previous Chapters. Like so, this function is dependent on the relative capabilities of the vehicles and their initial positions. It is also dependent on the number of the vehicles and the total area to be covered.
- The local utility function for every vehicle is the minimisation of the path length and maximisation of the average path angle as described in the previous Chapters.

- The local utility function of every vehicle might change in case a new no-fly zone appears or the energy needs become higher, thus decreasing its relative capability with respect to the other vehicles.
- The latter characteristic, dictates the need for a specific resources/needs schema, where the global and local utility functions affect each other.
- This need requests of a mechanism which will be responsible of obtaining the individual utility function results and try to maximize its own utility, through a fail-prone communication schema.

As such, initially a *coordinator* is chosen which performs the aforementioned initial partition of the whole area. This coordinator serves as an *auctioneer*, which is responsible of obtaining local information and compute an outcome which maximizes the global utility. Then, a team of communication relay vehicles is deployed in order to serve as a relay between the coverage UAVs and the coordinator. These relay vehicles configure their trajectory in order to communicate with the coverage vehicles, obtain local information and perform the communication with the auctioneer. The whole team of UAVs, relays and coverage, should be able to reconfigure depending on several user or mission criteria. These characteristics and methods are demonstrated in the next Section, where a test case of data acquisition from sea buoys in a marine biology scenario is required.

6.2 Test case: a multi-layer framework for buoy coordinated data acquisition

Following the MarineUAS objectives, bringing attention to the challenging problem of using UAVs for marine sciences, a buoy data extraction scenario is chosen. Figure 6.2 shows the proposed 3-layer architecture for a *Coordinated Data Acquisition (CDA)* model: in the top layer, a coordinator runs the initial partitioning and waypoint extraction algorithms and monitors the whole operation. The middle layer is responsible for the communication between the UAVs and the coordinator using a feedback control schema, described in the following Sections. Finally, in the bottom layer, the individual sub-area UAVs are responsible for data extraction from the buoys. Please note that the UAVs used in either of the last two levels can be of the same type or not, but are denoted differently, based on their role: the buoy searching UAVs (UAV_{Search}), which search for the buoys and communicate with them and the UAV communication relays (UAV_{Comm}) which are used to enable communication between the searching UAVs and the coordinator.

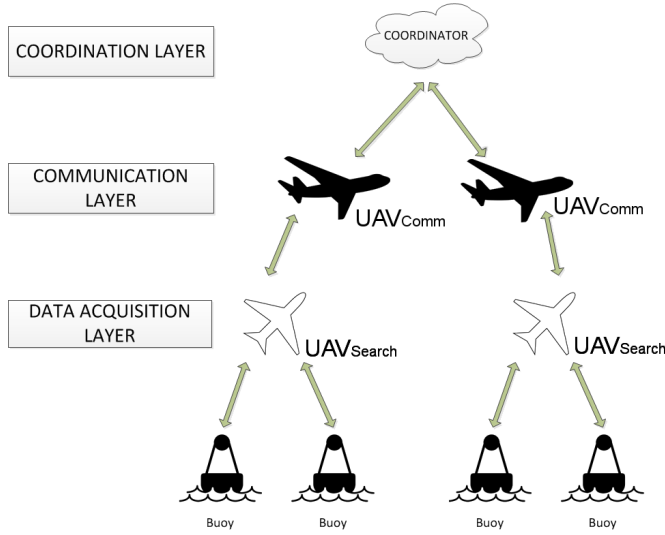


Figure 6.2 A multi-layered framework for buoy data acquisition.

Following the market-based paradigm as discussed before, each of the layers has to satisfy its own utility function: the coordinator has to make sure a fair partition is performed for every UAV_{Search} . The relay UAVs have to maintain the connectivity between the coordinator and the searching UAVs. Finally, the searching UAVs have to perform their coverage tasks, according with the path enhancement framework of Fig. 6.1.

The coordinator and search UAV functions have been described before, while the way the UAV_{Comm} manages to satisfy its tasks, is described in the following sub-sections.

6.2.1 Data acquisition layer and buoy specifications

Sea buoys are used in oceanic studies as a low consumption / large autonomy devices in order to gather data from either sea surface or from a specific depth. They are also used as data relays for underwater sensors or vehicles. When a buoy radio-device is active and waiting to receive or transmit data, it wastes energy on idle listening. Since traffic loads are usually low in buoy networks, such idle listening can use a considerable amount of energy unless efficient communication mechanisms are employed. Therefore, in order to save energy, buoys do not transmit data through the radio device continuously, but by utilising an interrogation technique. Common low-power RF devices periodically wake-up to check if another device is trying to make contact. However, a new type of low power device became available which only wakes-up when it is explicitly interrogated. In order to perform this interrogation, the UAV RF extraction transmitter has to send a RF signal to

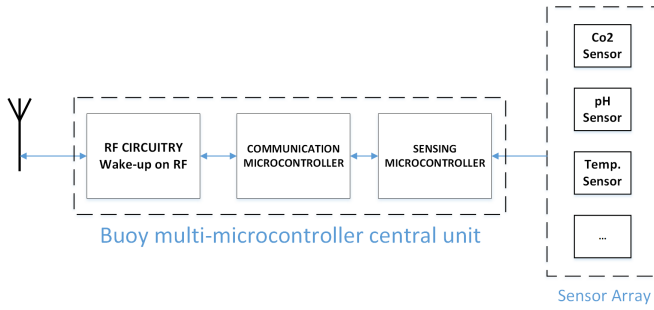


Figure 6.3 Multiprocessor and wake-up RF circuitry structure.

wake-up the buoy radio device in order to start the extraction. This wake-up process is performed by using wake detectors like the ones in [72] and [73]. These are simple circuits which include two nano-powered operational amplifier comparators in conjunction with a super regenerative RF receiver which wakes up the communication micro-controller from its low-consumption, deep sleep state. The buoy's embedded central unit structure is shown in Fig. 6.3.

In order to make contact and extract information from a given buoy (B_n) a UAV_{Search} has to be close enough to enable a wake-up, handshake and data transmission sequence. As described in Chapter 2 and shown in Fig. 2.9, the on-range area RA to enable transmission is determined by the transceiver specifications. In the given test case, the location of the buoys is unknown and the initial position of every UAV_{Search} is randomly generated. Then, for each of the searching UAVs, the area is decomposed, partitioned and coverage waypoint plans are extracted according with the methods described in the previous Chapters. An example is shown in Fig. 6.4.

When a UAV locates a buoy, executes loitering manoeuvres around the buoy in order to gather the data. Once the transmission is finished UAV_{Search} goes back to the original plan. Then, the UAV_{Comm} team is used to recover the data.

6.2.2 Communication layer as a feedback communication control model

This middle layer demonstrates a *Feedback Communication Control (FCC)* model, which actuates on a UAV when it has the role of a UAV_{Comm} . This control model actually replaces the waypoint path planner and path tracker of the proposed framework shown in Fig. 6.1. It is in charge of establishing a communication link between one or more UAV_{Comm} , multiple UAV_{Search} and the *Coordinator*. Depending on the restrictions which the aerial platforms will impose, the goal is to enable communications with a given desired *Quality of Service (QoS)* level, taking into consideration the energy consumption, number of vehicles, link

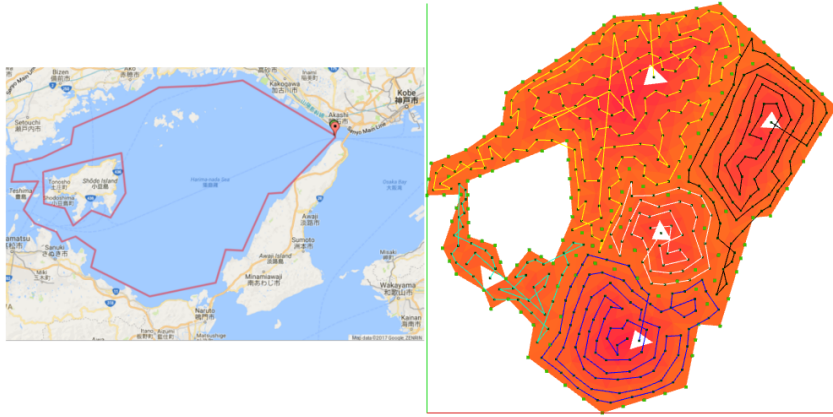


Figure 6.4 A buoy searching CDT example of a team of 5 UAV_{Search} searching vehicles, in Kobe, Japan. Each of the vehicles has different flight duration capabilities, while the communication cone of each buoy is the same, resulting in same size triangular cells for the decomposition.

bandwidth and time. This involves control and real time scheduling as in [74], where the UAV_{Comm} , has the role of a relay node to enable communication. The feedback strategy employed in this layer has the dual task of commanding and optimizing the UAV_{Comm} motion, adapting the scheduler algorithm according to the desired QoS level. This strategy allows the controller to correct the discrepancies between a real-time simplified estimation and actual flying and network conditions. This model, presented in [75], converts the user requirements given by a QoS selection to motion instructions for the unmanned vehicles. First, the required network variables and the physical constraints serve as input variables for the controller. Then, periodically a set of estimated variables are calculated to predict the behaviour and create the UAV paths. After that, the actuation output variables are sent to the UAVs. The variables used are summarized in Fig. 6.5a while Fig. 6.5b shows the proposed FCC model.

Once again, please note that in case a UAV takes the role of the UAV_{Comm} , then the aforementioned FCC model replaces the waypoint path planner and tracker modules of the framework shown in Fig. 6.1. In that manner, the flexibility of the system remains, as every UAV can take up any role, as described in the next Sections.

