

# Fixed-wing drones for communication networks

THÈSE N° 6859 (2016)

PRÉSENTÉE LE 18 MARS 2016

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR  
LABORATOIRE DE SYSTÈMES INTELLIGENTS  
PROGRAMME DOCTORAL EN SYSTÈMES DE PRODUCTION ET ROBOTIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

**Maja VARGA**

acceptée sur proposition du jury:

Dr D. Gillet, président du jury  
Prof. D. Floreano, directeur de thèse  
Prof. S. Bogdan, rapporteur  
Prof. D. Scaramuzza, rapporteur  
Prof. C. Jones, rapporteur



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE

Suisse  
2016



# Acknowledgements

I would like to express my deepest gratitude to my supervisor Prof. Dario Floreano for accepting me as his PhD student and for guiding me through the four years of the thesis. I would like to thank him for providing me with the freedom to explore different directions and for his great support through this process.

I wish to thank Prof. Denis Gillet for his accepting to be the jury president for my PhD oral exam and I am thankful to Prof. Stjepan Bogdan, Prof. Davide Scaramuzza and Prof. Colin Jones for accepting to participate in my thesis jury.

Also I would like to thank to all my colleagues from Swarmix and Smavnet II project, for amazing experience of working together and learning from each other. Furthermore, I would like to thank my colleagues from LIS for friendly and comfortable atmosphere at EPFL, day after day. Special thanks goes Meysam for being a great officemate and willing to invest his time into supporting me and discuss about my thesis. I would also like to thank to Greg who assisted me in all experiments and was immense help for hardware development in all our projects. I would like to thank all my colleagues from Larics Lab, for unforgettable moments and crazy parties. I would also like to thank Prof. Stjepan Bogdan. His scientific enthusiasm, creativity and curiosity were one of the main reasons why I decided to pursue a PhD in Robotics. He showed me how cool is to be a researcher and I will be always grateful for this. I want to thank my friends in Lausanne, for sharing the PhD program experience and going together through the best and the worst of it. Thank you for never letting me feel alone in the face of challenges. Another thanks goes to my friends in Zagreb, for standing by me despite the distance and reminding me of who I am outside the academia.

Finally but the most importantly I want to thank my aunt Marija, for making me feel at home in Switzerland, to my brother Matija, for always being there for me, for being my advisor and for being the rational one, my grandma Jelka, who was always full of understanding and deep caring and patience, my parents Zdenka and Zdravko for their unlimited love. If there was not their guidance and support, all this would not be possible.

Lastly, to my dear Darko. Thank you for bringing joy in my life day after day.



# Abstract

In the last decade, drones became frequently used to provide eye-in-the-sky overview in the outdoor environment. They are widely used for surveillance, mapping and exploring unknown terrains. Their main advantage compared to the other types of robots is that they can fly above obstacles and rough terrains and they can quickly cover large areas. These properties also open a new application; drones could provide a multi-hop, line of sight communication for groups of ground users.

The aim of this thesis is to develop a drone team that will establish wireless ad-hoc network between users on the ground and distributively adapt links and spatial arrangement to the requirements and motion of the ground users. For this application, we use lightweight, easy-portable fixed wing drones. Such platforms can be easily and quickly deployed. Fixed wing drones have higher forward speed and higher battery life than hovering platforms. On the other hand, fixed wing drones have unicycle dynamics with constrained forward speed which makes them unable to hover or perform sharp turns.

The first challenge consists in bridging unicycle dynamics of the fixed wing drones. Some control strategies have been proposed and validated in simulations using the average distance between the target and the drone as a performance metric. However, besides the distance metric, energy expenditure of the flight also plays an important role in assessing the overall performance of the flight. We propose a new methodology that introduces a new metric (energy expenditure), we compare existing methods on a large set of target motion patterns and present a comparison between the simulation and field experiments on proposed target motion patterns. Using this new methodology we examine the performance of state-of-the-art control strategies and based on simulations and field experiments we choose the strategy that has the best performance according to our comparison criteria.

The second challenge consists in developing a formation control algorithm that will allow fixed wing robots to provide a wide area coverage and to relay data in a wireless ad-hoc network. In such applications fixed wing drones have to be able to regulate an inter-drone distance. Their reduced maneuverability presents the main challenge to design a formation algorithm that will regulate an inter-drone distance. To address this challenge, we present a distributed control

## Abstract

---

strategy that relies only on local information. Each drone has its own virtual agent, it follows the virtual agent by performing previously evaluated and selected target tracking strategy, and flocking interaction rules are implemented between virtual agents. It is shown in simulation and in field experiments with a team of fixed wing drones that using this distributed formation algorithm, drones can cover an area by creating an equilateral triangular lattice and regulate communication link quality between neighboring drones.

The third challenge consists in allowing connectivity between independently moving ground users using fixed wing drone team. We design two distributed control algorithms that change drones' spatial arrangement and interaction topology to maintain the connectivity. We propose a potential field based strategy which adapts distance between drones to shrink and expand the fixed wing drones' formation. In second approach, market-based adaptation, drones distributively delete interaction links to expand the formation graph to a tree graph. In simulations and field experiments we show that our proposed strategies successfully maintain independently moving ground users connected.

Overall, this thesis presents synthesis of distributed algorithms for fixed wing drones to establish and maintain wireless ad-hoc communication networks.

Key words: communication networks, fixed-wing drones, formation control, flocking, connectivity maintenance, wireless ad-hoc network

# Résumé

Dans la dernière décennie, les drones sont utilisés souvent pour donner un aperçu aérien de l'environnement extérieur. Ils sont largement utilisés pour la surveillance, la cartographie et l'exploration des terrains inconnus. Leur avantage principal par rapport aux autres types de robots consiste dans le fait qu'ils peuvent voler au-dessus des obstacles et des terrains accidentés ainsi qu'ils peuvent couvrir rapidement des zones vastes. Ces propriétés indiquent également une nouvelle application : les drones pourraient fournir un multi-hop ainsi que la ligne de la communication visuelle des groupes d'utilisateurs terrestres.

On pourrait imaginer un scénario dans lequel une couverture de réseau cellulaire est médiocre ou l'infrastructure de communication est endommagée à cause des événements catastrophiques. Dans de tels cas, les utilisateurs terrestres pourraient bénéficier de leur propre réseau de communication dédié qui ne dépend pas d'une infrastructure cellulaire rigide. Ces réseaux des drones pourraient être déployés rapidement, pourraient fournir des voies de communication de haute qualité et adapter leur organisation spatiale du mouvement aux utilisateurs terrestres. Ces exigences présentent des défis principaux abordés dans cette thèse.

L'objectif de cette thèse est de développer un ensemble des drones capable d'établir un réseau sans fil ad hoc entre les utilisateurs terrestres et d'adapter les liens distributifs et l'arrangement spatial aux exigences et au mouvement des utilisateurs terrestres. À cette fin, nous utilisons les drones à voilure fixe portables et légers. Ces plateformes peuvent être facilement et rapidement déployées. D'un côté, les drones à voilure fixe ont une plus grande vitesse d'avancement et la vie de la batterie plus élevée que des plateformes planant. De l'autre côté, les drones à voilure fixe ont une dynamique monocycle avec la vitesse d'avancement limitée qui les rend incapables de planer ou d'effectuer des virages serrés.

Le premier défi consiste à contourner la dynamique monocycle restreignante des drones à voilure fixe. Certaines stratégies de contrôle ont été proposées et validées dans des simulations en se servant de la distance moyenne entre la cible et le drone en tant que métrique de performance. Cependant, outre les paramètres de distance, la dépense énergétique du vol joue également un rôle important dans l'évaluation de la performance globale de l'avion. Nous proposons une nouvelle méthodologie qui introduit une nouvelle mesure (la dépense

## Résumé

---

énergétique). Nous comparons des méthodes existantes sur un grand échantillon des modèles de mouvement des cibles et nous présentons la comparaison des modes du mouvement cible proposées entre la simulation et les expériences sur le terrain. Grâce à cette nouvelle méthode, nous examinons la performance des stratégies de contrôle de l'état de l'art. Sur la base des simulations et des expériences sur le terrain, nous choisissons la stratégie qui a la meilleure performance selon nos critères de comparaison.

Le deuxième défi consiste dans le développement d'un algorithme de contrôle de la formation qui permettra aux robots à voilure fixe de fournir une couverture de zone large et de relayer les données dans un réseau sans fil ad hoc. Dans de telles applications, les drones à voilure fixe doivent être en mesure de régler leur distance mutuelle. Dans ce contexte, leur maniabilité réduite présente le défi de concevoir un algorithme de formation capable de régler cette distance. Pour relever ce défi, nous présentons une stratégie de contrôle distribué qui ne repose que sur des informations locales. Chaque drone a son propre agent virtuel. Il suit cet agent virtuel en effectuant la stratégie de poursuite des cibles préalablement évaluées et sélectionnées. Les règles de flocage sont mises en œuvre dans les interactions des agents virtuels. Nous montrons, dans la simulation et dans des expériences sur le terrain avec une équipe de drones à voilure fixe, qu'en utilisant cet algorithme de formation distribué, les drones peuvent couvrir une zone en créant un réseau triangulaire équilatéral et peuvent maintenir des liens de communication de bonne qualité avec des drones voisins.

Le troisième défi consiste à permettre la connectivité indépendante entre les agents mobiles terrestres à l'aide de l'équipe des drones à voilure fixe. Nous concevons deux algorithmes de contrôle distribué qui changent l'arrangement spatial et la topologie de l'interaction des drones afin de maintenir la connectivité. Nous proposons une stratégie en fonction du potentiel de champ qui adapte la distance entre les drones pour serrer et étaler la formation des drones à voilure fixe. Dans la deuxième approche, basée sur la logique du marché, les drones suppriment distributivement des liens d'interaction pour développer le graphe de formation en graphe d'arbre. Dans les simulations et les expériences sur le terrain, nous montrons que nos stratégies proposées maintiennent avec succès les connexions entre les utilisateurs terrestres indépendants.

Globalement, cette thèse présente la synthèse des algorithmes distribués pour les drones à voilure fixe avec le but d'établir et de maintenir des réseaux de communication sans fil ad hoc.

Mots clefs : réseaux de communication, drones à voilure fixe, le contrôle de la formation, le flocage, connectivité maintenace, réseau sans fil ad hoc



# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Abstract (English/Français)</b>	<b>iii</b>
<b>List of figures</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and challenges . . . . .	3
1.2 State of the art . . . . .	4
1.2.1 Bridging unicycle kinematics - target tracking . . . . .	5
1.2.2 Formation control of drones . . . . .	5
1.2.3 Connectivity maintenance . . . . .	7
1.3 Main contributions and thesis organization . . . . .	9
1.4 Publications . . . . .	10
<b>2 Aerial target tracking</b>	<b>13</b>
2.1 Introduction . . . . .	15
2.2 Target tracking strategies . . . . .	16
2.2.1 Lyapunov Guidance Vector Field (LGVF) strategy . . . . .	16
2.2.2 Bearing-only strategy . . . . .	17
2.2.3 Oscillatory control strategy . . . . .	18
2.3 Evaluation method . . . . .	20
2.3.1 Performance measures . . . . .	20
2.3.2 Control parameter optimization . . . . .	21
2.3.3 Target motion patterns . . . . .	22
2.4 Experiments and results . . . . .	23
2.4.1 Simulation results . . . . .	24
2.4.2 Field results . . . . .	26
2.4.3 Robustness to wind - simulation experiments . . . . .	33

## Contents

---

2.4.4	Robustness to wind - field experiments . . . . .	34
2.5	Conclusion . . . . .	35
<b>3</b>	<b>Formation control with fixed wing drones</b>	<b>37</b>
3.1	Introduction . . . . .	39
3.2	Formation algorithm . . . . .	40
3.2.1	Proposed method . . . . .	40
3.2.2	Discussion on formation algorithm convergence . . . . .	44
3.3	Experiments and results . . . . .	48
3.3.1	Simulation experiments . . . . .	49
3.3.2	Field experiments . . . . .	50
3.4	Collision avoidance using altitude segregation . . . . .	54
3.4.1	Proposed method . . . . .	55
3.4.2	Results . . . . .	56
3.5	Use case: Communication link adaptation . . . . .	56
3.5.1	Characterization of wireless link quality . . . . .	57
3.5.2	Adaptation algorithm . . . . .	59
3.5.3	Simulation results . . . . .	60
3.6	Conclusion . . . . .	60
<b>4</b>	<b>Connectivity maintenance</b>	<b>63</b>
4.1	Introduction . . . . .	65
4.2	Theoretical background . . . . .	67
4.3	Potential field based method . . . . .	70
4.3.1	Proposed method . . . . .	70
4.3.2	Results . . . . .	73
4.4	Distributed market-based method . . . . .	81
4.4.1	Proposed method . . . . .	82
4.4.2	Results . . . . .	83
4.5	Extensions . . . . .	88
4.6	Conclusion . . . . .	89
<b>5</b>	<b>Conclusion</b>	<b>91</b>
5.1	Main accomplishments . . . . .	93
5.2	Potential applications . . . . .	94
5.3	Future directions . . . . .	95

<b>A Appendix - Materials and experimental setup</b>	<b>97</b>
A.1 Matlab and Netlogo simulator . . . . .	97
A.2 EMANE network emulator . . . . .	98
A.3 Ebee fixed wing robots . . . . .	98
A.4 Field experiments setup . . . . .	101
<b>Bibliography</b>	<b>110</b>
<b>Curriculum Vitae</b>	<b>111</b>



# List of Figures

1.1	Illustration of search and rescue scenario; rescuers use aerial wireless network to communicate . . . . .	3
2.1	Illustration of the LGVF strategy (a) graphical interpretation of the LGVF strategy (b) the drone's trajectory while tracking the ground agent, $v_a = 0.3v_d$ . . . . .	17
2.2	Illustration of the Bearing-only strategy (a) graphical interpretation of the bearing-only strategy (b) the drone's trajectory while tracking the ground agent, $v_a = 0.3v_d$	18
2.3	Illustration of the oscillatory strategy (a) graphical interpretation of the oscillatory strategy (b) the drone's trajectory while tracking the target, $v_a = 0.3v_d$ . . .	19
2.4	Repertoire of ground agent motion patterns used in simulation, to determine the optimal parameters of the target tracking strategies. . . . .	23
2.5	Experimental setup a) controller system structure - low level control is running on PIC microcontroller based platform and high-level control is running on Gumstix microcomputer. Both units are communicating over a serial link, and a dedicated API is programmed to exchange control outputs and sensor information b) testing field . . . . .	25
2.6	Results of the simulation experiments for the LGVF, Bearing-only and Oscillatory strategy for diverse motion patterns of ground agents. The distance measure is the average distance between the drone and the ground agent, normalized to the minimum turn radius of the drone. The energy consumption measure is the relative energy consumption compared to that of the level flight, given in %. The LGVF strategy outperforms the Bearing-only strategy for all motion patterns. Oscillatory strategy performs the best for straight and circle motion patterns. However, for Levy walk motion patterns, no optimal parameters were found during optimization. . . . .	27

## List of Figures

---

2.7	Simulation results for the stability analysis of the oscillatory control strategy. The set of parameters for which oscillatory strategy leads to stable behavior is plotted with black dots. The strategy shows low robustness to variable ground agent motion patterns and speeds. . . . .	28
2.8	Results of the optimization experiments for the field experiment comparison cases. We choose two sets of parameters obtained in the optimization for our field experiments. We choose parameters that give higher relevance to the energy performance objective and parameters that give higher relevance to the distance objective (upper and lower ends of Pareto fronts). . . . .	29
2.9	Drone trajectories obtained in outdoor experiments for three ground agent motion patterns for LGVF strategy. The ground agent's speed is set to $0.3v_d$ . The starting point of the ground agent trajectory is presented with a blue dot and the end point is presented with a green dot. . . . .	30
2.10	Results of outdoor flight tests for two sets of controller parameters for LGVF strategy. Experimental results obtained with the first controller parameter set (higher relevance to the energy performance objective) are within green rectangles and the results obtained with the second control parameter set (higher relevance to the distance objective) are within blue rectangles. The experimental results support the results from simulation using the simplified platform model. . . .	31
2.11	Results of outdoor flight tests for bearing-only strategy compared to results of the field experiments for LGVF strategy. The trajectory of the drone is shown in a). In b) we show the results of three experiments in the field in which chattering did not disrupt significantly the drone's flight. The experimental results support the results from simulation using the simplified platform model. . . . .	32
2.12	Experimental results for the bearing-only strategy, a) controller output (turn rate) b) angle between drone course and distance vector. In the outdoor experiments, the drone is following a ground agent performing straight line motion, $v_a = 0.3v_d$ .	32
2.13	Simulation results of experiments that examine robustness to constant wind. We run experiments for four different orientations between target motion and wind direction. The LGVF strategy shows better robustness to the constant wind than the bearing-only strategy . . . . .	33
2.14	Results of outdoor experiments in varying wind conditions. Experiments in low constant wind conditions (2 m/s) deviate less than 10% from the results obtained in simulation for the same wind speed. Experiments in high wind conditions also show good agreement with simulation results. . . . .	34

2.15 Comparison table showing the performance of the strategies for diverse ground agent motion patterns in simulation and field experiments. Grades are given according to the performance metrics presented in the paper. The best performance is given grade 3, and the worst grade 1. If a strategy is graded 0, it implies that the strategy showed unstable behavior. . . . . 36

3.1 a) Schematic Illustration of drone following virtual agent. b) Result from real world experiment involving one fixed wing drone and a virtual agent, illustrating the motion of a fixed wing drone flying at 10 ms<sup>-1</sup> and following its virtual agent moving at 1 ms<sup>-1</sup>. . . . . 41

3.2 Schematic illustration of the distance control module. Alignment, attraction/repulsion and navigation force are acting on the virtual agent. . . . . 42

3.3 Drones need to synchronize their phase angle with their neighbors. They synchronize their phase angles by altering their forward speed in a limited interval. In this way a formation of drones is converging to the formation of their respective virtual agents. . . . . 44

3.4 Deviation from equilateral triangular formation for different sample times. Mean deviation energy and standard deviation is presented. For each sample time, experiment is conducted in 50 trials. We observe that formation deviates from ideal equilateral lattice for higher sample time values. To achieve stable coherent formation, agents should exchange on average 10 messages per second. . . . . 49

3.5 Average distance between drones and lattice energy deviation in the formation. Experiment is conducted with 10 drones. Drones regulate their distance based on the desired distance, set by the user. a) Mean distance for the formation control b) Average phase angle deviation. c) Lattice energy deviation . . . . . 51

3.6 Different phases in distance adaptation of the flying robot formation. Drones start from random positions, synchronize their motion, form a uniform formation and increase/decrease inter-drone distance in a decentralized and distributed way. . . . . 52

3.7 A snapshot of the field experiment showing 3 eBee drones, marked by dashed circles, keeping the distance of 30m between their neighbors. . . . . 53

3.8 The motion paths of 3 drones in field experiment in which they keep distance of a) 30m and b) 50m with their neighbors. Motion paths are plotted relative to the launching point of the experiment (0,0) denoted by a black cross. . . . . 54

## List of Figures

---

3.9	Mean distance and phase angle measurements for 3 drones in the field experiment. The desired distance profile is stored in drones memory before the experiment. . . . .	55
3.10	Minimum drone 3D distance, drones run collision avoidance algorithm and formation control algorithm in which they should keep 20m distance between their neighbors. Drones loitering radius is 30m. The experiment is conducted in 100 simulation runs with 5 drones. . . . .	56
3.11	Altitude profiles of all drones during one simulation experiment . . . . .	57
3.12	Link quality, RSSI and SNR measures obtained in field experiments with 2 fixed wing drones. . . . .	58
3.13	Schematics of the link quality adaptation layer. . . . .	59
3.14	Fixed wing drones are adapting the quality of communication link between their neighbors. The desired communication link quality is set by the user. We show that drones are able to adapt the formation to achieve desired link quality. They run formation algorithm combined with adaptation layer. Adaptation layer controls the speed of adaptation to the desired link quality based on the level of synchronization between robots. If robots are not well synchronized, adaptation layer will slow down the adaptation to allow them to synchronize and then it will speed up the adaptation when robots reach desired level of synchronization. . . . .	61
4.1	Distance between ground node's center of mass and virtual agent's center of mass. The experiment is run in Emame emulator, 10 runs. Drones formation follows the ground agents with average of 10m distance between centers of mass. . . . .	74
4.2	Tracks of virtual agents, ground nodes and drones in a single emulation experiment. Agents positions and drone positions are shown in three experiment time instances. Virtual agents closely follow group of ground nodes, drones synchronize their phase and create the same formation as their respective virtual agents. . . . .	75
4.3	Tracks of virtual agents, ground nodes and drones in a single emulation experiment. Agents positions and drone positions are shown in three experiment time instances. Virtual agents closely follow group of ground nodes, drones synchronize their phase and create the same formation as their respective virtual agents. . . . .	76



4.4	Distance between ground node's center of mass and virtual agents center of mass in field experiment. The experiment is run with four drones and two ground nodes. The drone formation follows the ground agent formation with less than 10m distance between centers of mass. . . . .	77
4.5	Tracks of virtual agents, ground nodes and drones in a field experiments. The field experiment results show that drones can successfully follow a group of ground nodes and translate the formation. . . . .	77
4.6	Algebraic connectivity of ground-aerial connectivity graph in relation to the maximum distance between any pair of ground nodes. As ground nodes are spreading, they move apart, interaction links are elongating. Connectivity is maintained up to 750m which is three times higher distance than communication radius. . . . .	78
4.7	Consecutive snapshots of the topology of ground-aerial connectivity graph. Virtual agents are symmetrically adapting the inter-agent distances to spread the formation and adapt to the motion of ground nodes. . . . .	79
4.8	Field experiment with four drones and two ground nodes. Ground nodes move apart to the edges of the testing field. Swarm of drones receives their positions and increases inter-agent distance to spread the formation. Drones successfully synchronize their phases and create equilateral triangular formation. . . . .	80
4.9	Maximum distance between any pair of ground nodes for different number of ground agents. The drone number is fixed to 10. We show that performance of algorithm is invariant on the number of ground agents, since links are expanding symmetrically. . . . .	81
4.10	Algebraic connectivity for the auction based method. The results are shown for 10 simulation experiments, with 10 drones and 5 ground agents. By changing topology of the interaction graph, drones can extend the aerial network and cover larger distances between ground nodes. . . . .	84
4.11	Consecutive snapshots of the simulation experiments for auction based method. As ground nodes are moving apart, the interaction topology is transforming to a spanning tree, and links elongate to stretch the network. Such interaction topology allows higher distances between ground nodes than the equilateral triangular lattice topology. . . . .	85
4.12	Comparison between potential field based method and auction based method.	86

**List of Figures**

---

- 4.13 Emulation experiment for auction based method with 4 drones and 2 ground nodes. We show 4 consecutive snapshots a)3s b)66s c)83s and d)133s. In the first two snapshots, drones are running potential field based method and stretching links of equilateral triangular lattice. In the third snapshot, the auction starts, interaction links are deleted and drone formation extends to a chain. In fourth snapshot, drones are in chain state, they are moving towards furthest neighbour and gradually extending communication links. . . . . 87
  
- A.1 Comparison of real hardware setup and emulation setup . . . . . 99
- A.2 Ebee platform, wings can be detached from the main body and it can be folded for easy transport. . . . . 99
- A.3 Hardware setup consists of Gumstix Linux microcomputer and extension board which has an access to the autopilot data and Gumstix ports. . . . . 100

# 1 Introduction



I would like to thank to Darko Draskovic for the artwork in this thesis.

## 1.1 Motivation and challenges

Being connected to a global communication network has become one of the imperatives. In many situations we require to receive information about our environment and we communicate information to other users. We exchange bulky data such as voice and video messages and we require good coverage and high throughput channels. Our communication devices, smart phones and computers, are connected over wired networks, cellular, wireless and satellite networks.

Such communication networks depend on rigid, heavy infrastructure. In highly populated, urban areas the coverage of these networks is satisfying but in mountainous, unpopulated areas coverage can be poor. In some cases communication infrastructure can be damaged. Natural disasters, such as forest fires or earthquakes, can heavily damage power lines, network beacons and disable communication channels. Particularly in these cases it is very important that human users, especially rescuers, military and police can communicate, can collect and share information.

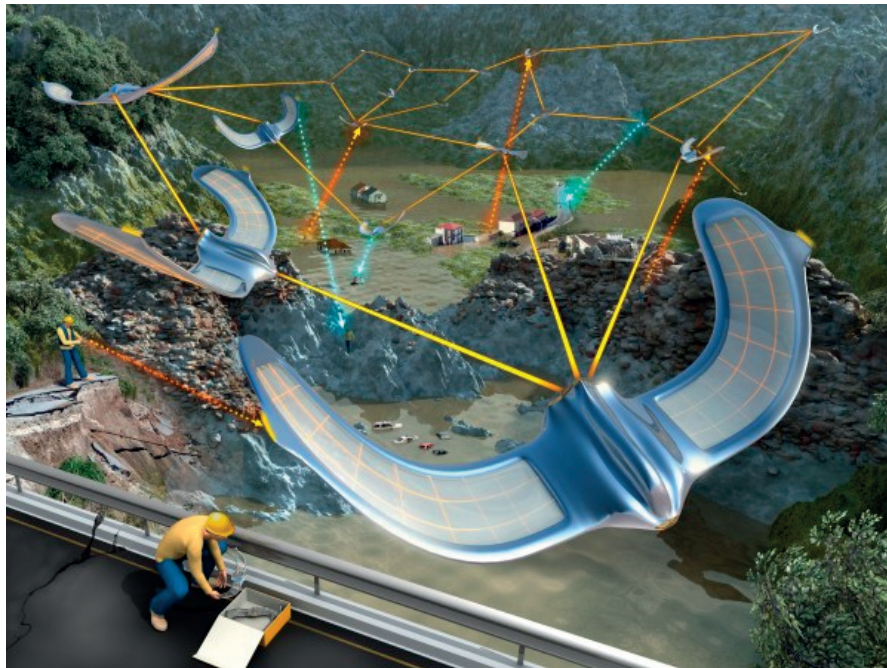


Figure 1.1: Illustration of search and rescue scenario; rescuers use aerial wireless network to communicate

To leverage these problems, there is a need to develop an easy-portable system that can be quickly deployed to establish dedicated wireless network for ground users. In this thesis we propose distributed fixed wing drone team which can be used to establish wireless ad-hoc

## Chapter 1. Introduction

---

network. Each drone in the team carries a wireless beacon and establishes connection with its intermediate neighbors. Drone team adapts its shape and topology to the ground users motion. The drones are lightweight and network can be established over large areas within minutes.

Among different designs of flying platforms we choose to implement formation control and connectivity maintenance algorithms on fixed-wing drones. Such platforms have higher battery life, they are prone to the wind disturbances, quickly deployable and can cover larger areas. These features have high importance in a fast and reliable disaster response. On the other hand, compared to a hovering platform, a fixed wing drone has more constrained kinematics. Fixed-wing drone cannot hover or perform sharp turns. This kinematics constraint presents a challenge for control and collective flight of such drone teams.

Another challenge is to design a decentralized formation algorithm for nonholonomic fixed-wing drones that will allow them to regulate distances to their neighbors. Fixed-wing drones should also adapt their spatial arrangement to the motion of ground users, they should follow their motion, expand and shrink to keep ground users connected at all times. Since it is not easy to predict the motion of ground users, they are considered as independent agents and drones need to adapt reactively to their motion.

Having these challenges, this thesis answers following research questions:

- What is the best way for fixed wing drones to follow slow moving ground agents?
- How can fixed wing drones fly in a formation in which they control distance to neighboring drones?
- How can fixed wing drones fly in a formation in which they keep independently moving ground agents connected in a wireless ad-hoc network?

### 1.2 State of the art

This thesis deals with problems of target tracking, formation control, area coverage and communication maintenance in multi-drone systems. In this section, the most relevant literature in the aforementioned fields is presented.

### 1.2.1 Bridging unicycle kinematics - target tracking

To follow slow moving target or to fly above some predefined area, fixed-wing drones need to perform flight patterns to slow down their overall ground speed. Fixed-wing drone kinematics can be modeled as Dubin's vehicles, unicycle vehicles with fixed forward speed [28]. In general, this problem can be defined as target tracking problem, where target moves slower than the drone. The target can be a ground agent or a virtual agent. In many current control strategies for target tracking, researchers assume that the trajectory of the ground agent is known at least to some extent. Dynamic programming and model predictive control is proposed by [63] and [47] to determine a control strategy to track a ground agent. The aim of this control strategy is to minimize the drone to ground agent distance using *a priori* knowledge about the ground agent motion. In [9] a grid-based Markov Decision Process (MDP) is proposed to determine the optimal drone path offline for different target speeds. In [7] authors assume that target is performing Brownian motion and design a stochastic control algorithm to track the moving target. In some applications, the motion of the target is not easy to predict. In these circumstances, control strategies rely solely on the current relative position and relative orientation between the ground agent and the drone. In [31], authors propose Lyapunov Guidance Vector Field (LGVF) strategy to follow a target performing orbital flight path. In [71] authors propose Bearing-only strategy that relies only on the relative bearing between the drone and the ground agent. In [51] authors propose oscillatory control to produce left-right (sinusoidal like) flight path while following the ground agent. To assess target strategies performance, in [89, 12] authors compare target tracking strategies in simulation experiments using only distance between the ground agent and the drone as a measure of target tracking performance. These comparison studies do not take into account energy expenditure of the proposed strategies despite being one of the main aspects of drone missions.

### 1.2.2 Formation control of drones

To establish aerial communication networks and distributively adapt communication link quality, drones need to fly in a formation and need to cover an area with their communication sensors. In multi-agent systems, strategies that solve area coverage problem are based on a) computational geometry based methods, b) gradient based and potential field based methods.

Methods based on the computational geometry [21, 48, 54, 16, 50, 72] propose an approach to the area coverage problem where convergence can be mathematically proven and the performance of the algorithm can be formally analyzed. These methods rely on the prior knowledge about the environment shape, important environment sectors that need to be

covered and information about all agents' positions.

On the other hand, gradient and potential field based methods allow more reactive behaviors and better adaptation to the changing environment. These strategies share a similar approach with variations in defining cost functions [66, 74]. Methods also vary in mapping between environment observations and potential fields and amount of prior knowledge that is needed to achieve distributed coverage. We focus on methods that rely only on local observations and do not require prior knowledge about agents' motion.

A method that relies only on local interactions with the environment and neighboring agents is proposed by Howard et. al. [41]. Agents are represented as virtual particles that are subject to virtual forces. The forces repel agents from their neighbors and from obstacles in the environment. Using this simple reactive rules authors managed to deploy ground robots in unknown indoor environment and maximized the covered area. This strategy has been formalized within the framework of physicomimetics, introduced by Spears et. al. [75] The main idea of the physicomimetics framework is to define a trajectory of each agent by the sum of virtual forces acting on its virtual mass. Forces are based only on a relative position between agent and the environment. Methodologies that rely on following a gradient of virtual potential fields and virtual forces are less conservative in approach then methods that rely on computational geometry and locational optimization. These methods are lightweight in computation, based only on few simple rules. They allow purely reactive approach and are based only on local information. However, these methods suffer from oscillations of energy between agents that can sometimes lead to unstable behaviors.

To leverage the problem with oscillations and formation stability, we further examine theoretical findings in potential field based formation control. A theoretical approach to formation control has been proposed by [38] where authors apply potential field control algorithms to make single integrator agents to arrange themselves on a circle and keep equal distances to each other. In this approach they showed that formation can be reached without oscillations in the agent position. A particularly interested work was presented by Olfati-Saber [60]. He proposed flocking algorithm [67] based on the bounded attraction/repulsion forces and consensus on velocity and he proved that if point mass, holonomic agents follow this control algorithm, their equilibrium formation will be an equilateral triangular lattice.

Since fixed-wing drones are highly nonholonomic vehicles we examine the literature in formation control of nonholonomic vehicles. The literature is mostly focused on unicycle vehicles, in which turn rate and speed control robot's position and orientation. In [55] showed



a theoretical framework for achieving circular formations of unicycles in which they keep equal distance to each other. In [82], authors proposed a method to control the flock of unicycle vehicles. They achieved stable flocking in which agents only aligned their speed and could fly in the coherent flock. If speed of the unicycle vehicles is bounded, a formation control problem becomes more challenging. In [58] authors proposed geodesic control to control a flock of unicycle vehicles with a fixed forward speed. They only showed a velocity alignment but they could not control a spatial arrangement of vehicles. In [46] authors showed in simulation and mathematical proved leader-follower formation of two unicycles with unit speed and they extended their approach for  $n$  vehicles. The drawback of their approach is that the whole formation always has to have an unit speed. In all presented approaches, a formation algorithm for a team of unicycle vehicles with fixed forward speed, in which they can adapt distances to their neighbors and average team speed has not yet been proposed.

### 1.2.3 Connectivity maintenance

Communication maintenance strategies are divided in two main groups, strategies that require global network properties, such as knowledge about network topology and strategies that rely on local observations within agents neighborhood, such as neighbor positions and communication link properties. Methods that rely on global properties of the network are mainly based on the graph theory [34] and they use mathematical formalisms such as a weighted Laplacian matrix, a connectivity matrix, Fiedler values to examine connectivity properties and to design control algorithms. The goal is to detect possible violations of the connectivity or to improve communication quality. In [56] authors presented an approach which relies on estimates of the network topology which provides the agent with a rough picture of the network structure. Furthermore, they use gossip algorithms [15] and the distributed market-based control [19] that allow link deletions without violating connectivity. This paper also investigates practical issues with implementing this methodology on a real system of a homogeneous team of ground robots. Similar approach was presented by Stump et. al. [79] in which authors developed an algorithm to control a team of robots to maintain the communication bridge between stationary robot in the environment with obstacles. Their approach relies on calculating a Fiedler value of the weighted Laplacian matrix which describes communication interactions between all robots and they use a  $k$ -connectivity matrix to detect which robots can interact through  $k$  or less intermediary robots. In approach presented by Cortez et. al. [22], authors are defining motion constraints of agents in a heterogeneous group consisted of sensing agents and relay nodes. By using distance based connectivity properties of the network, they define the control of relay nodes to extend and maintain communication with sensing nodes. The

main advantages of presented methods are the fact that they provide valuable insight in the topology of the network and allow the usage of the known distributed search algorithms on graphs to identify communication bottlenecks in the network. The main disadvantage is that they rely on a global property of the network which requires extensive communication to spread information about the topology to all agents in the network.

Strategies that rely on local measurements of the communication links quality are based on the potential field based control or the gradient-based control. They use local observations to derive potentials and gradient based control to guide the agents to stay connected with their neighbors. Usually authors propose unbounded potential field functions to maintain a communication link while performing formation control tasks. In [25], Dimarogonas et. al. proposed a method to maintain all the links in already established communication network by defining a bounded potential field function. The control scheme (i) maintains the edges of the graph which were set in initial conditions and (ii) drives agents to a common, known position in state space. In the paper proposed by Gil et. al. [33], the approach to maintain connectivity using gradient based control and local measurements has been extended to a heterogenous group of indoor flying robots and ground sensors. Flying robots are used in this method to improve the communication quality between static ground nodes. Their cost function is defined by Signal to Interference Ratio between neighboring robots and control algorithm follows distributed gradient descent on the cost function. The approach is demonstrated in reality using few static nodes and 3-5 quadrotors in an indoor experiment. Another application of maintaining communication in heterogenous systems using a gradient based approach is proposed by Hsieh et. al. [42]. Authors propose aerial-ground robots network where both groups of robots perform a gradient based control to maintain communication. The authors assume that the network topology of the ground robots is preserved throughout the experiment. In [83], Tardioli et. al. used a virtual forces approach to maintain connectivity between a fixed node and ground robots. They used spring forces model to maintain fixed topology while performing an exploration task. These strategies are more suited in applications where a network topology changes rapidly, in heterogeneous swarms where agents differ in mobility and controllability and agents should react in a reactive mode to the changes in the environment. A communication maintenance problem becomes extremely challenging if the communication should be maintained between independently moving agents. Such problem has been addressed in [32] where authors assume that they can only influence position of the robotic relay nodes and cannot control the motion of sensing nodes. They collect the data of all nodes in the network (position) and they calculate optimal position of robotic nodes to keep sensor nodes connected. The drawback of this approach is that it is centralized and

extremely computationally heavy approach.