6.2.3 Single UAV dynamic role reconfiguration

The dynamic nature and flexibility of the chosen architecture enables the robustness of the system against unpredicted conditions such additional energy consumption due to changing weather conditions. Moreover, the uncertainty of buoy positions may affect the

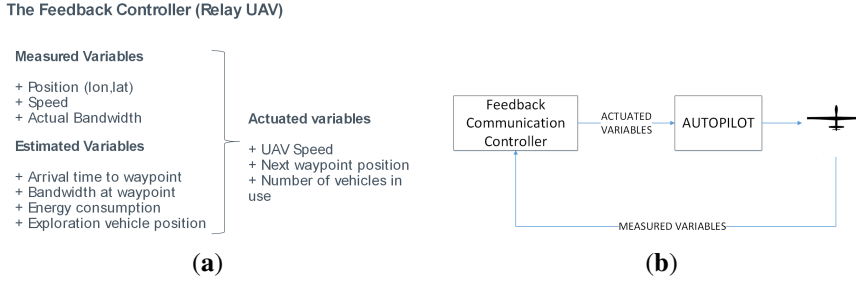


Figure 6.5 In (a), the FCC model variables and in (b), the FCC model.

overall performance of the system and the deployment of the communication network. To tackle this situation in real-time, in addition to the feedback controller, the proposed market-based strategy evaluates the objective functions of each UAV and two strategies have been implemented to dynamically adapt the roles of the deployed UAVs. The first one is the role reassignment which allows a UAV_{Search} to act as an UAV_{Comm} and vice versa. In this way, the architecture is able to change the number of UAV_{Search} as the buoy locations are being discovered and the distances between vehicles increase. On the other hand, in order to balance the communication network load, meet the QoS requirements and avoid congestion issues, the proposed architecture is able to recruit idle UAVs and convert UAV_{Search} into UAV_{Comm} .

Figure 6.6 shows the conversion diagram divided into the two steps:

- Converting UAV_{Search} into a UAV_{Comm} . During the mission, the required network quality through the QoS scheme could not be achieved and more UAV_{Comm} may be needed, so the main trigger for this conversion is based on a network-event. This forces the reconfiguration for the whole team and a repartition of the remaining area into new sub-areas. This operation also occurs when one or more searching vehicles are missing.
- Converting UAV_{Comm} into a UAV_{Search} . This is triggered based on events like a loss of a UAV_{Search} vehicle or heavy weather conditions which make the searching mission unfeasible.

In both cases, the number of available searching vehicles changes. In the context of the market-based architecture, what changes is the available global resources for the searching task. Like so, the auctioneer, or the coordinator, has to provide the new areas or sub-tasks to each of the remaining vehicles. This process resembles the bidding process, where

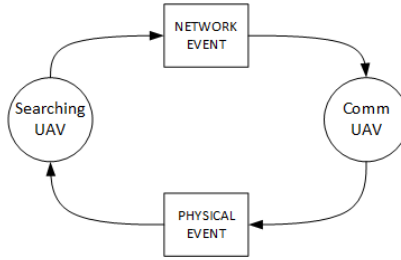


Figure 6.6 Role reassignment diagram, based on events.

every vehicle bids according to its current status: its position, the area that has already covered, its remaining relative coverage capability etc.

Dynamic team reconfiguration

The second strategy is the implementation of a decentralized team reconfiguration solution which increases the robustness of the system. It may occur that a group of UAV_{Search} and/or UAV_{Comm} lost connectivity with the *Coordinator* during the searching mission. It may also happen that during that time one or more of the isolated vehicles becomes inoperative. This scenario is shown in Fig. 6.7. After a period of isolation from the *Coordinator*, the UAV_{Search} and the UAV_{Comm} vehicles that are in the same area and are able to communicate between them, execute a regrouping algorithm. The algorithm workflow is as follows,

1. **A local coordinator is chosen.** Every vehicle acting as a network node and depending on its dynamic and communication properties, has a pre-assigned rank. When a disconnection from the *Coordinator* is detected by any vehicle, it starts to share data packages with its neighbours, propagating the rank of all vehicles. The behaviour is similar to the token-holder selection in token-ring local networks. After waiting a predefined time, the higher rank vehicle is selected as a local coordinator and it is the group leader ($UAV_{GLeader}$).
2. **Vehicle enumeration.** The $UAV_{GLeader}$ starts to enumerate the available vehicles and the local objectives like the remaining area to search or the acquired buoys. Since the coordination capabilities of $UAV_{GLeader}$ may be weaker than the main *Coordinator* in a given scenario, the motion control of the vehicles is performed by using self-triggered and event-triggered controllers. The self-triggered controller allows to perform the mission while decreasing the amount of data shared between vehicles as in [76].
3. **Regrouping.** It may occurs that one vehicle is again able to establish connection with the main group. In this scenario all communication relay vehicles (UAV_{Comm}) must

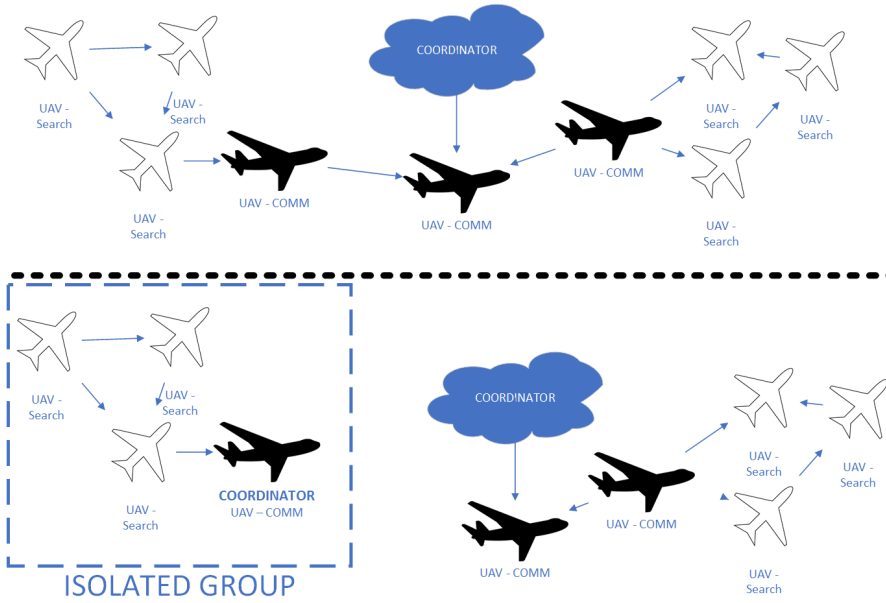


Figure 6.7 Team reconfiguration when a group is isolated.

move towards the main group but keeping the connectivity between them, otherwise a disconnection event may occur, going back to step 1. To resolve this issue, a time coordinated path following (TCPF) controller is implemented, with a method to reduce the frequency of information exchange between the vehicles, through the use of a self-triggered control strategy. Exploiting this architecture, it is possible to plan the motion of the UAV_{Comm} platoon keeping the connectivity between them.

The latter is performed by setting a set of QoS-based waypoints along with a desired temporal constraints. The self-triggered time coordinated path following (STCPF) control strategy makes use of the self-trigger results in [77] and the time-critical coordination ideas in [78] and [79].

6.3 Experimental setup

In order to obtain quantitative data about the parts of the aforementioned test case which have not been presented in the previous Chapters, in the following subsections we present the considered setup for the buoys and the communication devices. Then, the dynamic model for the relay communication vehicles is presented, along with the energy and throughput estimation models.

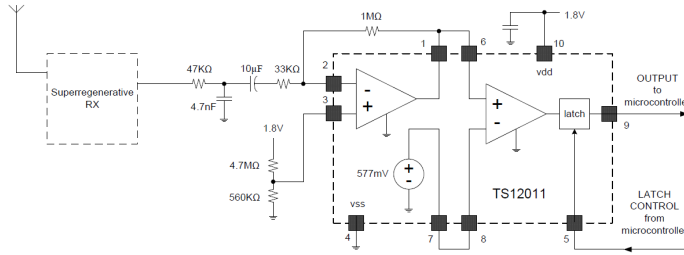


Figure 6.8 Low-power RF wake-up circuit.

6.3.1 Buoy

The schematic of the electronic part of the buoy communicator is presented in Fig. 6.8. It uses operational amplifiers to build an analog comparator specifically designed to consume less than $1.5\mu\text{A}$ in sleep mode. The comparator is in charge of continuously checking for incoming RF signals but consuming less power than a complete RF receiver. Once the signal is received, the comparator wakes-up the microcontroller and the main transceiver which makes a connection with the UAV_{Search} . The transceiver used is a Texas Instruments CC1101 transmitting at 915 MHz in order to avoid interferences over the 802.11 b/g 2.4 GHz devices that are used to create the UAV communication network. Fig. 6.9 shows the power received when using a pair of CC1101, one in a 3D printed buoy in the water and the other in a multirotor flying at 20 meters height. The receiver sensitivity is -94 dBm, so approximately 300 meters is the maximum distance to establish a connection and transmit data with 0dB antennas at 256 Kbaud. Using this data is possible to make a simple estimation of the available bandwidth depending on the distance, as in the work in [80].