### 1.3 Main contributions and thesis organization

This thesis proposes novel solutions for formation control and connectivity maintenance in ground-aerial wireless networks. Taking into account constraints of fixed-wing robots and independent motion of ground users we develop reactive formation algorithms in which drones adapt distance to their neighbors, adapt communication link quality and change topology to maintain connectivity between ground users. We validate all our algorithms in simulation and on real drones.

Since fixed wing drones have to maintain a minimal forward speed to remain airborne, the question is how to slow down these drones and how to make them follow slow moving targets. Possible solutions could be to make them circle around the desired waypoints or to perform a ribbon-like motion. The question is: which of these two strategies is better for our application and what should be criteria to compare such strategies. Therefore, an evaluation methodology for target tracking strategies is developed. A new measure, energy expenditure of the flight is introduced. We select three strategies and compare them in low wind scenarios for three types of target motion, both in simulation and in field experiments. In order to make a fair comparison between these strategies that have different parameters, we run a multi-objective optimization (MOO) experiment to find the optimal parameter configuration for each strategy in the comparison experiments. The objectives to be optimized are the average distance to the ground agent and the energy expenditure of the flight. Using this technique, we obtain Pareto fronts for each strategy and each comparison case, and use them to examine whether one strategy dominates the others when all strategies have optimal parameter settings. We choose two extreme sets of control parameters and conduct field experiments to compare to the simulation results. This resulted in an evaluation framework that allowed us to choose an energy efficient strategy that is also robust in field experiments.

Second research question was to examine how can fixed wing drones fly in a formation in which they can control distance to their neighbors. Existing approaches do not address the formation control for nonholonomic unicycle robots with fixed forward speed for controlling the distance between their neighbors. We propose a formation algorithm, where drones rely on the local information communicated from their single hop (or direct) neighbors. To guarantee that the forward speed of the drones are always greater than the stall speed, a virtual agent is defined for every drone and drones are controlled to continuously follow and circle

around their virtual agents. Drones emulate the motion of the virtual agent and follow it by performing spiraling motion using Lyapunov Guidance Vector Field algorithm [31]. Formation algorithm is then implemented between virtual agents. To map the formation of the virtual agents to the formation of drones, drones also synchronize their phase angles. We test the performance of our method in simulation and in field experiments.

As the third contribution, it is studied how fixed-wing drone formation serves as a connectivity relay between independently moving ground agents. We proposed two approaches to solve this problem. In the first approach, each drone receives information from independently moving agents, and calculates desired distance that it should keep with its neighbors. This potential field based method allows the network of flying robots to stretch and compress to the motion of the independently moving ground agents. But such structure stretches symmetrically and at some point if agents move far apart it will not be able to keep the connectivity. Therefore, we propose another method in which drones negotiate which links they should break in order to transform their interaction network to a tree graph. This method requires more communication between agents but it also allows much longer multi-hop communication links between independently moving agents.

The thesis consists of 5 chapters. In the first chapter, motivation, main challenges of this thesis and literature review are presented. Second chapter deals with evaluation of target tracking strategies to bridge unicycle dynamics of fixed-wing drones. In the third chapter, a formation algorithm for distance adaptation and area coverage fixed wing robots is presented. Fourth chapter introduces connectivity maintenance algorithms in ground-aerial swarms. In the fifth chapter, the concluding remarks and future directions are given. In the Appendix, materials used in field experiments and simulation and emulation platforms are described.

### 1.4 Publications

During the thesis work, following papers have been published:

- Hauert, S., Leven, S., Varga, M., Ruini, F., Cangelosi, A., Zufferey, J. C., & Floreano, D. (2011, September). Reynolds flocking in reality with fixed-wing robots: communication range vs. maximum turning rate. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (pp. 5015-5020). IEEE.
- Varga, M., Zufferey, J. C., Heitz, G. H. M., & Floreano, D. (2015). Evaluation of control strategies for fixed-wing drones following slow-moving ground agents. *Robotics and*

Autonomous Systems, 72, 285-294.

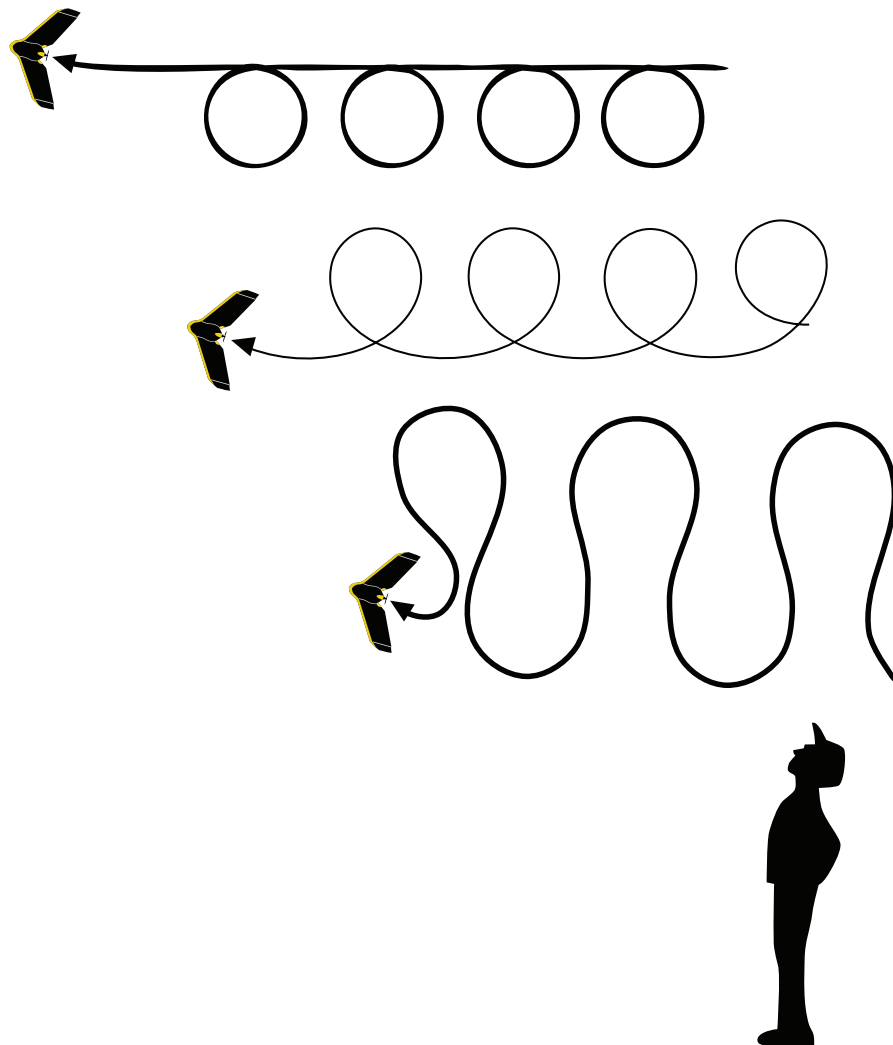
- Varga, M., Basiri, M., Heitz, G., & Floreano, D. Distributed Formation Control of Fixed Wing Micro Aerial Vehicles for Area Coverage. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on (pp. ). IEEE.

Following papers are under submission for possible publication:

- Varga, M, Basiri, M., Stovold, J., Heitz, G. & Floreano, D. Communication maintenance in drone networks. Under submission.



## 2 Aerial target tracking



This chapter is based on following publication:

Varga, M., Zufferey, J. C., Heitz, G. H. M., & Floreano, D. (2015). Evaluation of control strategies for fixed-wing drones following slow-moving ground agents. *Robotics and Autonomous Systems*, 72, 285-294.



## 2.1 Introduction

Fixed wing drones have been extensively used to track moving targets on the ground [84, 26, 64]. Since the speed of the target is usually slower than the speed of the drone, drones have to perform manoeuvres such as spiralling or sinusoidal motion along a target's trajectory to stay in the close vicinity of the target. When choosing the appropriate strategy, one is usually concerned with how well the drone tracks the target. In [63, 47, 9] authors used the distance between drone and ground agent positions as the performance metric they wished to optimize.

Energy performance of flight has not yet been included in any comparison of strategies, despite the fact that it plays a major role in determining flight duration. Energy performance is related to the type of trajectory that the drone performs while tracking a ground agent. While a fixed wing drone performs sinusoidal or spiral like motion, it has to change the bank angle to turn, which lowers the lift force. If we assume that the drone is performing level flight, to maintain the altitude the drone has to apply an additional thrust to counteract the reduction in the lift force. This additional thrust increases the energy consumption of the flight. Hence by measuring the turn rate of the drone while performing level flight, the additional energy required to perform a given manoeuvre is estimated.

Besides introducing a new metric, to improve the comparison of the strategies an experimental method should be carefully designed to capture most of the differences in performance. In the literature, strategies are usually compared in test cases where the target performs a straight line motion pattern [89]. Such experimental design does not allow one to identify whether the performance of the strategies varies with the change in the target motion patterns. Furthermore, each control strategy that is being compared has different parameters that can be tuned in order to modify the performance of the strategy. In [89, 71] authors set parameters to fixed values before the experiment and they do not tune these parameters to the same objectives. Lastly, comparisons proposed in the literature [89, 71, 12] are conducted only in simulation. Usually simulation experiments cannot capture performance degradation due to sensing and actuation noise from their simplified mathematical models.

In this chapter, we introduce a new measure: energy expenditure of the flight, and we propose a new methodology to compare target tracking strategies. We choose to compare three strategies following our experimental methodology. In the first strategy, the Lyapunov Guidance Vector Field (LGVF) strategy proposed by [31], the drone follows a ground agent by performing an orbital flight path. The second strategy, the Bearing-only strategy proposed by [71], relies only on the relative bearing between the drone and the ground agent and the drone performs a

combination of circular and straight fight paths. In the third strategy, the oscillatory control strategy proposed by [51], the drone follows a left-right (sinusoidal like) flight path while following the ground agent.

We design comparison experiments in low wind scenarios for three types of ground agent motion patterns, both in simulation and in field experiments. Each strategy has one or two control parameters that influence target tracking performance. In order to make a fair comparison between these strategies that have different parameters, we run a multi-objective optimization (MOO) experiment to find the optimal parameter configuration for each strategy in the comparison experiments. The objectives to be optimized are the average distance to the ground agent and the energy expenditure of the flight. Using this technique, we obtain Pareto fronts for each strategy and each comparison case, and use them to examine whether one strategy dominates the others when all strategies have optimal parameter settings. We choose two extreme sets of control parameters and conduct field experiments to compare to the simulation results.

## 2.2 Target tracking strategies

In this section an overview of target tracking strategies is presented. These target tracking strategies will be later compared using our methodology.

### 2.2.1 Lyapunov Guidance Vector Field (LGVF) strategy

The LGVF strategy, proposed by [31], determines a commanded turn rate of the drone based on the distance between the ground agent and the drone by constructing a circular vector field around the position of the ground agent. The drone kinematic model relative to the motion of the ground agent is given by:

$$\begin{aligned}\dot{x}^{drone} &= v \cos(\phi) - \dot{x}^{ga} \\ \dot{y}^{drone} &= v \sin(\phi) - \dot{y}^{ga} \\ \dot{\phi} &= \omega\end{aligned}\tag{2.1}$$

where  $v$  is drone forward speed,  $\omega$  is drone turn rate,  $\phi$  is drone orientation and  $[x^{ga}, y^{ga}]$  is ground agent position. Drone velocity vector is calculated using Lyapunov vector field. Lyapunov function is defined as:

$$V(r) = (r^2 - r_d^2)^2\tag{2.2}$$

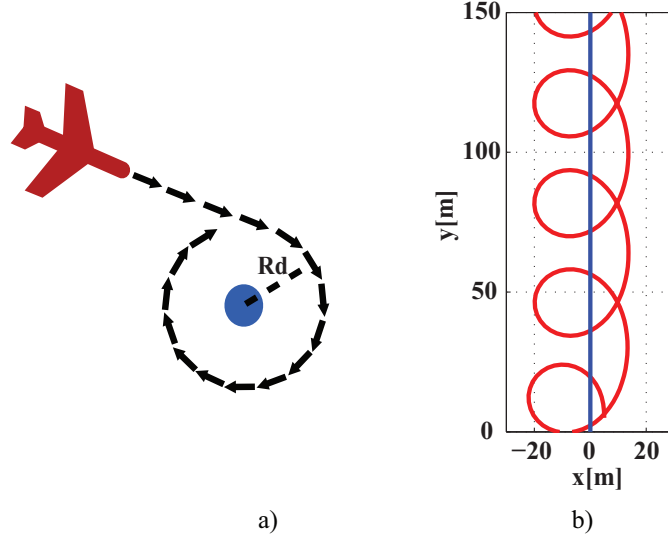


Figure 2.1: Illustration of the LGVF strategy (a) graphical interpretation of the LGVF strategy (b) the drone's trajectory while tracking the ground agent,  $v_a = 0.3v_d$

where  $r = \|p_{drone} - p_{ga}\|$  is  $L^2$  norm and  $r_d$  is a desired vector field radius. The time derivative of Lyapunov function  $V$  is nonpositive if desired drone relative velocity  $[\dot{x}_d, \dot{y}_d]$  is chosen by a guidance vector field given by:

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = -\frac{-v}{r(r^2 + r_d^2)} \begin{bmatrix} (r^2 - r_d^2) & 2rr_d \\ -2rr_d & (r^2 - r_d^2) \end{bmatrix} \begin{bmatrix} x^{drone} \\ y^{drone} \end{bmatrix} \quad (2.3)$$

The field guides the drone to perform a circular loitering pattern with radius  $r_d$  if the ground agent is still. On the other hand, if the ground agent starts to move, the vector field also moves with the agent and guides the drone to perform an orbital motion along the trajectory of the ground agent, as shown in Fig. 3.1.

### 2.2.2 Bearing-only strategy

The *Bearing-only* strategy, proposed by [71], determines commanded turn rate of the drone using only the relative bearing between the ground agent and the drone. It utilizes a simple control strategy using sliding mode control. If drone is ahead of the ground agent, it turns back with the maximum turn rate. If the drone is behind the ground agent, it adjusts its heading vector to align to the distance vector between drone and ground agent. Formally, if we define a distance vector,  $\mathbf{d}$  and a drone orientation vector as  $\mathbf{p}_r$ , then the control output (turn rate) is

## Chapter 2. Aerial target tracking

determined using following function:

$$|u| = \begin{cases} \omega_{max} f(\mathbf{d}, \mathbf{p}_r) & \text{if } |d| \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.4)$$

Generally, function  $f(\mathbf{d}, \mathbf{p}_r)$ , is defined as:

$$f(\mathbf{d}, \mathbf{p}_r) = \begin{cases} 0 & \beta = 0 \\ 1 & 0 < \beta \leq \pi \\ -1 & \pi < \beta < 2\pi \end{cases} \quad (2.5)$$

where  $\beta$  is the angle between the distance vector and the drone orientation vector, always measured counterclockwise from the distance vector. Graphical interpretation of the control strategy, together with resulting behavior of the drone is shown in Fig. 2.2.

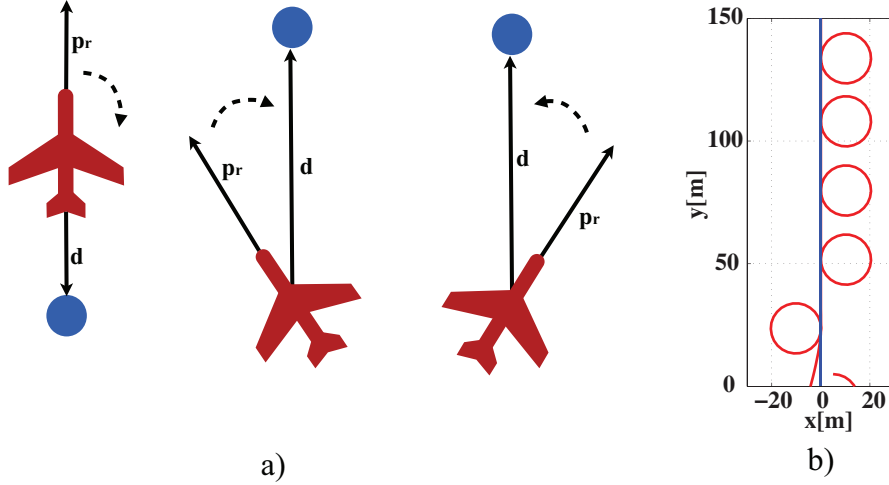


Figure 2.2: Illustration of the Bearing-only strategy (a) graphical interpretation of the bearing-only strategy (b) the drone's trajectory while tracking the ground agent,  $v_a = 0.3 v_d$

### 2.2.3 Oscillatory control strategy

The Oscillatory Control Strategy, introduced by [51], generates a sinusoidal (left-right) motion of the drone along ground agent's trajectory. To generate sinusoidal motion of the drone along

the ground agent's trajectory, the desired turn rate is defined as a sinusoidal function given by:

$$\begin{aligned} u(t) &= A \sin(\phi(t)) + B + C \sin(2\phi(t)) \\ \dot{\phi}(t) &= \omega_0 + k_\phi(\phi(t) - \phi_0) \end{aligned} \quad (2.6)$$

where  $\omega_0$  is angular frequency of the desired turn rate and  $k_\phi$  is phase gain.

The task of the tracking controller is to determine the parameters  $A, B, C$  that will guide the drone to align its average speed and position with the ground agent speed and position. The center of oscillation (CO) of the drone's trajectory is defined as an integral over the drone's trajectory. A virtual target is on the distance  $e$  from the CO and lies on the line which connects drone's position and CO.

$$\begin{aligned} A &= -k_e(e^* - e) \cos(\alpha) - k_v(v_{CO} - v_a) \\ B &= k_\alpha \alpha + v_{CO} \frac{\sin(\alpha)}{e} - v_a \frac{\sin(\theta)}{e} \\ C &= 0 \end{aligned} \quad (2.7)$$

where  $k_v$  is a speed gain of the control law,  $v_{CO}$  is velocity of the CO,  $\alpha$  is angle between CO and the target, measured from the virtual target and  $k_e$  is the distance gain. We found in preliminary simulation experiments that these two gains have a major influence on the performance and convergence of the oscillatory tracking strategy. The principles of the tracking law are depicted in Fig. 2.3.

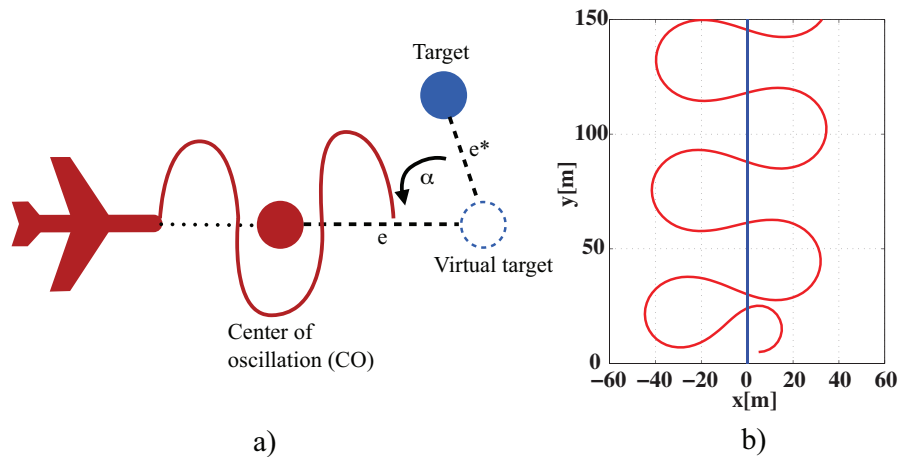


Figure 2.3: Illustration of the oscillatory strategy (a) graphical interpretation of the oscillatory strategy (b) the drone's trajectory while tracking the target,  $v_a = 0.3v_d$

## 2.3 Evaluation method

In this section we describe the novel experimental methodology to evaluate the target tracking strategies. We explain our choice of motion patterns and control parameters used to compare the strategies in simulation and in reality.

### 2.3.1 Performance measures

In this study we compare strategies for diverse ground agent motion patterns. We compare strategies on two measures: average distance to the ground agent and energy consumption. Average distance is a commonly used measure in the literature to assess the performance of target tracking strategies. This measure  $\eta_{distance}$  shows how close, on average, the drone is to the ground agent relative to its minimum turning radius. The measure is defined by averaging the distance between the ground agent and the drone over the entire flight interval.

$$\eta_{distance} = \frac{\sum_{n=1}^{N_s} \|\mathbf{p}^{drone} - \mathbf{p}^{ga}\|}{N_s} \cdot \frac{1}{R_{min}}, \quad T_{flight} = N_s \cdot T_s \quad (2.8)$$

where  $\mathbf{p}^{drone}$  is the drone's position vector,  $\mathbf{p}^{ga}$  is the ground agent's position vector,  $R_{min}$  is the drone's minimum turning radius,  $T_s$  is a sample time and  $N_s$  is the number of algorithm iterations.

The second measure that we introduce in this study quantifies the energy consumption of the drone when it is performing complex trajectories. In our energy model, we assume that wind does not disturb the motion of the drone. When performing turns, the drone needs an additional thrust to maintain the desired altitude. According to [78] the additional power required while performing coordinated turns is proportional to the square of the turn rate. The energy is determined by integrating the power over the flight interval:

$$E_{turn} = \int_0^{T_{flight}} P_{turn}(t) dt \quad (2.9)$$

where  $P_{turn} = f(\omega^2)$  is the propeller power required to perform the coordinated turn defined

as:

$$P_{turn} = \frac{T \cdot v}{\eta_p} \quad (2.10)$$

where  $\eta_p$  is the efficiency of the propeller,  $v$  is the forward speed of the drone, and  $T$  is the thrust force. In order to determine the thrust force,  $T$ , we examine the aerodynamic equilibrium, where the thrust force should be equal to the drag force to maintain constant airspeed. Additionally, the lift component should be equal to the weight of the aircraft to maintain altitude. The thrust force is a square function of the turn rate  $T(t) = f(\omega^2)$ . The relative energy consumption is defined as:

$$\eta_{energy} = \frac{(|E_{turn} - E_{straight}|)}{E_{straight}} \cdot 100[\%] \quad (2.11)$$

where  $E_{straight}$  is the energy consumption of a level straight flight and  $E_{turn}$  is the energy consumption of the drone while performing trajectory tracking.

### 2.3.2 Control parameter optimization

Tracking performance is influenced by the choice of the values of the control parameters. In order to make a fair comparison we need to determine optimal control parameter values for each strategy. We have two objectives, to minimize distance and energy measure. We use multi-objective optimization and optimize the control parameters for each comparison case. We choose standard a posteriori evolutionary algorithm, the NSGA II optimization algorithm [24] to optimize the problem defined as:

$$\begin{aligned} &\text{Minimize } \eta_{distance}(x) \text{ and } \eta_{energy}(x) \\ &\text{subject to } x \in [x_{min}, x_{max}] \end{aligned} \quad (2.12)$$

where  $x$  is a range of control parameters for each strategy. We run each optimization experiment for a population size of 100 and optimization is terminated when the spread of the Pareto front over 100 generations is lower than  $10e-4$ .

The choice of the parameter range is application driven. It is determined by a WIFI range

## Chapter 2. Aerial target tracking

---

that provides satisfying throughput for image transfer and by the range of visual overview of the ground agent. In case of the LGVF strategy, an important control parameter is the standoff radius  $r_d$ : the distance that the drone should keep from the ground agent. We set  $x = r_d \in [R_{min}, 3R_{min}]$ . For the Bearing-only strategy, we can tune the maximum value of the turn rate of the sliding mode controller as a control parameter. Therefore we set the range of the optimization variable to  $x = \omega_{max} \in [0.1 \frac{v_d}{R_{min}}, \frac{v_d}{R_{min}}]$ . For the Oscillatory strategy, we have two control parameters: speed gain  $k_v$  and distance gain  $k_e$ . We set the range for those parameters to  $k_v \in [0.01, 1]$  and  $k_e \in [0.01, 1]$ .

### 2.3.3 Target motion patterns

For each control parameter we run simulation experiments for various motion patterns of ground agents, and we measure the drone's average distance and energy consumption. We also define maximum distance between ground agent and the drone, which is defined by the WiFi range. If the distance between the ground agent and the drone is greater than maximum distance, drone does not receive position information from the ground agent and the ground agent is considered as lost. Therefore, in simulation experiments, if the drone reaches distance to the ground agent which is greater than the maximum distance to the ground agent, we assign a maximum distance to this experiment in optimization algorithm. In Figure 2.4 we present the repertoire of motion patterns used in these simulation experiments. Since the range of potential motion patterns is large, we are restricting ourselves to motion patterns that occur in the most prominent applications. We are therefore testing the performance of the strategies when ground agents are moving in a straight line with speeds that are 0.1 to 0.9 times the maximum forward speed of the fixed wing drone ( $v_d$ ). For the circular motion, the turning radius is set to  $5R_{min}$ , where  $R_{min}$  is the minimum turning radius of the drone. We choose the turning radius of  $5R_{min}$  which will allow us to repeat the field experiment multiple times during one battery lifetime of the drone. Also, we choose the value which is in between the turning radius of  $R_{min}$  (similar to the motion on the spot) and turning radius of  $10R_{min}$  (similar to straight line motion). To examine the tracking performance in following live subjects, we model ground agent behavior as a Levy walk [88] since in [87, 29] it has been shown that some groups of animals (albatrosses, reindeer, fish, humans [68, 17] ) exhibit Levy walking. We set the ground agent speed to a constant value in order to determine the robustness of the strategies for different ground agent motion patterns. Secondly, we run experiments in which we examine the strategy's robustness when the ground agent speed is increasing linearly. Simulation results presented in this figure are normalized to the minimum turning radius of the drone,  $R_{min}$  and the forward speed of the drone,  $v_d$ . We adopt a scale-



less approach to highlight the generality of our results for any arbitrary unicycle vehicle with limited turning curvature and fixed forward speed.

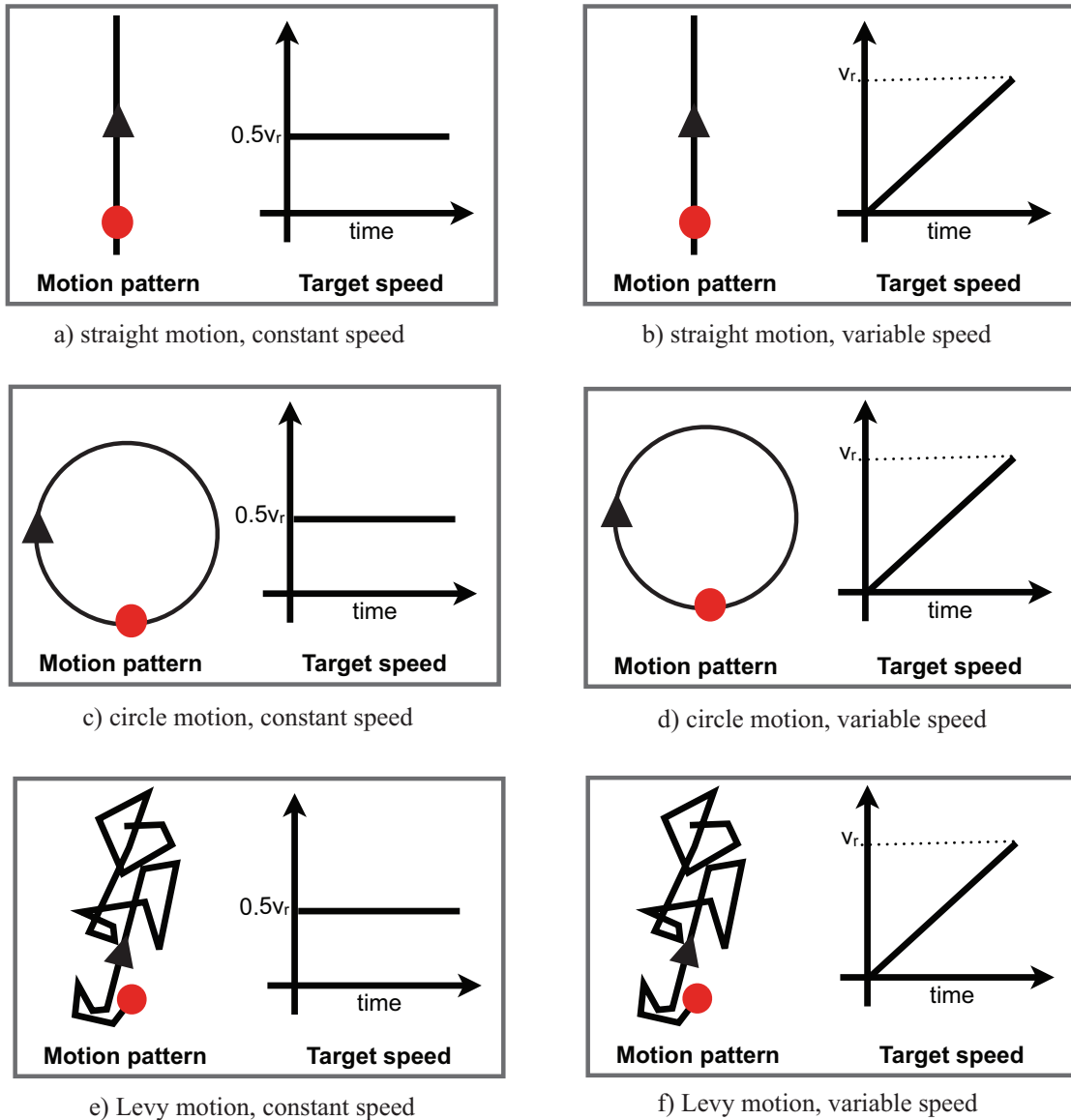


Figure 2.4: Repertoire of ground agent motion patterns used in simulation, to determine the optimal parameters of the target tracking strategies.

## 2.4 Experiments and results

We use Pareto fronts to examine the relative performance of the strategies in simulation. If a Pareto front of one strategy dominates the Pareto front of the other strategies, we can conclude that for a particular comparison case, that strategy performs better than the others for the

optimal control parameters. Simulation experiments are used as a benchmark to select two control parameter sets and test performance in the field tests. The outdoor experiments consist of deploying a fixed wing drone above a test field with a radius of 300m from the launch point of the drone Fig. (2.5b).

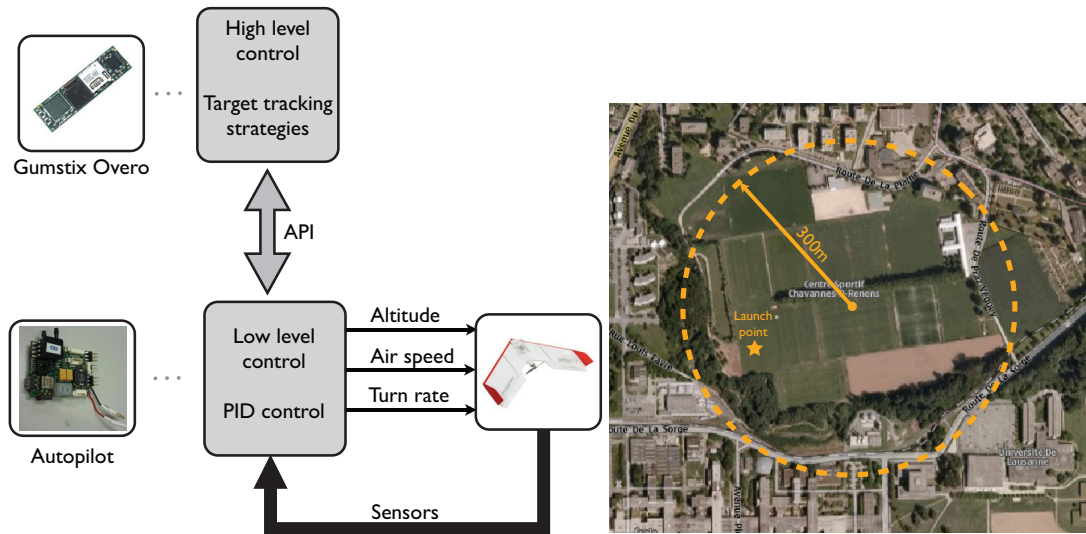
The technical properties of the platform are given in [52]. We use Swinglet platforms [80], which have 80cm wing span, total mass less than 0.5kg and the flight time is around 30 minutes. The control hardware is comprised of two parts: a low level autopilot that runs real-time low level control, and a Gumstix Linux board that runs the target tracking strategy. Low level controllers consist of several PID controllers in a 2-level cascade structure. The control inputs to the low level controllers are the desired airspeed, altitude and turn rate, and the control outputs are the thrust and the angle of the elevons (ailerons and elevator). The target tracking control strategy sets the turn rate of the low level controllers Fig. (2.5a). Altitude and airspeed are kept constant.

In each outdoor experiment, the drone tracks a ground agent whose trajectory is generated on the platform during the flight. In field tests we run the experiments in 3 comparison cases: a) a straight line motion with the speed of the ground agent set to  $v_a = 0.3v_d$ , b) a curved motion pattern with the radius of curvature set to  $r_d = 5R_{min}$  and speed set to  $v_a = 0.3v_d$  and c) a Levy motion pattern with the speed set to  $v_a = 0.3v_d$ . We run additional optimization experiments in simulation for these comparison cases. We conduct experiments in two extreme scenarios, a) we choose control parameters (among optimal parameters obtained in the NSGA optimization) that give higher relevance to minimizing distance performance and we also choose parameters that give higher relevance to minimizing the relative energy consumption. The starting point of the ground agent is the same in each experiment and the drone starts tracking from an arbitrary point that is within 30m from the initial ground agent position. The choice of parameters for the outdoor experiments is determined by the size of the testing field and is limited by the duration of the experiments.

### 2.4.1 Simulation results

This section presents the main results obtained in the simulation experiments on a kinematic model of the fixed-wing drone, as well as the results of the outdoor experiments.

Figure 2.6 shows the Pareto fronts of the distance measure vs. energy measure for the proposed strategies. Control parameters are set to the values obtained in the NSGA II optimization. Results of distance and energy performance closer to the origin of the figure indicate that this



a) Testbed (Flying platform + control hardware)

b) Testing field

Figure 2.5: Experimental setup a) controller system structure - low level control is running on PIC microcontroller based platform and high-level control is running on Gumstix microcomputer. Both units are communicating over a serial link, and a dedicated API is programmed to exchange control outputs and sensor information b) testing field

particular strategy has better performance. The LGVF strategy showed average performance for straight and circular ground agent motion patterns and the best performance for Levy motion patterns. In the LGVF strategy, a trajectory of the drone is guided by a smooth circular vector field centered at ground agent's position. This algorithm gives a commanded direction of motion to the drone, which is subtracted from the current orientation of the drone. The error between the current and commanded orientation of the drone is fed into a PD controller to obtain turn rate for the drone. In this way, there are no abrupt changes in the turn rate values and the drone smoothly changes the direction of motion. This results in better tracking performance, according to both the energy and distance measures, than the Bearing only strategy. This strategy shows the worst performance in all simulation test cases. The reason for this discrepancy in performance can be explained by the "smoothness" of control: the Bearing-only strategy uses a sliding mode control to establish the desired direction of motion of the drone to the desired turn rate. In this sliding mode control it can only switch between two values of the turn rate. This leads to a poor energy performance.

The oscillatory strategy has the best performance for the straight and circular ground agent motion patterns. This can be explained by the fact that the drone has to perform smaller bank angles while tracking the ground agent with higher speeds. However, for Levy walk motion

patterns, no optimal parameters were found during the optimization hence in Figure 2.6 c,f, only results for LGVF and the Bearing only strategy are presented.