6.3.2 Dynamic model for the relay UAVs

The dynamic model used for the communication relay UAVs is the same as presented in [81]. The definition of this model is presented here since its path generation is not achieved by the methods presented in the previous Chapters, but by using the FCC as described before. It uses a 3D point mass model so that the six degrees-of-freedom (DOF) motion equations can be expressed as:

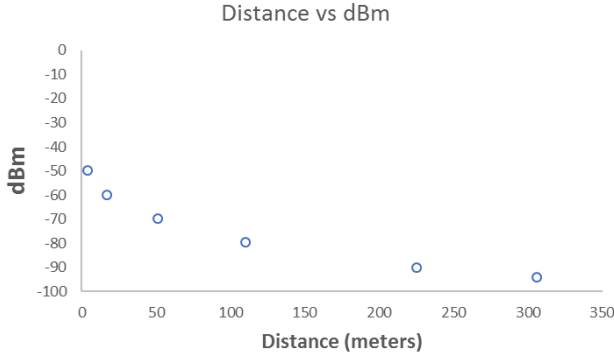


Figure 6.9 Power received vs Distance using a pair of Texas CC1101 sub-GHz transceivers.

$$\begin{aligned}
 \dot{x}_A &= v_A \cos \psi_A \cos \gamma_A, \\
 \dot{y}_A &= v_A \sin \psi_A \cos \gamma_A, \\
 \dot{z}_A &= v_A \sin \gamma_A, \\
 \dot{v}_A &= \frac{u_1 - D}{m} - g \sin \gamma_A, \\
 \dot{\gamma}_A &= \frac{1}{v_A} \left(\frac{L \cos \phi_A}{m} - g \cos \gamma_A \right), \\
 \dot{\psi}_A &= \frac{1}{v_A \cos \gamma_A} \left(\frac{L \sin \phi_A}{m} \right), \\
 \dot{\phi}_A &= u_2
 \end{aligned} \tag{6.1}$$

where (x_A, y_A, z_A) is the UAV position in the inertial frame, v_A is the UAV airspeed, ψ_A is the UAV flight path angle, γ_A is the UAV heading angle, ϕ_A is the UAV roll angle, T is the thrust, D is the drag force, L is the Lift force, m is the mass of the UAV, g is the Gravitational acceleration, ρ is the air density, S is the UAV reference area, C_L is the lift coefficient, C_D is the drag coefficient, CD_0 is the drag coefficient at zero lift and $k_{D/L}$ is the induced drag factor. In addition to that, the L , D and C_D values used can be seen in Table 6.1.

Please note that we actuate over the thrust of the vehicle to affect the airspeed of the UAV as input u_1 , while input u_2 is the roll rate which also affects the heading.

Table 6.1 Lift, Drag and Drag Coefficient Values used in Simulations.

Parameter	Value
L	$\frac{1}{2}\rho v_A^2 SC_L$
D	$\frac{1}{2}\rho v_A^2 SC_D$
C_D	$C_{D_0} + K_{D/L} C_L^2$

6.3.3 Energy estimation

Neglecting electronic equipment, energy consumption comes from thrust generation. Thus, energy consumption at a given moment depends on the UAV velocity. The FCC module is able to regulate the energy consumption by actuating on the velocity. One of the key objectives of this work is to achieve a certain level of energy QoS not knowing much about specific UAV flight parameters in advance.

The energy is estimated as follows: From [82] we can obtain the total drag force for an airfoil

$$D = C_D \frac{1}{2} \rho V^2 S, \quad (6.2)$$

where D is the total drag force, C_D is the drag coefficient, ρ is the air density, V is the airspeed speed over the airfoil and S is the airfoil area. It is correlated to the power consumption and the UAV velocity. The propeller efficiency that depends also on the airspeed is neglected. Thus, Except for V , we assume no changes on the rest of the variables in a UAV during flight at the same altitude.

The FCC module actuates over V . The changes on the velocity input affects the drag, and the relation between drag and speed can be seen in Fig. 6.10. Therefore, defining power consumption as P , a relation between the power consumption and velocity can be obtained like so

$$P = \frac{\delta W}{dt} = \frac{F \delta r}{dt} = FV = \frac{1}{2} \rho SC_D V^3 \quad (6.3)$$

Absolute energy consumption is difficult to calculate without the airfoil, air density and other aircraft parameters but it always grows with the third power of the velocity. However, it is possible to obtain a relative power consumption value based on the difference between the cruise speed and the actual speed. In this way, it is possible to bound the energy consumption and introduce it as a QoS index.

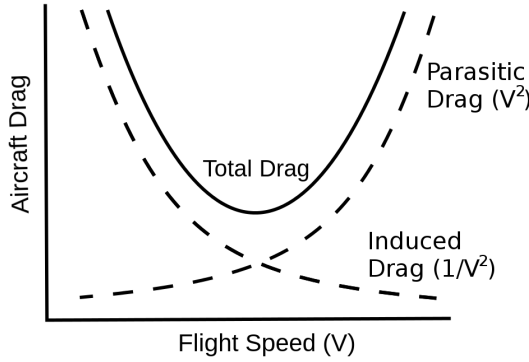


Figure 6.10 Total drag vs air speed.

6.3.4 Throughput estimation

Accurate throughput estimation is a highly computational difficult task that depends on many external factors such as the specific scenario, antenna selection, UAV attitude and altitude, weather conditions, etc. To avoid high computational cost operations we use a fast way to obtain the estimated throughput that is later corrected by the FCC system using actual measured data.

For simulation purposes we used the information from the router TP-LINK WN-722 2.4Ghz IEEE802.11b/g/n device as the coordinator node and a similar USB device for end nodes. To obtain a roughly estimation we need to know at least the device power transmission P_{TX} , the receiver sensitivity R_X , both receiver and transmitter antenna gains G_{TX} , G_{RX} and the modulation used. This variables could be obtained from the device datasheet.

Initially, by using the Friis equation

$$L = 20 \log_{10} \left(\frac{4\pi d}{\lambda} \right), \quad (6.4)$$

where L is the power attenuation in dB, λ is the wavelength and d is the transmitter-receiver distance in the same units as the wavelength, we can compute the path-loss over free space. Adding antenna gains, receiver sensitivity and transmission power to the equation, the available power P_R for data transmission is calculated using Eq. (6.4) and

$$P_r = P_{TX} + G_{TX} + G_{RX} + 20 \log_{10} \left(\frac{\lambda}{4\pi d} \right), \quad (6.5)$$

as adopted from [83].

Once we know the available power, we compute the error bit rate probability. The family of IEEE802.11b/g/n devices uses several modulation schemes, e.g. DBPSK modulation. The error bit rate in an additive white Gaussian noise channel, is given by

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right), \quad (6.6)$$

as adopted from [84], where P_b is the bit error probability, Q is related to the Gaussian probability density function, E_b is the energy per bit and N_0 is the Noise power spectral density in (W/Hz).

Using the datasheet values, the available power after attenuation and a value of 10^{-5} as the probability error, we are able to build a range versus throughput table for the free space scenario using our devices. Details on how the attainable bandwidth and the propulsion power consumption estimations are done can be found in the work of Braga [80].

6.4 Results

This section is divided in two parts. Initially, a QoS test case is presented in order to provide insight on the FCC model which accounts for the communication relay layer. This simulation scenario includes two sets of UAVs: the first set, is the set of buoy searching UAVs (UAV_{Search}) performing their predefined coverage paths. The mission variables of the UAV_{Search} are not affected by any communication requirements, but solely by the area coverage parameters. The second set is the set of relay UAVs (UAV_{Comm}). Their mission parameters are defined by the QoS selection done by the coordinator. The simulation includes simulated Gaussian noise for the communication channel and wind. The second part of the results demonstrates two sets of role reassignment and team reconfiguration examples.

6.4.1 QoS Simulations

Two different QoS objectives are used to illustrate the proposed FCC model. A set of multi-vehicle simulations composed by 5 exploration vehicles and 1 or 2 relay vehicles has been run with the parameters as presented in Table 6.2.

In these simulations, the searching vehicles are the end nodes of the network. They start the simulation ready to perform their exploration mission on their initial exploration location. Then, the relay communication vehicles start their mission from their initial locations at (0,0) and (50,20).

Table 6.2 QoS Simulations.

Simulation number	Energy	Bandwidth	No. Vehicles
#1. Keep Connectivity	No limit	No minimum	No limit
#2. Long Endurance	Efficient	≥ 0	bounded

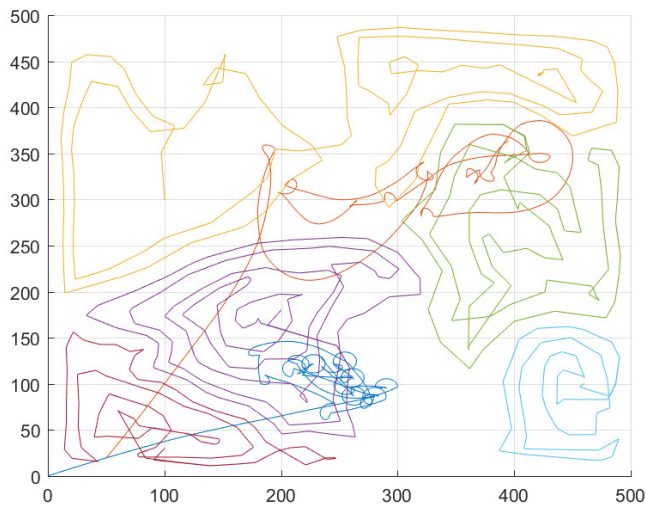


Figure 6.11 Simulation scenario including 5 exploration UAVs and 2 relay UAVs.