To better explain the results of the optimization algorithm for the oscillatory strategy, we run additional simulation experiments to visualize for which pairs of control parameters: distance gain  $k_e$  and speed gain  $k_v$ , the oscillatory strategy provides stable tracking behavior. Under experiments with stable tracking we consider the experiments in which distance to the ground agent is less than  $10R_{min}$ , where  $R_{min}$  is the minimum turning radius of the drone. Similarly, under experiments with non-stable tracking we consider the experiments in which distance to the ground agent is more than  $10R_{min}$ . We run the simulations for all ground agent motion patterns for each combination of control parameters. In Figure 2.7 we show the combinations of control parameters for which oscillatory control showed stable behavior (indicated by the black dot).

It can be seen in the simulation results that the oscillatory strategy provides stable behavior for most combinations of speed and distance gains only for straight line following and circular patterns with constant speed (Fig. 2.7 a,b). In contrast, for circular and random ground agent patterns with variable speed (Figure 2.7 d,e,f), the stability region is considerably reduced. Although this strategy showed the best performance for straight and circular patterns, we can observe that stability of the method is highly dependant on the ground agent trajectory. We choose not to conduct further experiments for the oscillatory strategy due to the lack of stability in simulation. It is to be expected that if the strategy is not robust in simulation, its robustness will not improve in real experiments. For this reason, for outdoor experiments we consider only the bearing-only and LGVF strategies.

### 2.4.2 Field results

We examine three comparison cases in the field in which the drone is following a circular, straight and Levy motion pattern of the ground agent. The ground agent is moving with constant speed equal to 30% of the drone's forward speed. The results of the optimization experiments for each comparison case are given in Figure 2.8. For each comparison case in the field we choose two different optimal control parameter values and we run five runs for each controller parameter value. The results of the flight experiments for the LGVF strategy are presented in Figure 2.10 together with the drone trajectories presented in Figure 2.9. The results from flight experiments support the results obtained in simulation, with a relative deviation of less than 20%. This deviation is thought to occur due to the aerodynamic effects and disturbances that were not modeled in the simplified simulation model.

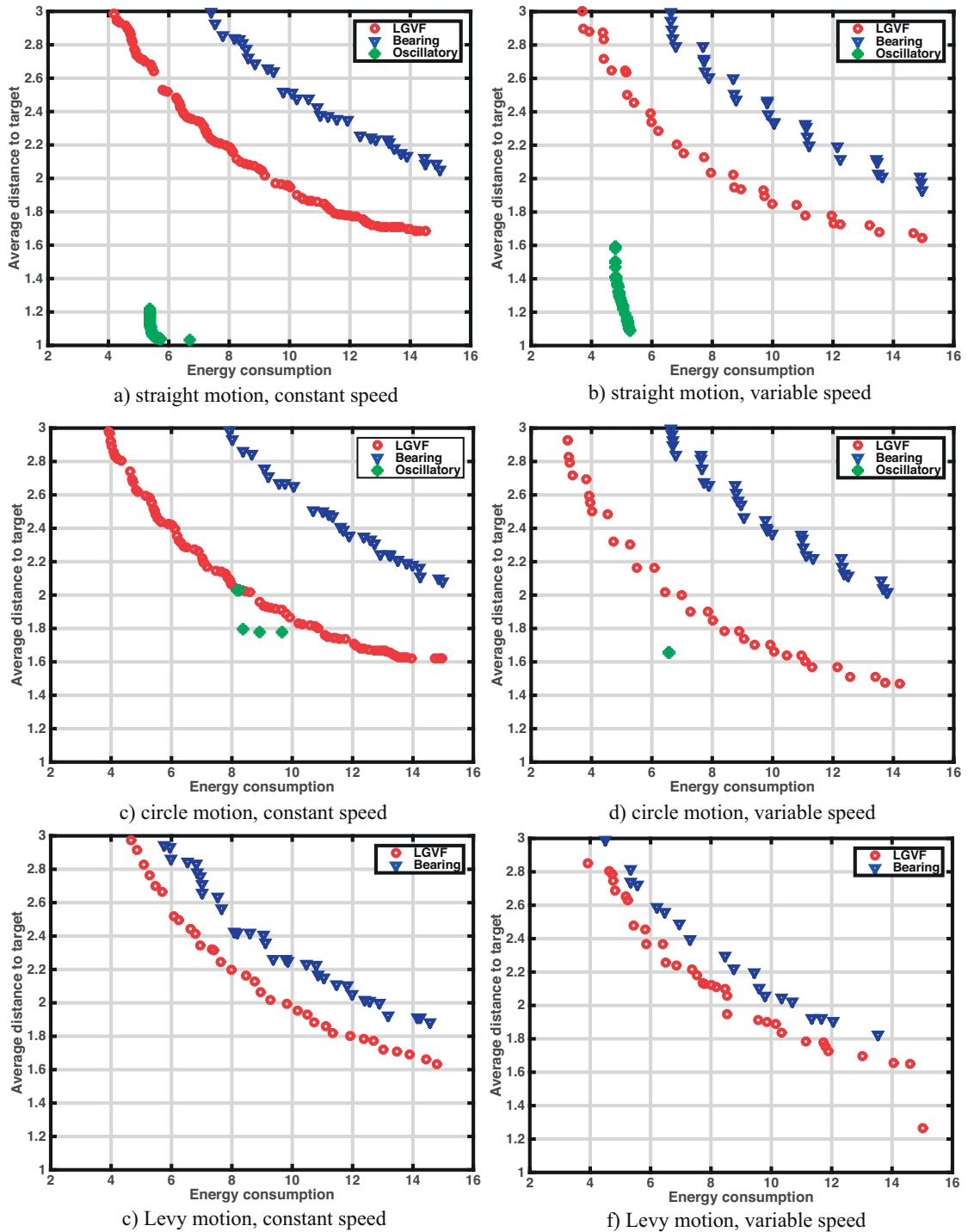


Figure 2.6: Results of the simulation experiments for the LGVF, Bearing-only and Oscillatory strategy for diverse motion patterns of ground agents. The distance measure is the average distance between the drone and the ground agent, normalized to the minimum turn radius of the drone. The energy consumption measure is the relative energy consumption compared to that of the level flight, given in %. The LGVF strategy outperforms the Bearing-only strategy for all motion patterns. Oscillatory strategy performs the best for straight and circle motion patterns. However, for Levy walk motion patterns, no optimal parameters were found during optimization.

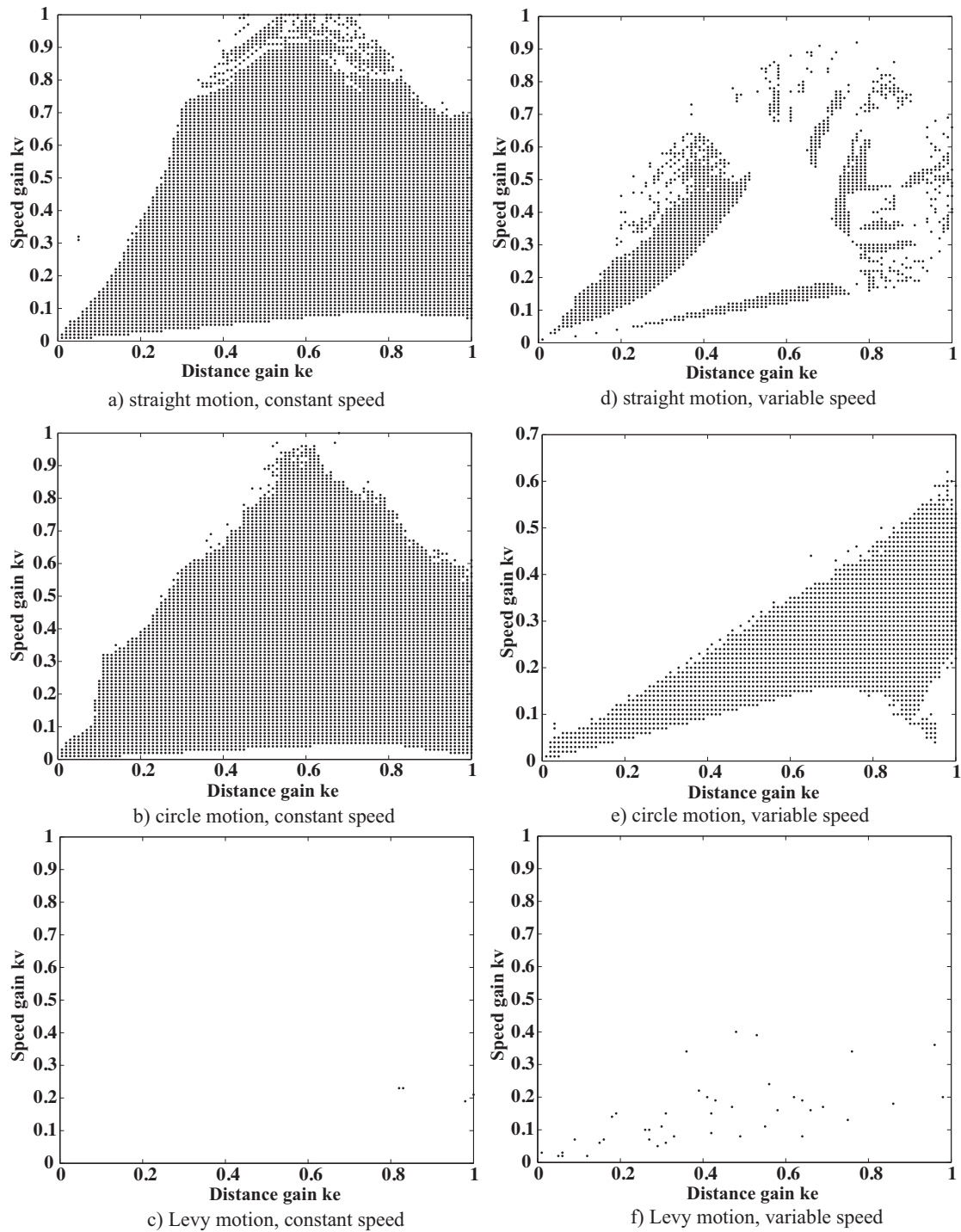


Figure 2.7: Simulation results for the stability analysis of the oscillatory control strategy. The set of parameters for which oscillatory strategy leads to stable behavior is plotted with black dots. The strategy shows low robustness to variable ground agent motion patterns and speeds.

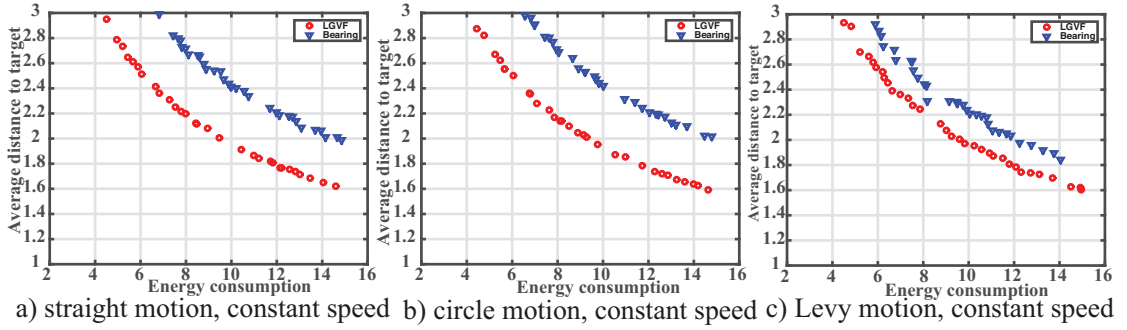


Figure 2.8: Results of the optimization experiments for the field experiment comparison cases. We choose two sets of parameters obtained in the optimization for our field experiments. We choose parameters that give higher relevance to the energy performance objective and parameters that give higher relevance to the distance objective (upper and lower ends of Pareto fronts).

For the bearing-only approach we run experiments only for the first comparison case (straight motion pattern of the ground agent with its speed set to  $v_a = 0.3v_d$ ). The results in the field show that bearing only strategy performs worse in field experiments than LGVF strategy, which supports our conclusion from simulation experiments. The results are shown in figure 2.11. In this experiment a chattering phenomenon is observed. The suspected cause of this phenomenon is the sliding mode control used in this strategy. The output of the bearing only control is determined by the angle between the course vector of the drone and the vector that points from the ground agent position to the drone position (distance vector). If the angle between these two vectors is  $(0, \pi]$  the output of the controller is the maximum turn rate,  $\omega_{max}$ . If the angle between these two vectors is  $(\pi, 2\pi]$  the output of the controller is  $-\omega_{max}$  (Eq. 2.4, Eq. 2.5). Under such sliding mode control, some chattering occurs due to the small variations around zero angle between vectors, which causes large variations in the controller output.

In general, the phenomena of chattering in sliding mode control is a well known problem that can be solved by adding a linear function in the sliding mode controller function [86]. Authors of the bearing-only strategy considered and solved the chattering problem in the case when the angle between the distance vector and course of the drone is around zero. When the angle between the distance vector and the course of the drone is close to  $\pi$  or  $-\pi$ , a small disturbance in outdoor experiments can cause a change in the relative angle from  $-\pi$  to  $\pi$  or vice versa. This sudden change, caused by a discontinuity in the value of the relative angle between the distance vector and the course of the drone causes sudden changes in the commanded turn rate from  $-\omega_{max}$  to  $\omega_{max}$  or vice versa. This causes an oscillation of the output of the controller. These oscillations do not appear in simulations since we consider an ideal kinematic model of the drone and do not take into the account many aerodynamic effects

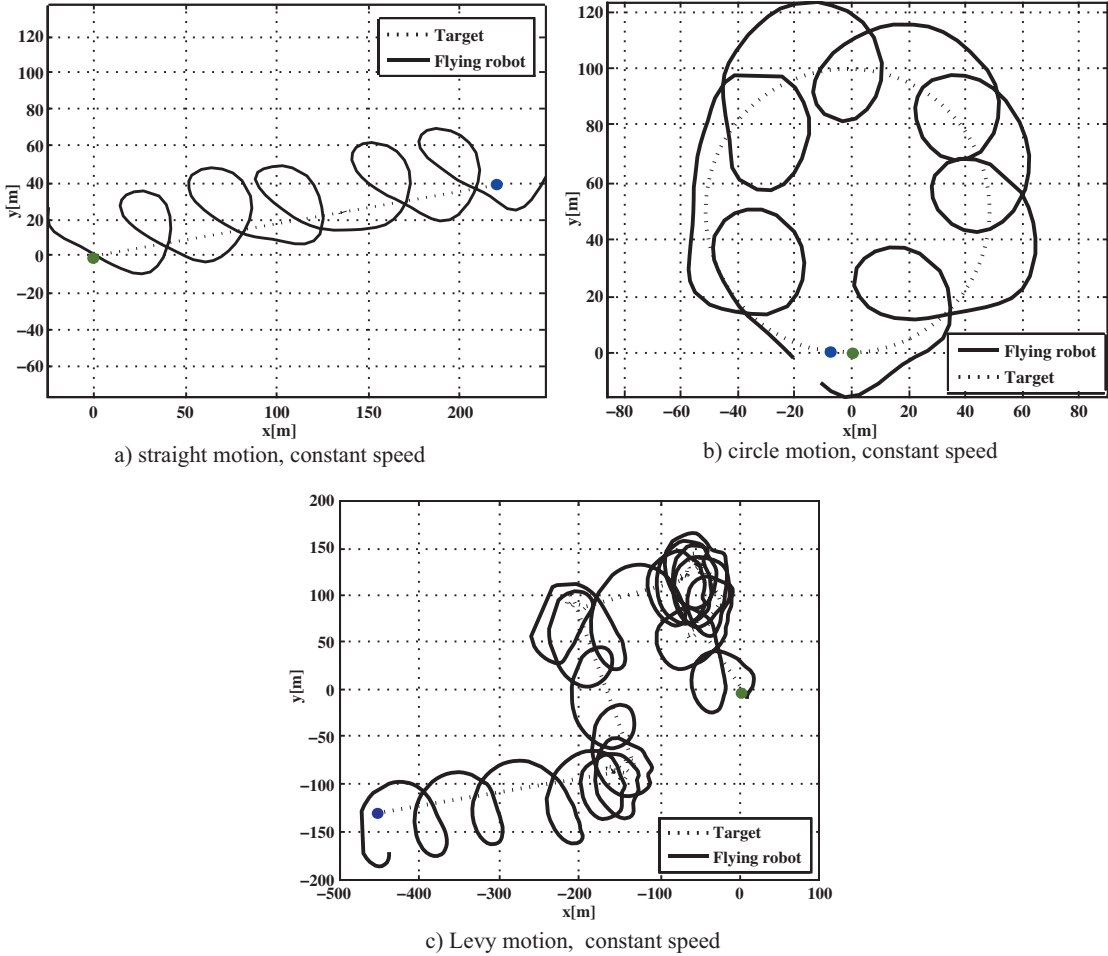


Figure 2.9: Drone trajectories obtained in outdoor experiments for three ground agent motion patterns for LGVF strategy. The ground agent's speed is set to  $0.3v_d$ . The starting point of the ground agent trajectory is presented with a blue dot and the end point is presented with a green dot.