Enhancing link connectivity The results for this simulation are presented in Fig. 6.11. The QoS index selected for is enhancing the connectivity. Here, the relay UAVs try to keep network connectivity calculating the minimum number of vehicles needed to do so. There is no energy consumption limitation for the relay UAVs. They are only tied to the maximum values set for the dynamic model used. There is also no limit for the number of vehicles in use at the same time. In this case in order to keep connectivity with the searching vehicles the minimum number was two. Fig. 6.11 shows the path for the 7 vehicles. The dark blue and red lines are the paths of relay UAVs, switching from a go-to-location behaviour to a loitering one continuously as needed. The rest of the coloured lines are the UAV_{Search} paths. In Figs. 6.12 and 6.13 we present the simulated bandwidth between each relay node and their assigned exploration vehicles. The objective is keeping the attainable bandwidth always above 0 in order to maintain the communication link up.

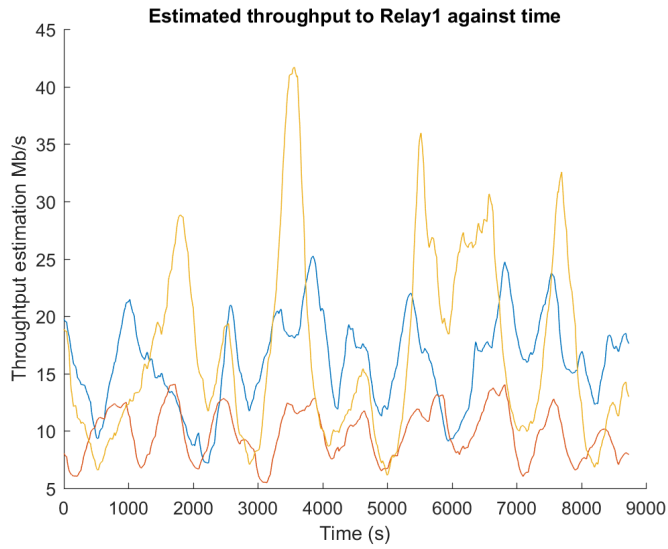


Figure 6.12 UAV Relay #1 link bandwidth to 3 exploration UAVs.

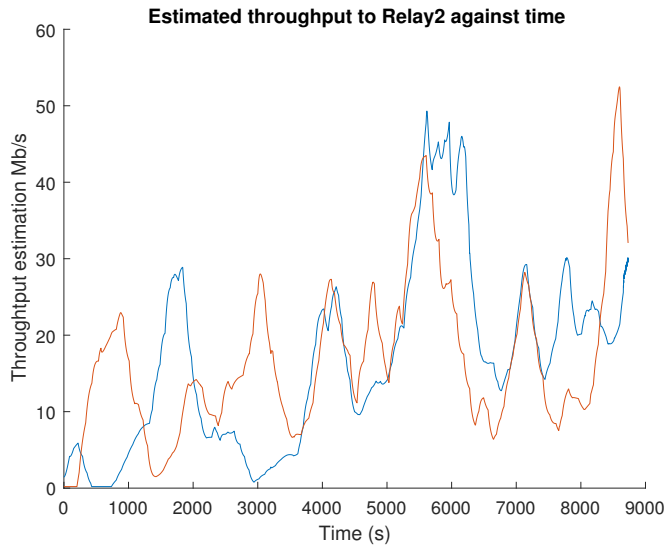


Figure 6.13 UAV Relay #2 link bandwidth to 2 exploration UAVs.

Saving total propulsion energy by reducing the number of UAVs and bounding energy consumption.

In this simulated scenario, shown in Fig. 6.14a, only one vehicle is used as a relay (dark

blue line path). Figure 6.14c shows the relative power consumption of the relay UAV, while the measured attainable bandwidth of all exploration vehicles to the relay UAV is shown in Fig. 6.14e. Please note that in this case study, there is no connectivity with some of the exploration UAVs during certain periods of time. In addition to the reduced number of vehicles, we have also added a reduced power consumption QoS level. In order to demonstrate the difference when the UAV relay power consumption is limited, sub-figures (b)¹, (d) and (f) of Fig. 6.14 show the relay UAV path, the relative power consumption and the difference in attainable bandwidth connectivity and when limiting the relay UAV speed.

6.4.2 Role reassignment and team reconfiguration experiments

The following simulation illustrates a role reassignment scenario and its connection with the market based architecture. A set of UAV_{Search} is deployed for individual area coverage and a set of UAV_{Comm} is deployed as a communication network. At a given moment, the number of available UAV_{Comm} to act as relays is not enough to fulfil the user network quality requirements. Thus, in order to avoid the network overload or a disconnection scenario, the *Coordinator* auctions the conversion of 2 UAV_{Search} into UAV_{Comm} . Like so, the searching layer is reconfigured to cover the non-searched area using the remaining UAV_{Search} . From this point of view, the behaviour of the system is the same as if those UAV_{Search} are missing. In those cases, when the global resources change, the *Coordinator* acts as the auctioneer and the UAV_{Search} vehicles as bidders. The bid of every vehicle is calculated from the area that has already covered, the area which remains on its initial plan and its remaining flight time. Then, the *Coordinator* performs the auction based on the local information, or bids, of every vehicle. This information also includes any area changes, like the appearance of a new no-fly zone.

In Fig. 6.15, the green area is the already travelled paths, the red triangles with the numbers are the current positions of the remaining UAV_{Search} and the red X's are the last seen positions of the other two converted UAVs. In Fig. 6.16 the reconfigured scenario is shown.

Team reconfiguration In order to test the dynamic reconfiguration using self-triggered control we run a simulation to test the behaviour of the lost vehicles when they are trying to recover contact. The objective of this simulation is to move all together toward a given position but keeping the connectivity during the movement and reducing the number of exchanged messages between them to avoid unexpected network congestion. In Fig. 6.17

¹ Fig. 6.14b: <https://vimeo.com/user6158089/fbfig614b>

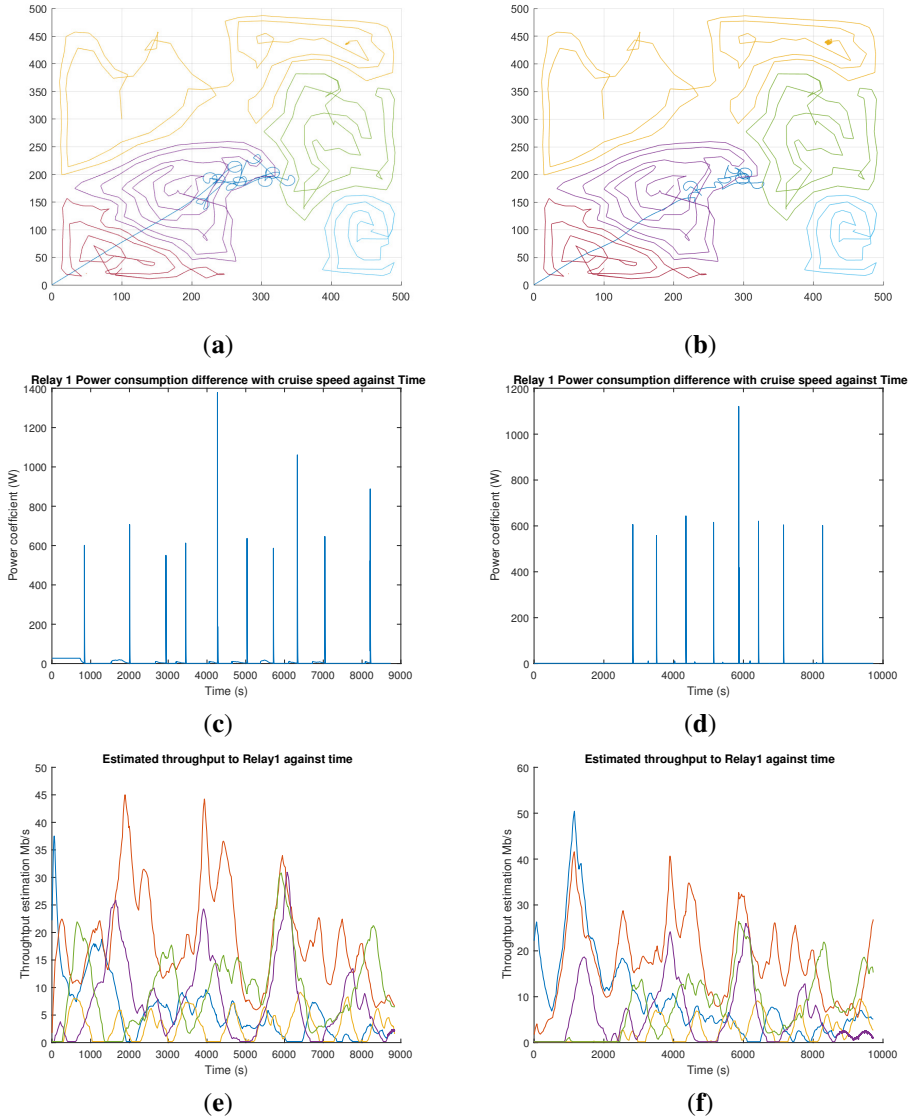


Figure 6.14 In (a) a long endurance scenario where the relay UAV path is denoted by the dark blue line. In (c), the unlimited speed UAV Relay relative power consumption, while in (e) the link bandwidth measurements to all 5 searching UAVs is presented. On the other hand, sub-figures (b), (d) and (f) show the respective limited speed case.