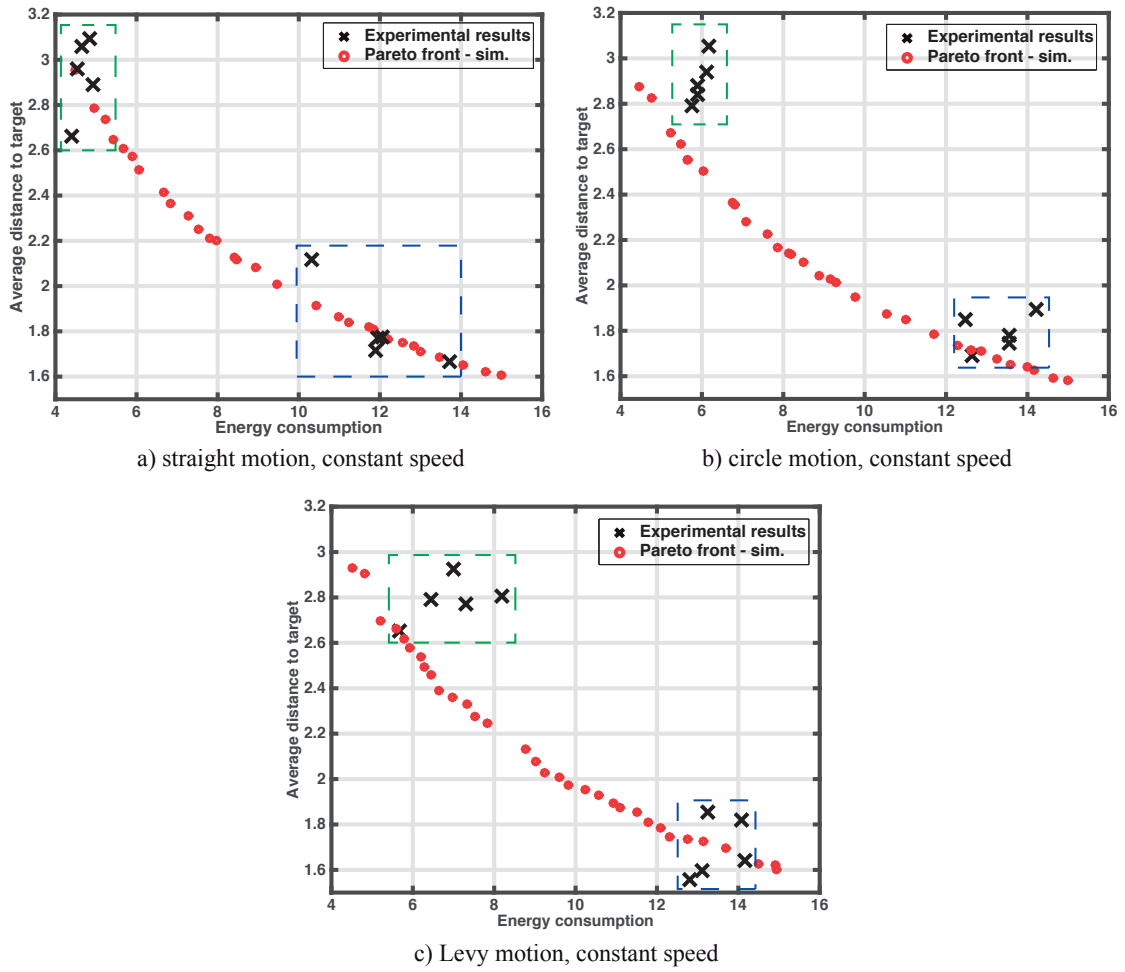


Figure 2.10: Results of outdoor flight tests for two sets of controller parameters for LGVF strategy. Experimental results obtained with the first controller parameter set (higher relevance to the energy performance objective) are within green rectangles and the results obtained with the second control parameter set (higher relevance to the distance objective) are within blue rectangles. The experimental results support the results from simulation using the simplified platform model.

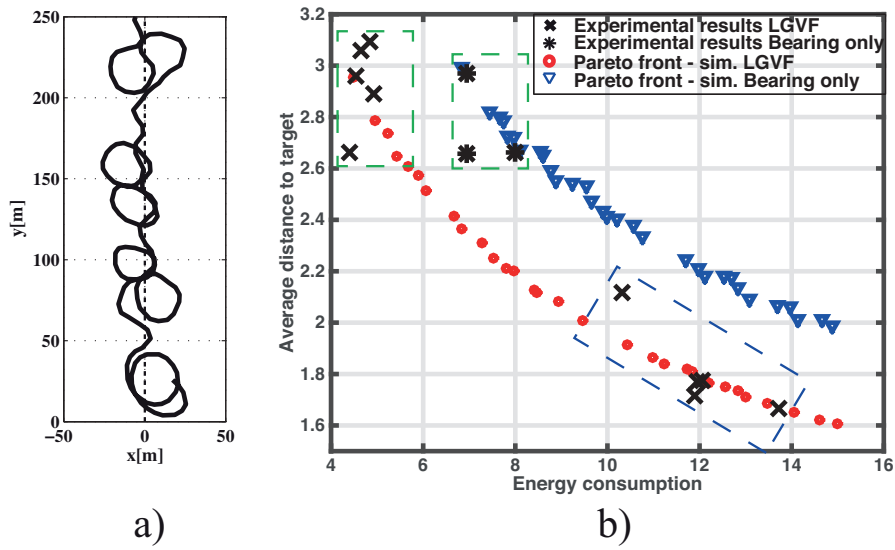


Figure 2.11: Results of outdoor flight tests for bearing-only strategy compared to results of the field experiments for LGVF strategy. The trajectory of the drone is shown in a). In b) we show the results of three experiments in the field in which chattering did not disrupt significantly the drone's flight. The experimental results support the results from simulation using the simplified platform model.

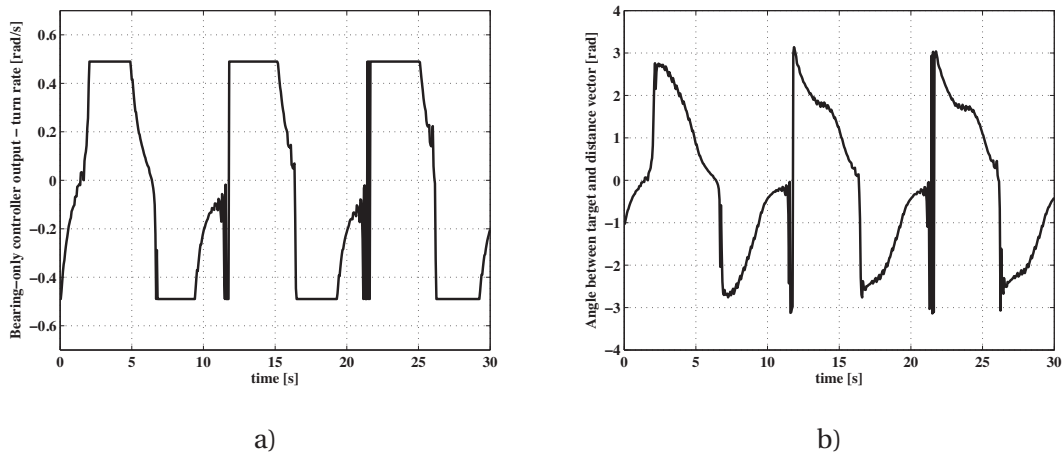


Figure 2.12: Experimental results for the bearing-only strategy, a) controller output (turn rate) b) angle between drone course and distance vector. In the outdoor experiments, the drone is following a ground agent performing straight line motion,  $v_a = 0.3v_d$ .

and disturbances. Regardless of the control architecture of the autopilot, these oscillations cause potentially unstable behavior. Unstable behavior was detected in tests in which the drone was required to perform more aggressive maneuvers (higher bank angles). Therefore, further tests on more aggressive trajectories were not conducted.

### 2.4.3 Robustness to wind - simulation experiments

Simulation experiments are conducted to examine the effect of wind in the LGVF and bearing-only strategies, for four directions of wind and wind speeds ranging from 0.1 – 0.6 times of the forward speed of the drone. The maximum wind speed is chosen based on the maximum allowable wind for fixed wing platforms [13]. In these experiments we are interested in the effect of wind so we set the ground agent motion to straight line motion with a low speed set to  $0.3v_d$ .

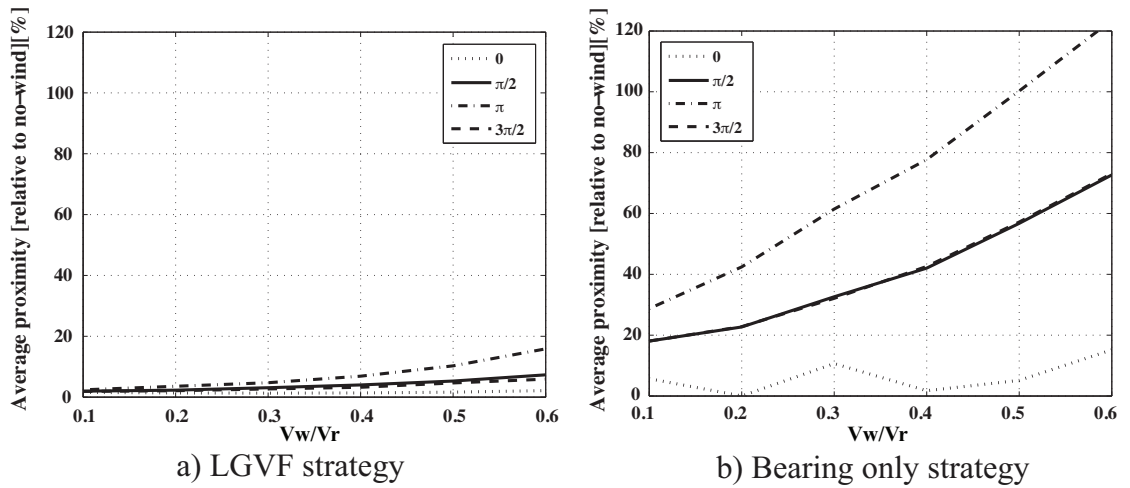


Figure 2.13: Simulation results of experiments that examine robustness to constant wind. We run experiments for four different orientations between target motion and wind direction. The LGVF strategy shows better robustness to the constant wind than the bearing-only strategy

Results for the Bearing-only control and the LGVF control under various wind conditions are presented in Figure 2.13. These results show that the Bearing-only control has stable behavior in all wind conditions but the tracking error increases with higher wind speeds. In addition, for higher wind speeds the deviation from the straight ground agent trajectory is more than 30% compared to the no wind case. In contrast, the LGVF strategy has better performance under wind, with a deviation of less than 20% . We postulate that this is due to the control algorithm’s ability to compensate for the wind disturbances.

2.4.4 Robustness to wind - field experiments

The outdoor experiments in changing wind conditions are conducted for only the LGVF strategy, which showed stable performance both in simulation and in field experiments regardless of the ground agent motion pattern. The aim was to investigate the effectiveness of the control strategy in handling wind in real experiments. We ran experiments for different wind conditions, for low and high wind speed. Although wind conditions can be simulated and robustness can be tested in simulation, in field experiments our aim is to determine whether non-modeled aerodynamic effects have an influence on the performance in windy conditions. We run a set of experiments in which we change the orientation of the target's straight motion pattern relative to the wind direction. For each target orientation we run 5 experiments.

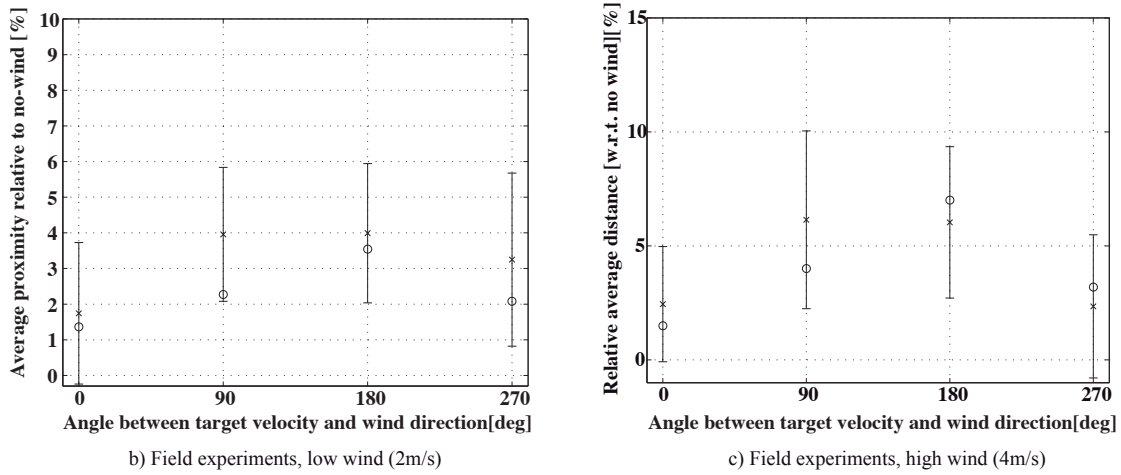


Figure 2.14: Results of outdoor experiments in varying wind conditions. Experiments in low constant wind conditions (2 m/s) deviate less than 10% from the results obtained in simulation for the same wind speed. Experiments in high wind conditions also show good agreement with simulation results.

Figure 2.14 presents the results of mean relative distance to the target over the whole flight time (normalized to the simulated no-wind case) for both extreme wind tests. Results suggest that the LGVF strategy is robust to changing wind conditions. In the low-wind case (2m/s), results match the values obtained in simulation. We use the mean distance and the standard deviation to compare experimental results with simulation results. We can observe that the deviation between simulation results and experimental results is within a few percent and overall mean relative distance in experiments is lower than 5% which also endorses the fact that the LGVF strategy is robust to constant wind. In the high wind case, the field experiments for the LGVF strategy show good agreement with simulation experiments. The mean value

of the relative distance to the target in the field experiments deviates from the simulation experiments within a 5% range.

## 2.5 Conclusion

By using only a distance measure to assess the performance of the target tracking strategies, we cannot reach conclusions about the energy performance of the strategies which plays an important role in field applications. Introducing a new measure, energy expenditure, allows us to identify not only which strategy will allow the drone to track the ground agent with minimal distance but also which strategy will do so by using the least energy. We use a systematic approach to perform a fair comparison of the strategies. We set parameters of the strategies to the optimal values obtained in optimization, which had minimal energy and distance as objectives, and then we compare the Pareto fronts obtained in the optimization to assess which strategy performs the best.

Out of the three strategies compared in this study, results of the simulation and field experiments showed that both good distance and energy performance can be achieved with the Lyapunov Guidance Vector Field (LGVF) strategy. In Table 2.15 we show the results of our comparison method for all ground agent motion patterns for simulation and field experiments. We give grades for the performance of each strategy. If a strategy performs the best according to our measure, it gets the maximum grade 3. If a strategy was not tested due to the unstable behavior, the grade is not assigned (N/A). The LGVF strategy has the highest cumulative grade, since its performance does not significantly change with variation in the ground agent motion patterns and in field experiments. The bearing only control showed good robustness to changing ground agent motion patterns but it showed problems with chatter in the field experiments. We also noticed that the oscillatory control strategy, although it outperformed other strategies for circular and straight line motion, failed in following the ground agent performing a Levy walk motion pattern.

To conclude our comparison and examine the performance of the LGVF strategy in wind conditions, we ran an additional set of experiments in windy conditions. We show results of the comparison based on the distance measured in low and high wind conditions for the LGVF strategy. In simulation and in field experiments we show that the LGVF compensates wind disturbance well, the relative difference in distance measure compared to the no-wind experiment is less than 20%. It also compensates wind disturbances better than the bearing only strategy.

## Chapter 2. Aerial target tracking

---










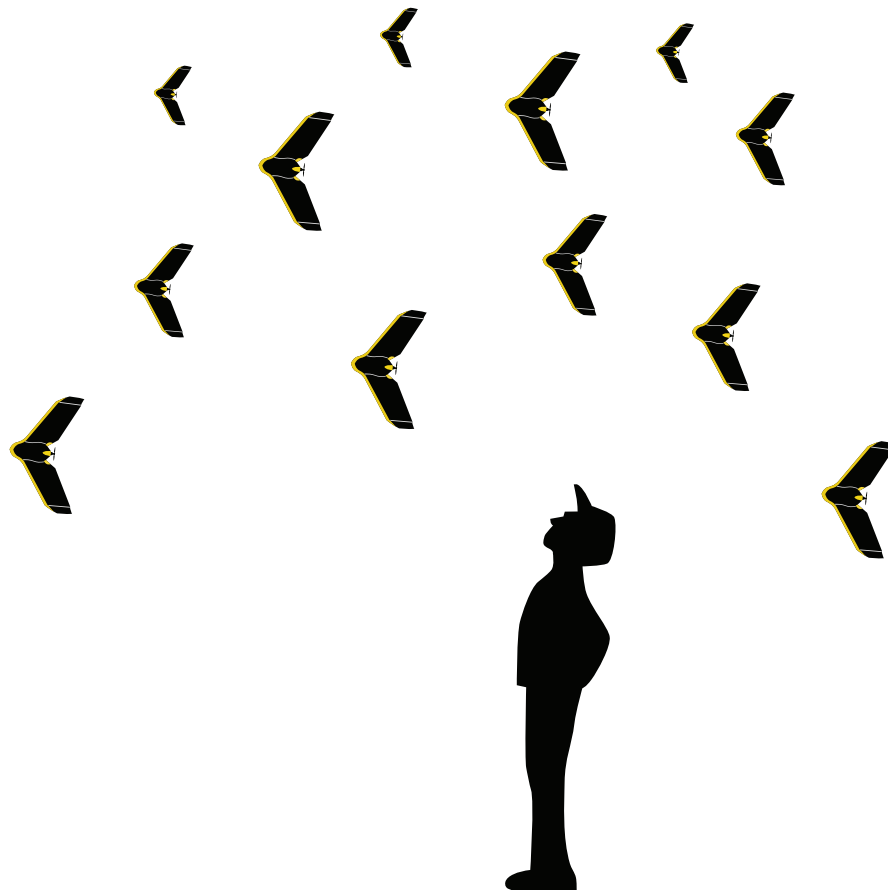
	LGVF			Bearing only			Oscillatory		
									
Simulation	2	2	3	1	1	2	3	3	N/A
Field exp.	3	3	3	2	N/A	N/A	N/A	N/A	N/A

Figure 2.15: Comparison table showing the performance of the strategies for diverse ground agent motion patterns in simulation and field experiments. Grades are given according to the performance metrics presented in the paper. The best performance is given grade 3, and the worst grade 1. If a strategy is graded 0, it implies that the strategy showed unstable behavior.

By introducing a larger set of ground agent motion patterns into the experimental method we were able to identify the strategies that were not robust to the different ground agent motion patterns. By running field experiments in addition to simulation experiments we were able to observe the chattering phenomena in the bearing only control which could not be detected in simulation experiments.

### 3 Formation control with fixed wing drones



This chapter is based on following publications:

Hauert, S., Leven, S., Varga, M., Ruini, F., Cangelosi, A., Zufferey, J. C., & Floreano, D. (2011, September). Reynolds flocking in reality with fixed-wing robots: communication range vs. maximum turning rate. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (pp. 5015-5020). IEEE.

Varga, M., Basiri, M., Heitz, G., & Floreano, D. Distributed Formation Control of Fixed Wing Micro Aerial Vehicles for Area Coverage. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (pp. ). IEEE.