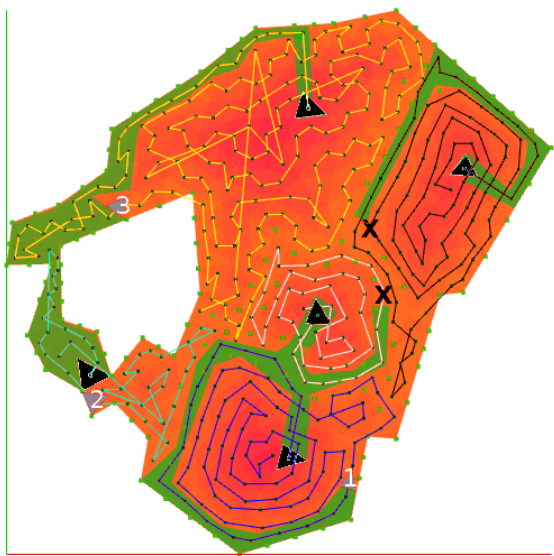


Figure 6.15 Searching algorithm waypoint plans. The green area is the already travelled paths, the grey triangles with the numbers are the current positions of the remaining UAV_{Search} and the black X's are the last seen positions of the other two converted UAVs. The black cells indicate the initial positions of the team.

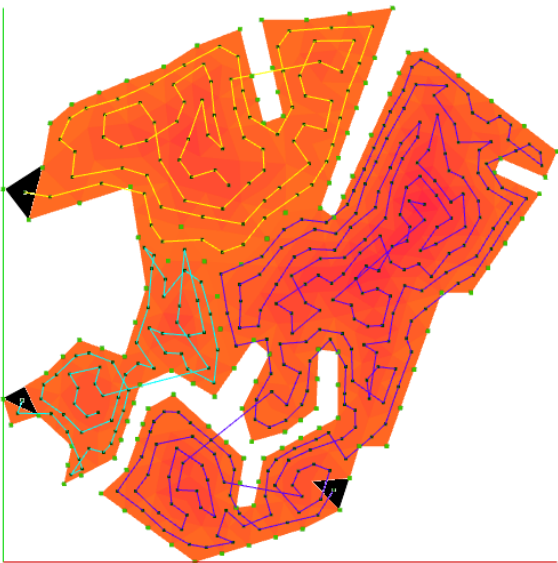


Figure 6.16 Reconfigured waypoint plans after team reassignment. The black cells indicate the initial positions. Note that the already covered areas are not considered no fly zones but the repeated coverage is not preferred.

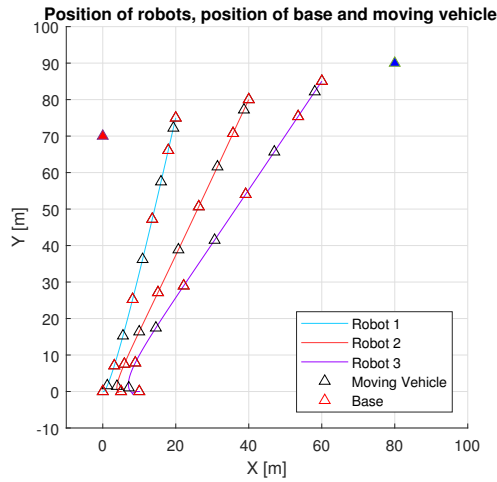


Figure 6.17 Motion simulation for 3 Relay-UAVs platoon deployment.

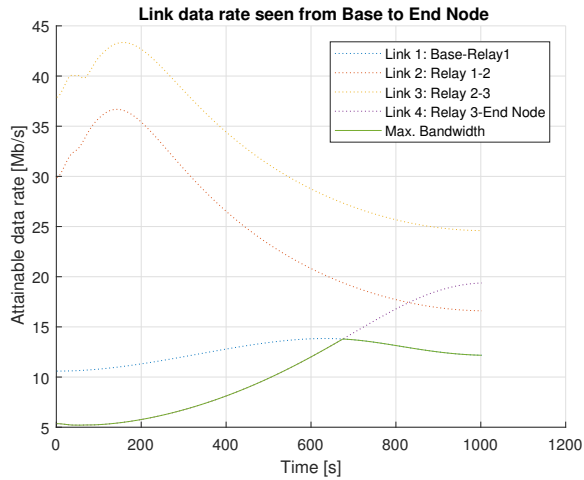


Figure 6.18 Estimated link data-rate for 3 Relay-UAVs platoon deployment motion simulation. Non-dashed line is the effective attainable bandwidth.

we present a simulated scenario with 3 UAV_{Comm} previously disconnected from the main *Coordinator*. They are moving towards a location where they can communicate with the *Coordinator* directly or through another connected UAV_{Comm} . In Fig. 6.18 the impact over the available bandwidth during motion is shown and how the effective bandwidth (green non-dashed line) always exists between extreme nodes as required.

6.5 Conclusions

In the multi-robot task allocation domain, the main question which describes the problem is "*which robot handles which task*". In this Thesis, we have tried to decompose this question, providing an architecture which analyses the roles of each part of a UAS, proposing a method which adds flexibility in a variety of applications.

The architecture discussed in this Chapter addresses the challenges of role reassignment, reconfiguration and replanning of a TOR, by demonstrating how a control framework for communication purposes can fit in the framework presented in the previous Chapters. The presented test-case of a middle communication relay between a different parts of an UAS and the coverage UAVs shows that the single-UAV framework can be part of a flexible architecture and be used by several distributed systems. The results indicate that the application of a market-based approach is dependent on the application itself and the QoS properties which are user defined.

7 Conclusions and future development

*Through the sorrow all through our splendour
Don't take offence at my innuendo*

THE QUEEN

In this chapter a summary of the main contributions and results is presented while highlighting the future directions this line of research can follow. Moreover, several advantages and disadvantages are discussed, while proposing some methods to address the issues faced.

7.1 Conclusions

The main contribution of this Thesis is the proposal a modular, multi-vehicle area partitioning and coverage framework for complex area coverage tasks. The design of the proposed framework takes into consideration area and platform characteristics which impose restrictions, in the context of modern team configurations and tasks over coastal areas. The proposed methods and algorithms address the issues of complex area decomposition and partition for an arbitrary number of robots, depending on their on-board sensors and relative capabilities. These properties increase the flexibility of the system since different robots can acquire any role in a given mission, and the partition can be recalculated accordingly; this is also true in case the area restrictions change over time. Then, coverage

waypoint flight plans are derived, aiming at producing full coverage paths having in mind the non-holonomic nature of fixed-wing robots. The latter is a design choice derived from the nature of low-cost COTS platforms used in a vast majority of modern UAS for research or commercial purposes. This Thesis also proposes a path tracking method which can be applied in any autopilot hardware by the addition of a low cost companion computer. Moreover, with the platform parametrisation methods we have presented, shorter paths and energy consumption is observed. The modular and decentralised nature of the proposed framework implies that it can be applied in any coastal and off-the-coast task with little effort.

Even though this Thesis is focused on teams of aerial non-holonomic robots operating on coastal areas, its application on other types of vehicles or areas is rather trivial. Moreover, the multi-layered treatment of an area as a graph, proposes that this framework can be extended to include any type of heterogeneous robot or sensor, as shown in the final Chapter.

7.1.1 Decomposition of complex coastal areas

In Chapter 2, a triangulation method by the means of a CDT is presented. The constraint of this triangulation is the FoV projection of the respective sensor to the surface to be covered. In that manner, a coupling of an area approximation with the actual characteristics of the platform is achieved. This method manages to create an abstraction of the continuous space and extract a list of the future coverage waypoint plans, since every centroid of every triangular cell will serve as a waypoint. If every waypoint is visited then complete coverage can be considered.

The produced mesh is an Apollonian network, while the resulting graph structure of connecting each waypoint is a third degree undirected graph; every waypoint can have at most three neighbours. This property implies a simplification of the algorithms as presented in the following chapters over a 6-neighbour method or a square grid decomposition which would have at least four (4) neighbours or at most eight (8). The results and test cases show that in comparison with a square grid decomposition, a CDT manages to perform better regarding complex shapes. This property is evident in non-square borders and no-fly zones, since a square grid decomposition cannot have variable orientation for every square and maintain graph connectivity, while a triangulation can, without necessarily produce smaller triangles in order to properly decompose an area.

The results show that this method properly decomposes any complex, non-convex area with internal holes and provides a two-dimensional point cloud with an associated graph. Nevertheless we have to note that a triangulation produces more waypoints in comparison

with a grid mesh. In order to resolve this issue, some triangles could be omitted, especially in narrow areas with many cells can be observed. In some cases, a combination of a CDT and a square grid decomposition might be more appropriate, since if the area is relatively simple or has large internal uninterrupted areas, its borders could be triangulated and the rest of the space be decomposed in a grid like manner. This would require a proper method to identify those areas and proper algorithms to perform the transition between the different segments. These cases can be tested in a future research effort on the same basis.

7.1.2 Area partitioning for multiple vehicles

In Chapter 3 a novel partitioning algorithm has been presented in order to tackle the heterogeneous, multi-robot area allocation problem. The deadlock handling capabilities of this algorithm make it a valuable addition to tackle complex coastal areas and the probability of new restrictions or team reconfiguration during task execution. Moreover, the proposed methods take into consideration the initial positions of the vehicles, proposing in that manner a flexible and robust strategy for a variety of tasks. This is particularly true in remote oceanic missions where the take-off and landing positions of the vehicles are dynamic.

These methods are derived by the initial choice of decomposing and treating the whole area as a graph. The cell weights produced in this step which belong to a vehicle, are used later on in the next steps.

The results indicate that the goal of producing uniform sub-areas, according to the relative capabilities of the vehicles is accomplished. However, the proposed solution is still an approximation, due to the nature of area discretisation. A continuous solution along with an effort to reduce the complexity of the deadlock handling algorithm is considered as a future direction on that matter.

7.1.3 Coverage Plans

In Chapter 4, each of the triangular mesh centroids has been treated as a waypoint and by their prioritisation, full coverage waypoint lists have been produced. These lists are treated as a flight plan composed by the segments connecting consecutive waypoints. The spiral nature of these plans has been a conscious decision, having in mind fixed-wing vehicle dynamics, which would permit smoother turns in order to further aid the cause of maintaining the flight path inside the ROI. As described before, this is also a mitigation in the safe separation issues between vehicles of the team or external aerial or ground restrictions. Moreover, an algorithmic technique has been developed in order to identify area characteristics which would affect the overall plan by introducing repeated coverage.

These algorithms are once more a direct manipulation of the mesh-as-a-graph strategy, by evaluating the weights introduced in the previous steps. The results show that in comparison with a boustrophedon plan, the resulting trajectories are actually shorter due to the fact that there is no extra movement in order to take a turn in every straight segment. This also has a positive side effect on lower energy consumption, since wider turns need less energy.

However we have to note that the proposed algorithms are platform dependent. A holonomic vehicle can follow these paths with greater ease than a non-holonomic one. Moreover, the distance between the waypoints along with the dynamics of the platform show that some waypoint prioritisation cannot be followed by non-holonomic vehicles without taking unnecessary turns and back and forth motions. Modern autopilots solve this issue by applying an acceptance radius of waypoint visit success. Nevertheless this method is not always sufficient. A solution to this issue could be to apply a kinodynamic path planning algorithm; it would have as an input the vehicle dynamics along with the waypoint distances and prioritise some waypoints over others. In that manner longer trajectories are expected but an exact waypoint visit and smoother trajectories are expected.

7.1.4 Path tracking

In Chapter 5 a platform and autopilot independent method of enhancing the on-board path tracking algorithms has been developed. This method has two parts: the angle dependent look-up tables and a platform parametrisation. The first can either be simulated beforehand or can be part of the latter platform parametrisation for each vehicle. The results show that these methods manage to reduce the flight error by using a computationally inexpensive method of a table lookup, thus aiding the safe separation mitigation methods.

However, we have to note that this method can be partially automatised, by the use of an on-board companion computer, in conjunction with the autopilot and the control framework. However, this process would still require some coefficient limits and proper gain parametrisation, it would greatly simplify the procedure.

7.1.5 A distributed framework

In Chapter 6 the modular nature of the previous steps is organised in a top-down framework, which can be applied in any vehicle. These modules are characterised as path-enhancement methods for any team of robots since each module can be replaced by systems or frameworks that are already present in a given setup. This is demonstrated in this Chapter through a MarineUAS specific test-case, where a FCC module acts as communication relay in a

3-layered experimental framework of buoy data acquisition. In that test case, a market-based like organisation architecture is identified, where each layer of the framework is responsible for a given task. As results show, our proposed algorithms and methods manage to provide flexibility and robustness in the system since each robot can acquire the role of any other robot, the team can reconfigure itself and the whole system is prone to area or team changes.

We have to note that even though the given test case manages to demonstrate some qualitative aspects of the proposed framework, several other experiments and actual test flights with different types of robots are needed in order to further generalise the solution.

7.2 Future work

Future work should be focused mainly on the drawbacks as presented in the previous section and Chapters. Regarding the two dimensional mesh generation, further developments could include a three dimensional representation of space in order to include relevant information, such as wind fields and aerial restrictions. It would also permit a proper treatment of space for robots which operate in no-flat surfaces, like the sea bottom or uneven shores. Moreover and since this framework is focused on heterogeneous synergies, the produced graph should incorporate task defined weights and properties in order to include uncertainties in motion depending on the dynamics of each vehicle.

Regarding coverage waypoint plans, the introduction of task dependent belief maps would permit the parametrisation of waypoint prioritisation according to on-line belief updates. Combining this information with the aforementioned three dimensional representation, more complicated trajectories can be derived. Once more, the latter will have the positive side-effect of properly treating the safe-separation issues faced in multi-robot teams in complex environments.

Even though path tracking is a widely investigated issue, modern, *Vertical Take-Off and Landing (VTOL)* platforms introduce a valuable tool which permits covering wide areas with relatively low energy needs while being able to maintain a certain position when needed, combining in that way the advantages of holonomic and non-holonomic vehicles. The integration of these vehicles in a TOR would require an adequate path tracking technique. Please note that these vehicles are ideal in coastal and oceanic missions, since they manage to solve the frequent problem of deployment and landing on mobile platforms or ships.

By introducing an on-line learning method for integrating real-world disturbances like wind into the path tracker, a decentralised scheduling schema for the whole team could be

derived. Finally, real-world applications and needs from different disciplines of coastal and marine sciences, characterise the most commonly used platforms. Their properties and capabilities permit a proper definition of what kind of prioritisation is needed in these distributed systems. Hence, a dynamic architecture has to be further developed, including modern communication protocols and devices, always focusing on the low cost COTS material.

List of Figures

1.1	A coastal science scenario	2
1.2	Objectives - Chapters	5
2.1	A problematic found in grid decomposition methods	13
2.2	A coastal area complexity example	15
2.3	Common decomposition issues in literature	16
2.4	Aerial restrictions example	18
2.5	CDT of a coastal area	21
2.6	The coordinate frames of the considered system	22
2.7	Calculating the CDT side constraint in an simple scenario	23
2.8	Calculating the CDT side constraint in more complex scenarios	25
2.9	CDT calculation in a communication sensor scenario	27
2.10	CDT based on sensor frame rate	28
2.11	A CDT of a non-convex polygon with an internal no-fly zone	29
2.12	Grid decomposition performance of the area in Fig 2.11	31
2.13	A square grid decomposition on an actual area	31
3.1	A partition example on an actual area	37
3.2	A partition deadlock scenario	41
3.3	Transition sequence selection	43
3.4	Selected testing areas	45
3.5	Partitioning algorithms comparison	46
3.5	Partitioning algorithms comparison	47
3.6	Partitioning algorithm comparison	47
3.7	Area partition. Even vs Random distribution	48
3.8	A graphical representation of Tables 3.4 and 3.5	55
4.1	Time graph for every operation of Table 3.3	61
4.2	Spiral planning, trajectory and coverage of an area	62
4.3	Area characteristics where valleys and inlets are present	63

4.4	I , P and Θ correlation	64
4.5	The intersection of P and Θ for different area sizes, in relation with I	65
4.6	Different values of I are required in different scenarios	66
4.7	Experimental framework	68
4.8	The qTnP main application	69
4.9	Spiral path and trajectory of initial algorithm	70
4.10	A comparison between the proposed method and a classic grid decomposition with boustrophedon motions	71
4.11	An actual area for a simulated scenario	72
4.12	Spiral paths for different inlet coefficient values	72
4.13	Algorithm 6 and Algorithm 7 comparison	73
4.14	Spiral paths for different I values in an actual area	74
4.15	Angle distribution graph of Abaco region case	75
4.16	Simulated trajectory and groundstation view of Abaco region case	75
4.17	Angle distribution in different Lloyd iteration trials	76
4.18	Flight area and platform	77
4.19	Comparison of boustrophedon and proposed spiral path (field experiment)	78
5.1	Lateral acceleration and convergence in pure pursuit algorithms	84
5.2	Correlation of L , θ and R	84
5.3	Linear model of [67]	85
5.4	A pure pursuit based algorithm	87
5.5	The L coefficient which reduces area A is different for different θ angles.	87
5.6	Odroid XU4 and PX4 autopilot	88
5.7	Groundstation view of trajectories for different L values.	89
5.8	Plot corresponding to Table 5.1	90
5.9	Proposed tracking versus onboard autopilot.	91
5.10	The effect of different L coefficient values	91
5.11	Area of performed simulations, segregation and planning.	92
5.12	Integrated area for different test cases	93
5.13	Flight trajectories view from the groundstation, indicating smaller error.	94
5.14	Comparison of autopilot tracker and proposed method	95
5.15	UAV platforms	95
5.16	Tuning and enhanced trajectories of conducted experiments	96
5.17	The pitch control loop	97
5.18	The tuning trajectory	97
5.19	Pure pursuit field experiment	99
6.1	The top-to-bottom proposed framework	103
6.2	A multi-layered framework	107
6.3	Buoy RF-circuitry	108
6.4	A buoy searching CDT example	109
6.5	The FCC variables and model	110
6.6	Role reassignment diagram	111
6.7	Team reconfiguration when a group is isolated	112

6.8	Low-power RF wake-up circuit	113
6.9	Power received vs Distance(Texas CC1101)	114
6.10	Total drag vs air speed	116
6.11	Simulation scenario including 5 exploration UAVs and 2 relay UAVs	118
6.12	UAV Relay #1 link bandwidth to 3 exploration UAVs	119
6.13	UAV Relay #2 link bandwidth to 2 exploration UAVs	119
6.14	Path, power consumption and bandwidth of relay UAVs	121
6.15	Searching algorithm waypoint plans	122
6.16	Reconfigured waypoint plans after team reassignment	122
6.17	Motion simulation for 3 Relay-UAVs platoon deployment	123
6.18	Estimated link data-rate for 3 Relay-UAVs	123

List of Tables

2.1	CDT performance based on lower angle bound constraint	30
2.2	Square grid and CDT decomposition methods comparison	30
3.1	F metric comparison of evenly distributed robots	46
3.2	F metric comparison of randomly distributed robots	48
3.3	Computational complexity of partitioning	49
3.4	Deadlock handling for even distribution of robots	51
3.4	Deadlock handling for even distribution of robots	52
3.5	Deadlock handling for random distribution of robots	53
3.5	Deadlock handling for random distribution of robots	54
4.1	Path metrics comparison when a classical grid decomposition strategy has been used for the areas shown in Fig. 4.10	70
4.2	Metrics on inlet coefficient corresponding to Fig. 4.12	73
4.3	Comparison between Algorithm 6 and Algorithm 7, corresponding to Fig. 4.13	74
4.4	Metrics on inlet sensitivity values corresponding to Fig. 4.14	75
4.5	Time in seconds for Lloyd optimization in the whole area and in each UAV configuration space	77
4.6	Comparative experimental results for the boustrophedon and proposed method missions	78
5.1	Lookup distance coefficient table based on different path angles	90
5.2	MarineUAS Platform characteristics	96
5.3	Tuned Parameters for MarineUAS-1	98
5.4	Experimental results in average for the different missions	98
6.1	Lift, Drag and Drag Coefficient Values used in Simulations	115
6.2	QoS Simulations	118