### 3.1 Introduction

There is an increasing interest in deploying teams of drones to achieve distributed sensing of the environment and to establish an aerial wireless ad-hoc network to connect users on the ground [13, 43, 39]. In such applications, drones are required to uniformly position themselves to cover an area with their on-board sensors and to optimize communication performance with other drones in the network. To achieve this, drones need to control their distance to other drones. To achieve scalable performance and to be prone to the drone failures, drones in the team should perform the mission in a decentralized and distributed way.

Drones that can hover, such as quadrotors and helicopters, are used either to distributively adapt their mutual distance to optimize the coverage with on-board cameras [73] or adapt their mutual distance based on the quality of the communication signal [33].

Compared to quadrotors and helicopters, fixed wing drones have higher coverage speed, better robustness to wind disturbances and longer battery life [53]. Due to these properties, fixed wing drones are more suitable for some outdoor applications where large areas should be covered or communication should be relayed over large distances. Fixed-wing drones have constrained kinematic properties, they cannot suddenly change direction due to the limitations in the bank angle and they always need to maintain a forward speed in order to remain airborne. Therefore, existing algorithms for formation control of hovering platforms cannot be applied for fixed wing drones.

Existing literature on formation control of fixed wing drones is mostly focused on keeping the formation of fixed wing drones connected and moving in the same direction. In [82], flocking rules are presented that govern fixed wing like vehicles to move in a common direction without adapting inter-vehicle distance. Similar velocity alignment in 3D is presented in [58]. In [6] three fixed wing drones are flying in a triangular formation while following a path that is known to each agent in the formation in advance.

In this chapter, a method for controlling the formation of a swarm of fixed wing drones from arbitrary initial positions to a uniform formation with equal inter-robot distances is proposed. Drones need to uniformly cover an area and keep communication link quality on a desired level. Our aim is to deploy a swarm of fixed wing drones to cover an area with their on-board cameras, collect aerial images and at the same time serve as aerial wireless network for ground users.

In our formation algorithm, drones rely on the local information communicated from their

single hop (or direct) neighbors. To guarantee that the forward speed of the drones are always greater than the stall speed, a virtual agent is defined for every drone and drones are controlled to continuously follow and circle around their virtual agents. Drones emulate the motion of the virtual agent and follow it by performing spiraling motion using Lyapunov Guidance Vector Field algorithm [31]. Formation algorithm is then implemented between virtual agents. To map the formation of the virtual agents to the formation of drones, drones also synchronize their phase angles. We test the performance of our method in simulation and in field experiments.

### 3.2 Formation algorithm

This section presents the formation algorithm that allows fixed wing drones to uniformly cover an area and keep communication link quality on a desired level. To achieve this, each drone should regulate the distance to its neighbors.

#### 3.2.1 Proposed method

Our method consists of three control modules: virtual agent following module, distance control module and synchronization module. The virtual agent following module, controls the drone to follow its virtual agent by performing an orbital flight path along virtual agent's trajectory. The orbital motion is designed by Lyapunov Guidance Vector Field (LGVF) approach [31] which showed the best performance for target tracking in Chapter 2. The trajectory of the flying robot while following a virtual agent, obtained in a field experiment is shown in the Figure 3.1.

In distance control module, we implement a modified version of the flocking rules proposed by Olfati-Saber [60]. Each drone calculates flocking rules for their virtual agents, with information received from its interaction neighborhood defined as:

$$I_i^{drone} = \{va_j : \|\mathbf{p}_j^{va} - \mathbf{p}_i^{va}\| < k_{int}d_{ij}\} \quad (3.1)$$

where  $d_{ij}$  is the desired distance between drone  $i$  and drone  $j$ . The desired distance is limited by the communication radius  $r_c$ . Parameter  $k_{int}$  should be set to a value greater and close to 1 (in our simulation we set  $k_{int} = 1.2$ ). If every point mass dynamics agent has the same interaction range set by (3.1), it has been empirically shown in [60] that a flock of point mass dynamics agents will converge to a uniform formation.

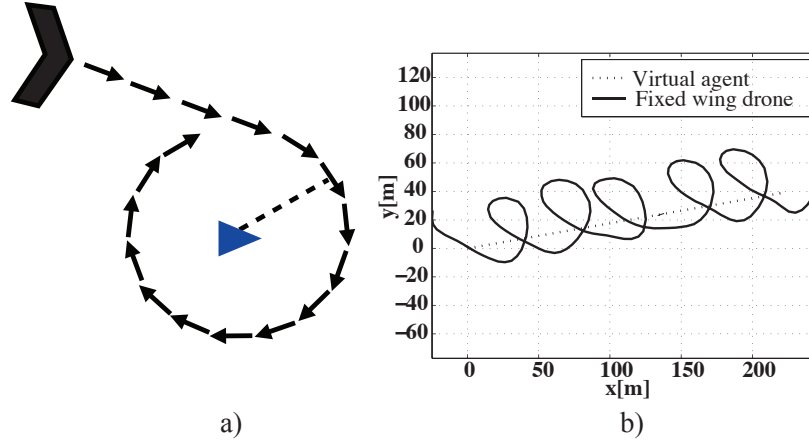


Figure 3.1: a) Schematic Illustration of drone following virtual agent. b) Result from real world experiment involving one fixed wing drone and a virtual agent, illustrating the motion of a fixed wing drone flying at 10 ms<sup>-1</sup> and following its virtual agent moving at 1 ms<sup>-1</sup>.

Output of the distance control module is the virtual agent's control input, a force vector,  $\mathbf{u}_i$  and is defined as:

$$\mathbf{u}_i = \mathbf{F}_i^{va} = \mathbf{F}_{attr/rep_i}^{va} + \mathbf{F}_{align_i}^{va} + \mathbf{F}_{nav_i}^{va} \quad (3.2)$$

where  $\mathbf{F}_{attr/rep_i}^{va}$  is the attraction/repulsion force,  $\mathbf{F}_{align_i}^{va}$  is the alignment force and  $\mathbf{F}_{nav_i}^{va}$  is the navigation force, Figure 3.2.

Virtual agent is attracted towards the neighbors whose distance is greater than the desired distance but is repelled from agents whose distance is less than the desired distance. This interaction is defined by the attraction/repulsion force:

$$\mathbf{F}_{attr/rep_i}^{va} = \sum_{j \in I_i^{drone}} \phi(\|\mathbf{p}_j^{va} - \mathbf{p}_i^{va}\|) \delta\left(\frac{\|\mathbf{p}_j^{va} - \mathbf{p}_i^{va}\|}{r_c}\right) \mathbf{n}_{ij} \quad (3.3)$$

where  $\mathbf{n}_{ij} = \frac{(\mathbf{p}_j - \mathbf{p}_i)}{\|\mathbf{p}_j - \mathbf{p}_i\|}$  is a unit vector along the direction of virtual agent  $j$ ,  $\phi(q)$  is a sigmoid

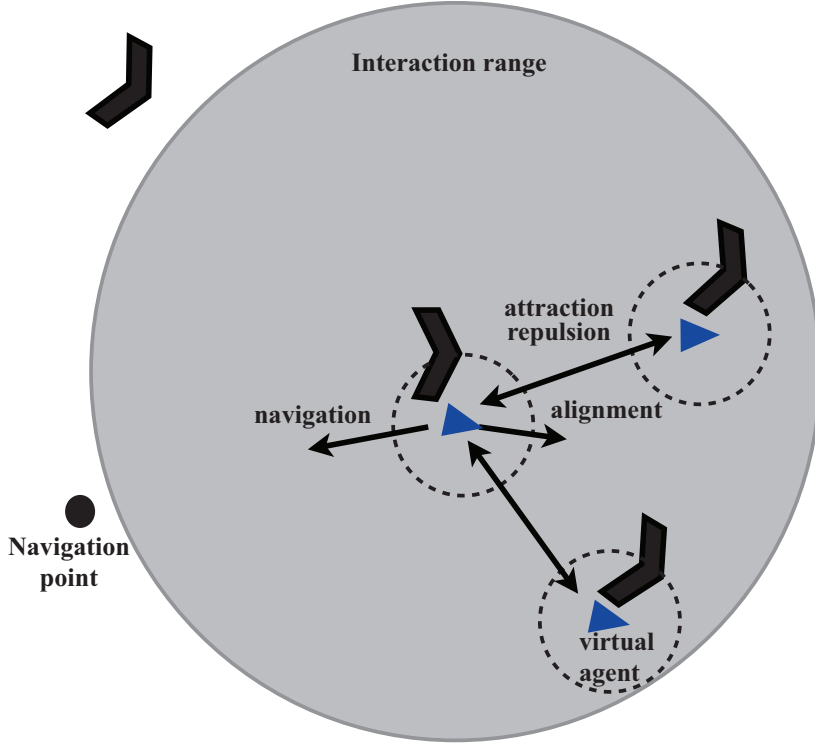


Figure 3.2: Schematic illustration of the distance control module. Alignment, attraction/repulsion and navigation force are acting on the virtual agent.

function given by:

$$\phi(q) = \frac{1}{2} \left( (a+b) \frac{q+c}{\sqrt{1+(q+c)^2}} + (a+b) \right) \quad (3.4)$$

where  $a$ ,  $b$  and  $c$  are tunable parameters that determine the value of attraction/repulsion force. Parameters  $a$  and  $b$  are set to the maximum and minimum attraction/repulsion force value and parameter  $c$  is set to the desired distance. Bump function  $\delta(q)$  is used to smoothly set the attraction/repulsion force to zero outside the communication radius, is defined as:

$$\delta(q) = \begin{cases} 1 & q \in [0, h) \\ \frac{1}{2} [1 + \cos(\pi \frac{q-h}{1-q})] & q \in [h, 1) \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

where  $h$ ,  $h < r_c$  is the parameter that determines when the bump function will start to set

the value to zero. Virtual agents align their headings by applying alignment force:

$$\mathbf{F}_{align_i}^{va} = \sum_{j \in I_i^{drone}} (\mathbf{v}_j^{va} - \mathbf{v}_i^{va}) \quad (3.6)$$

In addition, a navigational force is applied on the virtual agents to keep them from splitting in sub-flocks. Each drone in the formation knows the position and velocity of the navigational point,  $\mathbf{p}^{np}$  and  $\mathbf{v}^{np}$ . The force is defined as:

$$\mathbf{F}_{nav_i}^{va} = -f_\sigma(\mathbf{p}_i^{va} - \mathbf{p}^{np}) - f_\sigma(\mathbf{v}_i^{va} - \mathbf{v}^{np}) \quad (3.7)$$

where  $f_\sigma$  is a sigmoid defined as  $\phi(q)$  (3.4) with parameters  $a$  and  $b$  set to maximum and minimum navigational force value and parameter  $c = 0$ .

Previous two modules provided a method for generating a uniform formation between the virtual agents of all drones. As illustrated in Figure 3.3, in order to obtain equal distance between the drones themselves, drones are also required to synchronize their orientation with their neighbors. The synchronization module controls the drone's phase angle by controlling the forward speed of the drone within the allowed interval. Based on the position of the neighboring drones and their respective virtual agents, each drone can determine the phase of its neighbors and compute the change on its forward speed to synchronize the phase angle  $\theta$  by:

$$\Delta v_i^{drone} = \frac{1}{N(C_i^{drone})} \sum_{j \in C_i^{drone}} f_\sigma(\theta_j - \theta_i) \quad (3.8)$$

where  $N(C_i^{drone})$  is a number of communication neighbours of the  $drone_i$ ,  $\theta_i$  is a phase angle of the  $drone_i$ . If the  $drone_i$  phase angle is less than  $drone_j$  phase angle,  $drone_i$  will increase its forward speed to align its phase angle with its neighbor. When difference in phase angles in the communication neighborhood of the  $drone_i$  is equal to zero, then the  $\Delta v_i^{drone}$  will be also equal to zero and drones will all reach consensus in phase angle. The drone's altitude is controlled by the PID controller on the autopilot control level. The presented modules allow a team of fixed wing drones to achieve a uniform formation with a desired inter robot distances.

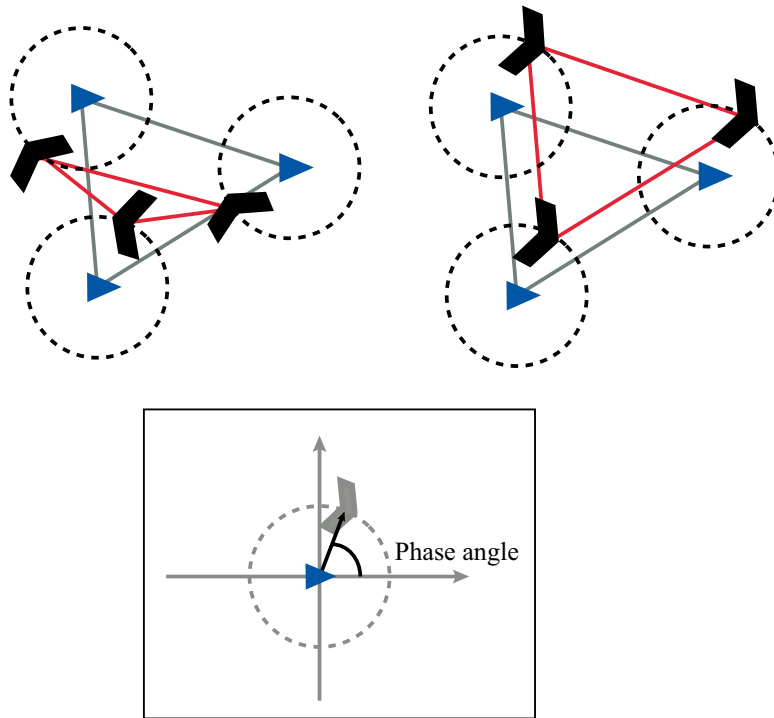


Figure 3.3: Drones need to synchronize their phase angle with their neighbors. They synchronize their phase angles by altering their forward speed in a limited interval. In this way a formation of drones is converging to the formation of their respective virtual agents.

### 3.2.2 Discussion on formation algorithm convergence

The convergence of the drones formation to an equilateral triangular lattice depends on the stability properties of all three formation algorithm modules. In this section, the stability of each module will be commented and analyzed. Flocking module drives the motion of the virtual agent and provides input to the target following module. Since fixed wing drone can only follow targets which move up to its maximum forward speed, we analyze whether this constraint has impact on the algorithm convergence.

Firstly, collective properties of the virtual agents formation are analyzed, such as cohesiveness of the formation and whether formation reaches an equilibrium. It is assumed that the navigational point speed is equal to zero. For such analysis, a collective formation potential and Laplacian matrix that encodes properties of the interaction graph should be defined. In global coordinates, virtual agents' collective dynamics applying interaction force (3.2) is given

by:

$$\begin{aligned}\dot{\mathbf{p}}^{va} &= \mathbf{p}^{va} \\ \dot{\mathbf{v}}^{va} &= -\Delta V(\mathbf{p}^{va}) - L(\mathbf{p}^{va})\mathbf{v}^{va} + f_{nav}(\mathbf{p}^{va}, \mathbf{v}^{va}, \mathbf{p}^{nav}, \mathbf{v}^{nav})\end{aligned}\quad (3.9)$$

where  $V(\mathbf{p}^{va})$  is a collective potential of the lattice,  $L(\mathbf{v}^{va})$  is a Laplacian matrix of the interaction graph, and  $f_{nav}(\mathbf{p}^{va}, \mathbf{v}^{va}, \mathbf{p}^{nav}, \mathbf{v}^{nav})$  is a navigation force. Formal analysis of convergence is transferred to the formation motion frame. If  $\mathbf{p}_c$  and  $\mathbf{v}_c$  are center of mass and average velocity of all virtual agents in the formation, then the position and velocity in the moving frame are given by:

$$\begin{aligned}\tilde{\mathbf{p}}_i^{va} &= \mathbf{p}_i^{va} - \mathbf{p}_c \\ \tilde{\mathbf{v}}_i^{va} &= \mathbf{v}_i^{va} - \mathbf{v}_c\end{aligned}\quad (3.10)$$

Then, the structural dynamics of the virtual agents formation is given by

$$\begin{aligned}\dot{\tilde{\mathbf{v}}}^{va} &= \tilde{\mathbf{p}}^{va} \\ \dot{\tilde{\mathbf{p}}}^{va} &= -\Delta U_\lambda(\tilde{\mathbf{p}}^{va}) - D(\tilde{\mathbf{p}}^{va})\tilde{\mathbf{v}}^{va}\end{aligned}\quad (3.11)$$

where  $U_\lambda$  is an aggregate potential function and  $D(x)$  is a damping matrix. The aggregate potential function incorporates navigation force and attraction/repulsion force and the damping matrix incorporates navigation force and alignment force. Having the structural dynamics defined, following theorem proves stability of the virtual agent's formation algorithm:

**Theorem 1** (Olfati-Saber[60]) *If we consider virtual agents applying interaction force proposed by 3.2 and structural dynamics given by (3.11). If initial structural energy is finite and interaction range satisfies  $r/d = 1 + \epsilon$  ( $\epsilon \ll 1$ ), where  $r$  is interaction radius and  $d$  is desired distance. If local minimum of  $U_\lambda$  induces connected graph, then almost every solution of the structural dynamics asymptotically converges to an equilibrium which is an equilateral triangular lattice.*

A formal proof of this theorem is based on the formation energy analysis. It is shown that the derivation of structural dynamics Hamiltonian is decreasing, which means that the formation will reach an equilibrium at finite time. At the equilibrium every virtual agent has zero speed. In the process of reaching equilibrium, virtual agents could maybe have speed which is higher than maximum forward speed and fixed wing drones will not be able to follow them. If we assume that maximum distance between virtual agents is always lower than communication radius of drones, we can assume that in these cases there will be no fragmentation of the formation due to the communication loss. The equilibrium point of the virtual agent  $va_i$  is

### Chapter 3. Formation control with fixed wing drones

---

given by  $[x_i^{va}, y_i^{va}]$ .