Bibliography

- [1] National Aeronautics and Space Administration. UTM: Air traffic management for low-altitude drones, 2015.
- [2] Jean-daniel Boissonnat, Olivier Devillers, Sylvain Pion, Monique Teillaud, and Mariette Yvinec. Triangulations in CGAL. *Computational Geometry*, 22(1-3):5–19, may 2002.
- [3] S M LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006.
- [4] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.
- [5] João Valente, Jaime Del Cerro, Antonio Barrientos, and David Sanz. Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach. *Computers and Electronics in Agriculture*, 99:153–159, 2013.
- [6] Jean Berger and Nassirou Lo. An innovative multi-agent search-and-rescue path planning approach. *Computers & Operations Research*, 53:24–31, 2015.
- [7] Joon Seop Oh, Yoon Ho Choi, Jin Bae Park, and Yuan F. Zheng. Complete coverage navigation of cleaning robots using triangular-cell-based map. *IEEE Transactions on Industrial Electronics*, 51(3):718–726, 2004.
- [8] Yan Li, Hai Chen, Meng Joo Er, and Xinmin Wang. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. *Mechatronics*, 21(5):876–885, 2011.
- [9] Howie Choset and Philippe Pignon. Coverage Path Planning : The Boustrophedon Cellular Decomposition. *Autonomous Robots*, 9(3):247–253, 1997.
- [10] Liam Paull, Carl Thibault, Amr Nagaty, Mae Seto, and Howard Li. Sensor-Driven Area Coverage for an Autonomous Fixed-Wing Unmanned Aerial Vehicle. *IEEE Transactions on Cybernetics*, PP(99):1–1, 2014.

- [11] Stanislav Bochkarev and Stephen L Smith. On minimizing turns in robot coverage path planning. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1237–1242. IEEE, aug 2016.
- [12] Matthew Coombes, Wen-hua Chen, and Cunjia Liu. Boustrophedon coverage path planning for UAV aerial surveys in wind. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1563–1571. IEEE, jun 2017.
- [13] Adiyabaatar Janchiv, Dugarjav Batsaikhan, ByungSoo Kim, Won Gu Lee, and Soon-Geul Lee. Time-efficient and complete coverage path planning based on flow networks for multi-robots. *International Journal of Control, Automation and Systems*, 11(2):369–376, 2013.
- [14] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner. Networked UAVs as aerial sensor network for disaster management applications. *e & i Elektrotechnik und Informationstechnik*, 127(3):56–63, mar 2010.
- [15] Carmelo Di Franco and Giorgio Buttazzo. Coverage Path Planning for UAVs Photogrammetry with Energy and Resolution Constraints. *Journal of Intelligent {&} Robotic Systems*, 83(3):445–462, 2016.
- [16] Bijan Ranjbar-Sahraei, Gerhard Weiss, and Ali Nakisaee. *A Multi-robot Coverage Approach Based on Stigmergic Communication*, pages 126–138. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [17] David Saldaña, Luiz Chaimowicz, and Mario F M Campos. Searching for Regions Out of Normal Conditions Using a Team of Robots. pages 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
- [18] Daisuke Kurabayashi, Jun Ota, Tamio Arai, and Eiichi Yoshida. An algorithm of dividing a work area to multiple mobile robots. In *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 2, pages 286–291. IEEE, 1995.
- [19] Ercan U Acar, Howie Choset, Alfred A Rizzi, Prasad N Atkar, and Douglas Hull. Morse decompositions for coverage tasks. *The International Journal of Robotics Research*, 21(4):331–344, 2002.
- [20] R.A Jarvis and J.C. Byrne. Robot navigation: Touching, seeing and knowing. In *Proceedings of the 1st Australian Conference on Artificial Intelligence*, 1986.
- [21] Daisuke Kurabayashi, Jun Ota, Tamio Arai, and Eiichi Yoshida. Cooperative sweeping by multiple mobile robots. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 2, pages 1744–1749. IEEE, 1996.
- [22] David A Haessig, Ron T Ogan, and Mark Olive. “sense and avoid”-what’s required for aircraft safety? In *SoutheastCon, 2016*, pages 1–8. IEEE, 2016.

- [23] Ricardo Román Cordón, Francisco Javier Sáez Nieto, and Cristina Cuerno Rejado. Rpas integration in non segregated airspace: the sesar approach. *Proceedings of the SESAR Innovation Days*, 2014.
- [24] Nathan M Paczan, Jeremy Cooper, and Eric Zakrzewski. Integrating unmanned aircraft into nextgen automation systems. In *Digital Avionics Systems Conference (DASC), 2012 IEEE/AIAA 31st*, pages 8C3–1. IEEE, 2012.
- [25] Laurent Rineau. 2D conforming triangulations and meshes. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.7 edition, 2015.
- [26] Jonathan Richard Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22(1-3):21–74, 2002.
- [27] Marcelo Kallmann, Hanspeter Bieri, and Daniel Thalmann. Fully Dynamic Constrained Delaunay Triangulations. *Geometric Modelling for Scientific Visualization*, pages 241 – 257, 2003.
- [28] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.
- [29] Massimiliano Pepe and Giuseppina Prezioso. Two approaches for dense dsm generation from aerial digital oblique camera system. In *Proceedings of the 2nd International Conference on Geographical Information Systems Theory, Applications and Management*, pages 63–70, 2016.
- [30] D Blake Barber, Joshua D Redding, Timothy W McLain, Randal W Beard, and Clark N Taylor. Vision-based target geo-location using a fixed-wing miniature air vehicle. *Journal of Intelligent and Robotic Systems*, 47(4):361–382, 2006.
- [31] Cgal - 2d conforming triangulations and meshes - 2.5 optimization of meshes with lloyd. http://doc.cgal.org/latest/Mesh_2/index.html#secMesh_2_optimization, 2018.
- [32] Jane Tournois, Pierre Alliez, and Olivier Devillers. 2D centroidal Voronoi tessellations with constraints. *Numerical Mathematics*, 3(2):212–222, 2010.
- [33] Susan Hert and Vladimir Lumelsky. Polygon Area Decomposition for Multiple-Robot Workspace Division. *International Journal of Computational Geometry and Applications*, 8:437–466, 1998.
- [34] Mahdi Hassan, Dikai Liu, Shoudong Huang, and Gamini Dissanayake. Task oriented area partitioning and allocation for optimal operation of multiple industrial robots in unstructured environments. In *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 1184–1189. IEEE, dec 2014.
- [35] Athanasios Ch. Kapoutsis, Savvas A. Chatzichristofis, and Elias B. Kosmatopoulos. DARP: Divide Areas Algorithm for Optimal Multi-Robot Coverage Path Planning. *Journal of Intelligent & Robotic Systems*, jan 2017.

- [36] Jose J. Acevedo, Begoña C. Arrue, Jose Miguel Diaz-Bañez, Inmaculada Ventura, Ivan Maza, and Anibal Ollero. One-to-one coordination algorithm for decentralized area partition in surveillance missions with a team of aerial robots. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 74(1-2):269–285, 2014.
- [37] J.J. Acevedo, B.C. Arrue, J.M. Diaz-Banez, I. Ventura, I. Maza, and A. Ollero. Decentralized strategy to ensure information propagation in area monitoring missions with a team of UAVs under limited communications. In *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS 2013)*, pages 565–574, 2013.
- [38] J.J. Acevedo, B. Arrue, I. Maza, and A. Ollero. Cooperative perimeter surveillance with a team of mobile robots under communication constraints. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5067–5072, Tokyo, Japan, 2013.
- [39] A Kassir, R Fitch, and S Sukkarieh. Communication-aware information gathering with dynamic information flow. *International Journal of Robotics Research*, 34(2):173–200, 2015.
- [40] Muzaffer Kapanoglu, Mete Alikalfa, Metin Ozkan, Ahmet Yazıcı, and Osman Parlaktuna. A pattern-based genetic algorithm for multi-robot coverage path planning minimizing completion time. *Journal of Intelligent Manufacturing*, 23(4):1035–1045, 2012.
- [41] Yi Guo and M Balakrishnan. Complete coverage control for nonholonomic mobile robots in dynamic environments, 2006.
- [42] K.S. Senthilkumar and K.K. Bharadwaj. Multi-robot exploration and terrain coverage in an unknown environment. *Robotics and Autonomous Systems*, 60(1):123–132, jan 2012.
- [43] Yongguo Mei, Yung-Hsiang Lu, Y Charlie Hu, and CS George Lee. Energy-efficient motion planning for mobile robots. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4344–4349. IEEE, 2004.
- [44] The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 4.8.1 edition, 2015.
- [45] JF Araujo, PB Sujit, and João B Sousa. Multiple uav area decomposition and coverage. In *Computational Intelligence for Security and Defense Applications (CISDA), 2013 IEEE Symposium on*, pages 30–37. IEEE, 2013.
- [46] Pooyan Fazli, Alireza Davoodi, and Alan K Mackworth. Multi-robot repeated area coverage. *Autonomous Robots*, 34(4):251–276, 2013.
- [47] Seyed Abbas Sadat, Jens Wawerla, and Richard Vaughan. Fractal trajectories for online non-uniform aerial coverage. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2971–2976. IEEE, 2015.

- [48] I. Maza and A. Ollero. *Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms*, volume 6 of *Distributed Autonomous Robotic Systems*, pages 221–230. Springer Verlag, 2007.
- [49] Anqi Xu, Chatavut Viriyasuthee, and Ioannis Rekleitis. Optimal complete terrain coverage using an unmanned aerial vehicle. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2513–2519, 2011.
- [50] N. Kreciglowa, K. Karydis, and V. Kumar. Energy efficiency of trajectory generation methods for stop-and-go aerial robot navigation. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 656–662, June 2017.
- [51] Susan Hert, Sanjay Tiwari, and Vladimir Lumelsky. A terrain-covering algorithm for an AUV. *Autonomous Robots*, 3(2-3):91–119, 1996.
- [52] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.
- [53] Rviz – 3d visualization tool for ros. <http://wiki.ros.org/rviz>, 2016.
- [54] Mavros – mavlink extendable communication node for ros with proxy for ground control station. <http://wiki.ros.org/mavros>, 2016.
- [55] Fixed wing ardupilot instance for autopilot hardware. <http://ardupilot.org/plane/index.html>, 2016.
- [56] Autopilot software in the loop simulation. <http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html>, 2016.
- [57] Jsbsim, the open source flight dynamics model in c++. <http://jsbsim.sourceforge.net/>, 2016.
- [58] Qgroundcontrol - ground control station for small air - land - water autonomous unmanned systems. <http://qgroundcontrol.org/>, 2016.
- [59] L. Meier, D. Honegger, and M. Pollefeys. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, may 2015.
- [60] Omead Amidi. Integrated mobile robot control. Technical report, Carnegie-Mellon University, Pittsburgh PA, Robotics Institute, 1990.
- [61] Anibal Ollero and Guillermo Heredia. Stability analysis of mobile robot path tracking. In *IEEE International Conference on Intelligent Robots and Systems*, volume 3, 1995.
- [62] V. Gírbés, L. Armesto, and J. Tornero. Path following hybrid control for vehicle stability applied to industrial forklifts. *Robotics and Autonomous Systems*, 62(6), 2014.

- [63] M. Park, S. Lee, and W. Han. Development of steering control system for autonomous vehicle using geometry-based path tracking algorithm. *ETRI Journal*, 37(3), 2015.
- [64] Y. Shan, W. Yang, C. Chen, J. Zhou, L. Zheng, and B. Li. CF-Pursuit: A Pursuit Method with a Clothoid Fitting and a Fuzzy Controller for Autonomous Vehicles. *International Journal of Advanced Robotic Systems*, 12(9), 2015.
- [65] PB Sujit, Srikanth Saripalli, and Jogto Borges Sousa. An evaluation of uav path following algorithms. In *Control Conference (ECC), 2013 European*, pages 3332–3337. IEEE, 2013.
- [66] Derek R Nelson, D Blake Barber, Timothy W McLain, and Randal W Beard. Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23(3):519–529, 2007.
- [67] Sanghyuk Park, John Deyst, and Jonathan P How. A new nonlinear guidance logic for trajectory tracking. In *AIAA guidance, navigation, and control conference and exhibit*, pages 16–19, 2004.
- [68] G. Heredia and A. Ollero. Stability of autonomous vehicle path tracking with pure delays in the control loop. *Advanced Robotics*, 21(1-2), 2007.
- [69] B P Gerkey and M J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939–954, 2004.
- [70] G Ayorkor Korsah, Anthony Stentz, and M Bernardine Dias. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*, 32(12):1495–1512, oct 2013.
- [71] M Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.
- [72] A unique, ultra-low power analog ic enables rf wakeup applications. <https://www.digikey.be/Web%20Export/Supplier%20Content/touchstone-1238/pdf/touchstone-wp-rf-wakeupl.pdf>, 2016.
- [73] K Kaushik, Deepak Mishra, Swades De, Jun-Bae Seo, Soumya Jana, Kaushik Chowdhury, Stefano Basagni, and Wendi Heinzelman. Rf energy harvester-based wake-up receiver. In *SENSORS, 2015 IEEE*, pages 1–4. IEEE, 2015.
- [74] Chenyang Lu, John A Stankovic, Sang H Son, and Gang Tao. Feedback control real-time scheduling: Framework, modeling, and algorithms. *Real-Time Systems*, 23(1-2):85–126, 2002.
- [75] Juan Braga, Fotios Balampanis, Pedro Aguiar, João Sousa, Ivan Maza, and Anibal Ollero. Coordinated Efficient Buoys Data Collection in Large Complex Coastal Environments using UAVs. In *Oceans17 MTS IEEE, Anchorage, Alaska, AK, USA*, 2017.

- [76] R Praveen Jain, A Pedro Aguiar, and João Sousa. Self-triggered cooperative path following control of fixed wing unmanned aerial vehicles. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pages 1231–1240. IEEE, 2017.
- [77] Yuan Fan, Lu Liu, Gang Feng, and Yong Wang. Self-triggered consensus for multi-agent systems with zeno-free triggers. *IEEE Transactions on Automatic Control*, 60(10):2779–2784, 2015.
- [78] Venanzio Cichella, Ronald Choe, Syed Bilal Mehdi, Enric Xargay, Naira Hovakimyan, Vladimir Dobrokhodov, Isaac Kaminer, Antonio M Pascoal, and A Pedro Aguiar. Safe coordinated maneuvering of teams of multirotor unmanned aerial vehicles: A cooperative control framework for multivehicle, time-critical missions. *IEEE Control Systems*, 36(4):59–82, 2016.
- [79] E Xargay, I Kaminer, A Pascoal, N Hovakimyan, V Dobrokhodov, V Cichella, A Aguiar, and R Ghabcheloo. Time-critical cooperative path following of multiple uavs over time-varying networks. Technical report, Naval Postgraduate School, Monterey CA, Department of Mechanical and Astronautical Engineering, 2011.
- [80] J Braga, A Alessandretti, A Pedro Aguiar, and João Sousa. A feedback motion strategy applied to a uav to work as an autonomous relay node for maritime operations. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pages 625–632. IEEE, 2017.
- [81] Alessandro Rucco, PB Sujit, Antonio Pedro Aguiar, and João Sousa. Optimal uav rendezvous on a ugv. In *AIAA Guidance, Navigation, and Control Conference*, page 0895, 2016.
- [82] Daniel Raymer. Aircraft design: A conceptual approach, 2012.
- [83] B Whitfield Griffith. *Radio-electronic transmission fundamentals*. SciTech Publishing, 2000.
- [84] JM Hernando. Transmisión por radio, 4ª edición, editorial centro de estudios ramón acres, sa etsit, 1995.

Index

- ADS-B, 18
- ATM, 7, 12, 18
- CDA, 106
- CDT, 4, 11, 12, 20, 37, 58, 64, 67
- CGAL, 19, 20, 68
- COTS, 6, 7, 77, 83, 86, 101
- FAA, 18
- FCC, 108
- FoV, 12, 14, 20, 58, 66
- NASA, 12
- QoS, 108
- ROI, 57, 59, 78, 102
- ROS, 66, 68
- RWS, 60, 65
- SESAR, 12
- SITL, 4
- TOR, 101
- UAS, 1, 6, 12, 17, 18
- UAV, 2, 12, 14, 16–19, 22, 57
- UTM, 18
- VTOL, 129
- art gallery problem, 58
- ATM, 18
- auctioneer, 106
- border proximity cost, 65
- buoys, 2
- capability coefficient, 35
- CDT, 20
- Cellular Decomposition, 12
- coordinator, 102, 106
- Delaunay Triangulation, 19
- FoV, 29, 69, 77
- Grid, 12
- heterogeneous, 2
- inlet, 65
- inlet sensitivity, 61
- inlets, 61, 64
- moveAWP, 41
- moveRWS, 41
- multiple Travelling Salesman (mTSP), 57
- NASA, 18
- non-holonomic, 3
- prioritised, 57
- pure pursuit, 81
- Seven Bridges of Königsberg, 57
- simplicial complex, 4
- valleys, 61, 64

Glossary

ADS-B Autonomous Dependent System-Broadcast. 18, 145

ATM Air Traffic Management. 7, 12, 18, 145

CDA Coordinated Data Acquisition. 106, 145

CDT Constrained Delaunay Triangulation. 4, 11, 12, 20, 37, 58, 64, 67, 145

CGAL Computational Geometry Algorithms Library. 19, 20, 68, 145

COTS Commercial-Off-The-Shelf. 6, 7, 77, 83, 86, 101, 145

FAA Federal Aviation Administration. 18, 145

FCC Feedback Communication Control. 108, 145

FoV Field of View. 12, 14, 20, 58, 66, 145

NASA National Aeronautics and Space Administration. 12, 145

QoS Quality of Service. 108, 145

ROI Region Of Interest. 57, 59, 78, 102, 145

ROS Robotic Operating System. 66, 68, 145

RWS Reverse Watershed Schema. 60, 65, 145

SESAR Single European Sky ATM Research. 12, 145

SITL Software-In-The-Loop. 4, 145

TOR Team Of Robots. 101, 145

UAS Unmanned Aerial System. 1, 6, 12, 17, 18, 145

UAV Unmanned Aerial Vehicle. 2, 12, 14, 16–19, 22, 57, 145

UTM Unmanned Air Traffic Management. 18, 145

VTOL Vertical Take-Off and Landing. 129, 145