The drone motion is controlled by Lyapunov guidance vector field defined by 2.3 in Chapter 2. The Lyapunov function is used to define a stable vector field given by:

$$V(x_i^{drone}, y_i^{drone}) = (r^2 - r_d^2)^2 \quad (3.12)$$

where  $r = \sqrt{(x_i^{drone} - x_i^{va})^2 + (y_i^{drone} - y_i^{va})^2}$ . The time derivative of Lyapunov function is given by:

$$\frac{dV}{dt} = \frac{-4\alpha v_0 r (r^2 - r_d^2)^2}{r^2 + r_d^2} \quad (3.13)$$

where  $v_0$  is forward speed of fixed wing drone and  $\alpha > 0$  is a control parameter. By substituting  $r = r_d$  it can be observed that the time derivative of the Lyapunov function is equal to zero. This is equivalent to the case when drone performs circle motion around position of virtual agent. Thus, vector field is globally stable to the circle around virtual point. This means that each drone from any initial condition can converge to the orbital motion around the equilibrium point of its virtual agent in finite time. Under the assumption that distance between any two drones will be always smaller than  $r_c$ , virtual agents will reach an equilibrium and their final speed will be equal to zero. If a virtual agent's equilibrium position is not in the infinity, then each drone will converge to a stable circling around its virtual agent in finite time.

Since virtual agents converge to the lattice and drones converge to a stable circle around their equilibrium points, the next step is to show whether drones will be able to synchronize their phase. Drones synchronize their phase by altering their forward speed in the tight interval. In the steady state, we showed that virtual agents reach their equilibrium positions, their speed is zero and drones perform circular loitering motion around virtual agents' position. By slightly speeding up or slowing down their motion, drones can adjust their phase angle to the phase angle of their neighbors by applying following control law:

$$\Delta v_i = \frac{1}{N(C_i^{drone})} \sum_{j \in C_i^{drone}} f_\sigma(\theta_j - \theta_i) \quad (3.14)$$

where  $f_\sigma$  is a sigmoid function given by (3.4) with parameters  $a, b$  set to limit the  $\Delta v_i$ . Forward speed of the robot can be changed only in the interval  $[v_{stall}, v_{max}]$ . If we define nominal



forward speed  $v_0$ , then:

$$\begin{aligned} v_0 + \Delta v_{max} &< v_{max} \\ v_0 - \Delta v_{max} &> v_{stall} \end{aligned} \quad (3.15)$$

Therefore, parameters  $a, b$  are set to:

$$\Delta v_{max} = \frac{v_{max} - v_{stall}}{2} \quad (3.16)$$

Since drone performs circular motion, the forward speed change  $\Delta v_i$  is related to the phase angle change  $\dot{\theta}_i$  as

$$\Delta v_i^{drone} = k_\theta \dot{\theta}_i \quad (3.17)$$

where  $k_\theta$  is control gain. Then equation (3.14) can be written as:

$$\dot{\theta}_i = \frac{1}{k_\theta N(C_i^{drone})} \sum_{j \in C_i^{drone}} f_\sigma(\theta_j - \theta_i) \quad (3.18)$$

The sigmoid function (3.18) is linearized in the interval  $||(\theta_i - \theta_j)|| < \pi$  and is given by:

$$\dot{\theta}_i = \frac{\Delta v_{max}}{k_\theta N(C_i^{drone}) \pi} \sum_{j \in C_i^{drone}} (\theta_j - \theta_i) \quad (3.19)$$

which represents a consensus algorithm. It has been shown in [61] that such algorithm will converge to a consensus in which  $\theta_i - \theta_j = 0$  for every drone pair  $(i, j)$ .

Drones exchange information over wireless communication network. Virtual agents' motion depends on the received information from neighboring drones. This information is accessible in discrete time steps, since communication channel bandwidth is limited and only finite number of messages can be sent and received by neighboring agents. Therefore, it is necessary to analyze the algorithm convergence in a discrete domain. In a discrete domain, virtual agents motion is expressed as:

$$\begin{aligned} \mathbf{p}_i^{va}[k+1] &= \mathbf{p}_i^{va}[k] + T \mathbf{v}_i^{va}[k] + \frac{T^2}{2} \mathbf{u}_i[k] \\ \mathbf{v}_i^{va}[k+1] &= \mathbf{v}_i^{va}[k] + T \mathbf{u}_i[k] \end{aligned} \quad (3.20)$$

where  $T$  is sample time and  $\mathbf{u}_i[k]$  is control input at  $k$  discrete time index. In discrete domain,

virtual agent control input is defined as:

$$\mathbf{u}_i[k] = \mathbf{F}_{attr/rep_i}^{va}[k] + \mathbf{F}_{align_i}^{va}[k] + \mathbf{F}_{nav_i}^{va}[k] \quad (3.21)$$

As we discussed in previous section, formation control law consists of pairwise potential term, velocity consensus term and navigation rule which is a position consensus term. In the literature there are only formal approaches to examine the stability of the consensus in double integrator dynamics. The goal is to determine for which sample times the system will still be stable. In [65] authors applied formal approach based on algebraic graph theory. The goal was to define sample time  $T$  in relation to consensus gain that will keep the system stable. Similar approach but using Lyapunov function was proposed in [40].

In our case, we examine control law which consist of consensus term and potential field term which depends on distance between neighbors. Hence, it is not straightforward how to express this system in linear form which could allow us to use algebraic graph theory and linear algebra to analyze stability. Therefore, firstly we examine the performance in the simulation for different sample times and give ideas and future directions for a formal proof. We run simulations in Matlab for different sample times and we measure deviation from equilateral triangular lattice, defined as:

$$E_{dev} = \frac{1}{1 + n_l} \sum_{i,j \in I^{drone}} \left( \frac{d_{ij} - d_d}{d_d} \right)^2 \quad (3.22)$$

where  $n_l$  is number of links in the lattice,  $d_{ij}$  is Euclidean distance between virtual agents  $i$  and  $j$  and  $d_d$  is desired distance between virtual agents.

In Figure 3.4, the mean deviation energy for different sample times is presented. It is observed as sample times are increasing, and agents are exchanging less messages per second, the formation stability is violated. To achieve stable equilateral triangular formation, agents should exchange on average 10 messages per second. For theoretical analysis of this phenomenon, we could adopt approach presented in [40]. A Lyapunov function of the formation should be defined and the change of Lyapunov energy function could be examined for two discrete time steps. This could provide an insight whether the energy of the system is decreasing for certain range of sample times.

### 3.3 Experiments and results

This section presents simulation and field results of the proposed formation control algorithm.

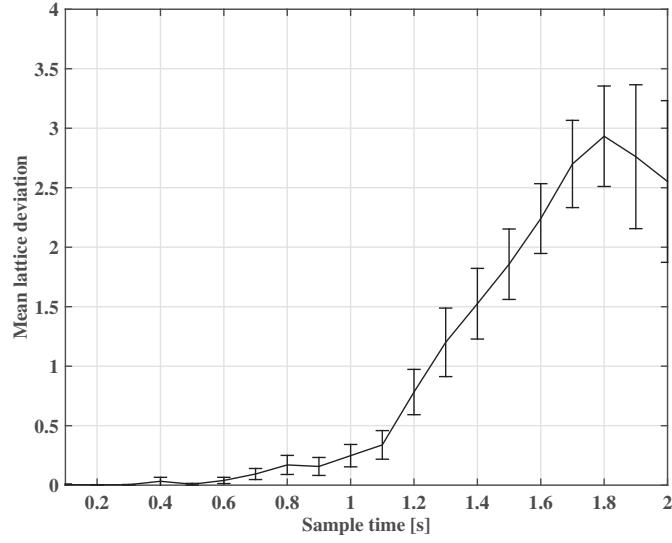


Figure 3.4: Deviation from equilateral triangular formation for different sample times. Mean deviation energy and standard deviation is presented. For each sample time, experiment is conducted in 50 trials. We observe that formation deviates from ideal equilateral lattice for higher sample time values. To achieve stable coherent formation, agents should exchange on average 10 messages per second.

### 3.3.1 Simulation experiments

Multiple experiments were performed in simulation and in field with real drones to test the formation control algorithm for different inter-drone distances and different communication link qualities. The performance of the formation algorithm is quantified using three measures. As a performance criteria we introduce the distance measure between each pair of neighboring drones averaged over the number of pairs. This measure gives us the notion on the quality of adaptation to the user-defined inter-drone distance. In addition to distance measure, we define formation energy deviation measure which identifies the formation deviation from an ideal equilateral triangular lattice given by Eq. 3.22. Furthermore, we examine how well drones adapt to the user-defined communication link quality, hence we measure the average link quality in the formation. Finally, we measure the mean phase angle in the swarm to show how fast drones synchronize their phase angles, how the synchronization degree affects the distance and link quality adaptation algorithm.

Simulation experiments were conducted in a 2D simulator implemented in Matlab. The AVLib Simulink library [57] and custom developed Simulink modules were used to simulate the fixed wing drone's and the motion of the virtual agents. We assume that drones are flying on

the different altitudes since at this stage we do not implement collision avoidance algorithm. Simulation parameters were tuned to best represent the eBee drone platform [2], that was used throughout the real experiments. Drones fly at speeds in the interval  $[12, 16]m/s$  and the turn rate of the robot is also limited in the interval  $[-1, 1]rad/s$ , due to the limited bank angle of the platform. We examine how do drones regulate the inter-drone distance in a formation and whether they create equilateral triangular lattice. Drones get information from the user which distance they should keep with their neighbors. Drones should keep inter-drone distance of 30m then increase the distance to 50m, and decrease the distance to 30m. The desired inter-drone distance profile is known to every drone in the formation. If drones are running as shown in Figure 3.5a, average distance measure closely follows desired distance and formation of the virtual agents is mapped on the formation of respective flying robots, Figure 3.5b. A deviation between desired and average distance occurs only at the beginning of the formation algorithm, since robots start the algorithm in a desynchronized state. The response time is affected by the speed of convergence of virtual agents and speed of synchronization of the drones. The response time to the commanded step signal is significantly shorter than the average flight time of the fixed wing drone.

Figure 3.6 shows different phases of the formation change in a simulation experiment consisting of 10 drones. Drones start from random positions and orientations in the radius of  $20m$  from the launching point. Each robot initializes the position of its virtual agent to the random position within  $10m$  from the launching point. The subfigure a) shows the state of the formation after the launching. Virtual agents start to organize in a uniform formation and drones follow their motion. Drone's phase angles are not synchronized, hence their arrangement does not resemble to the arrangement of their virtual agents. In subfigure b), drones start to synchronize their phases while they follow the motion of their virtual agents. The arrangement in the virtual agents' formation converges to uniform formation. In subfigure c) after 400s of the adaptation, drones form a hexagonal structure that resembles to the formation of their virtual agents. They also increase their inter agent distance while synchronizing their phases. In the subfigure d), drones decrease their inter agent distance and still keep a hexagonal formation.

### 3.3.2 Field experiments

Furthermore, to endorse our simulation results, we run field tests with 3 fixed wing drones. Ebee fixed wing drones are equipped with an autopilot which takes care of autonomous take-off, landing and low level control, and Gumstix microcomputer that runs communication

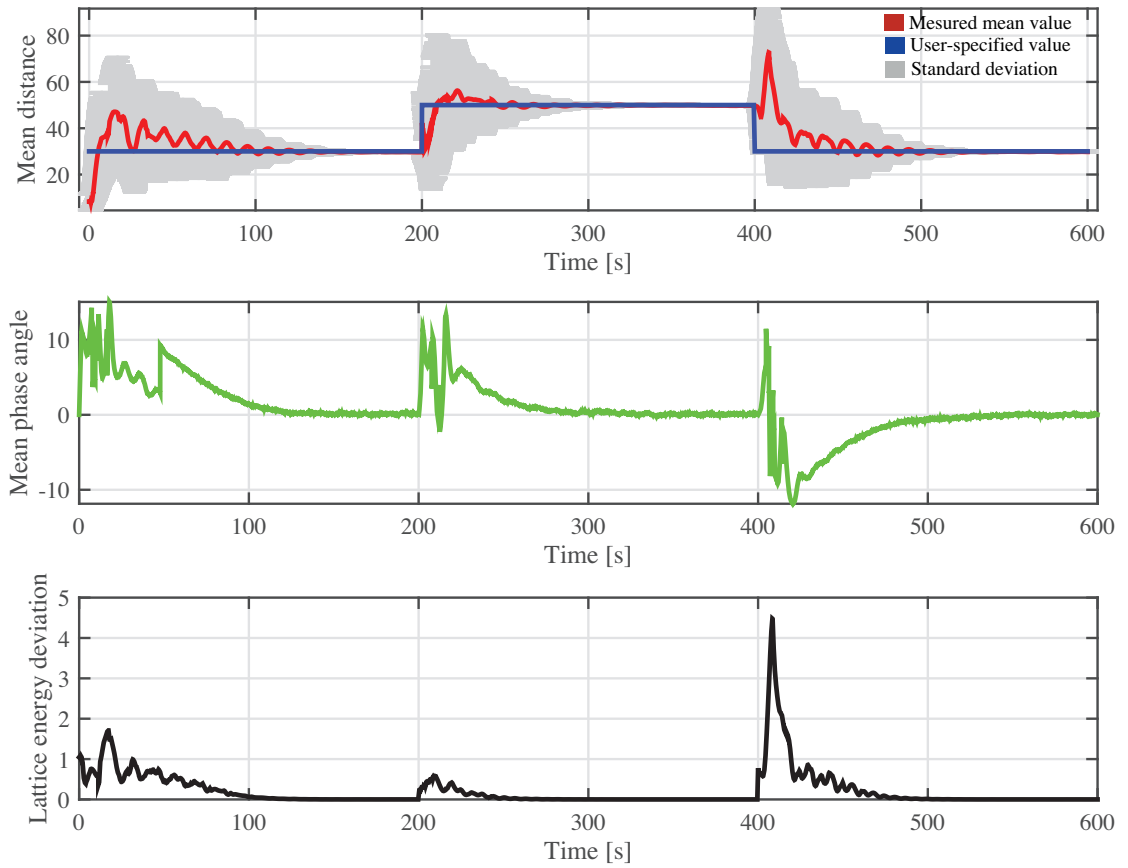


Figure 3.5: Average distance between drones and lattice energy deviation in the formation. Experiment is conducted with 10 drones. Drones regulate their distance based on the desired distance, set by the user. a) Mean distance for the formation control b) Average phase angle deviation. c) Lattice energy deviation

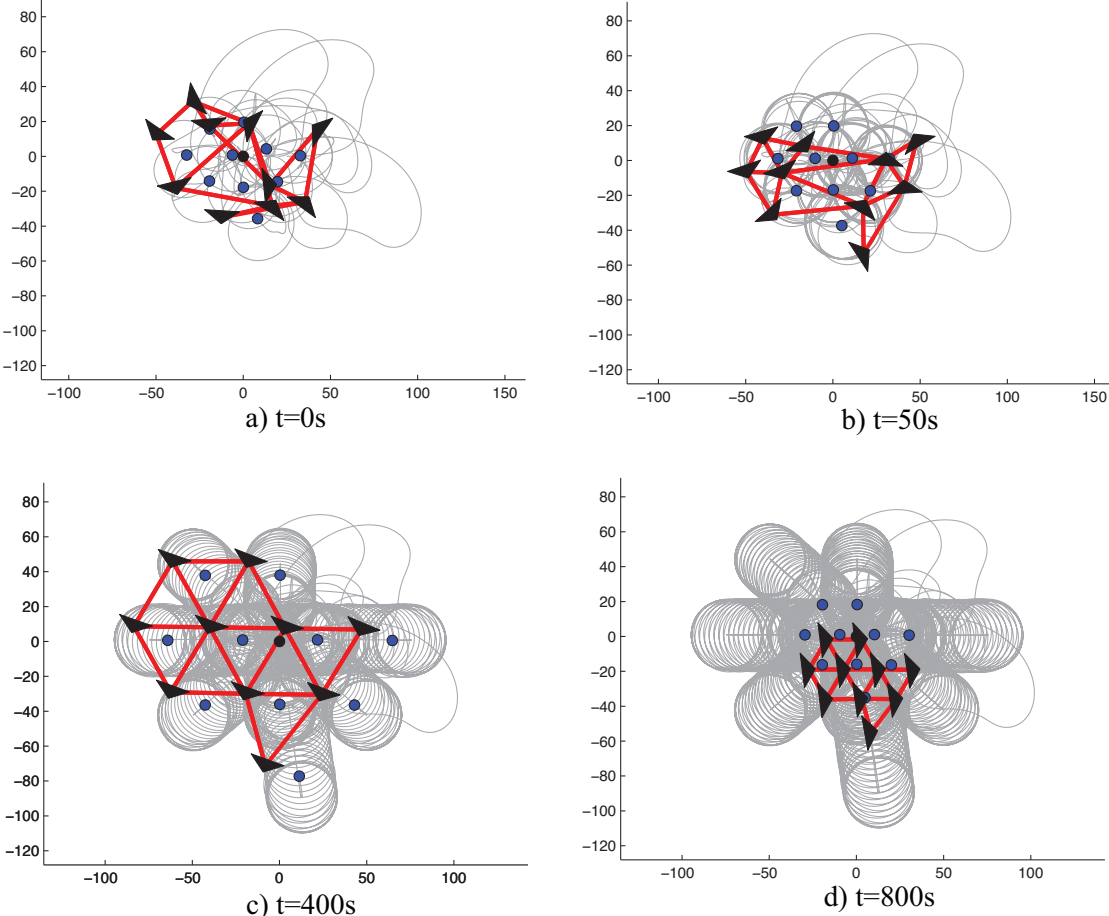


Figure 3.6: Different phases in distance adaptation of the flying robot formation. Drones start from random positions, synchronize their motion, form a uniform formation and increase/decrease inter-drone distance in a decentralized and distributed way.

algorithm and formation control algorithm. Each drone was equipped with Linksys AE3000 USB dongle using 802.11n. To avoid interferences, the frequency was set to 5 GHz. Drones established an ad-hoc wireless network and each drone sent message packets to its single hop neighbors. A message packet consist of its GPS position, course angle, position and velocity of its virtual agent.

Field experiments were conducted with 3 eBee fixed wing drones. States of all drones were monitored at the ground control station running eMotion software [3]. Robots were autonomously launched and they circle on their predefined home waypoint. User initiates the experiment by sending a message from ground control station. Each drone has desired distance stored in its memory. Figure 3.7 shows the aerial snapshot of the field experiment.

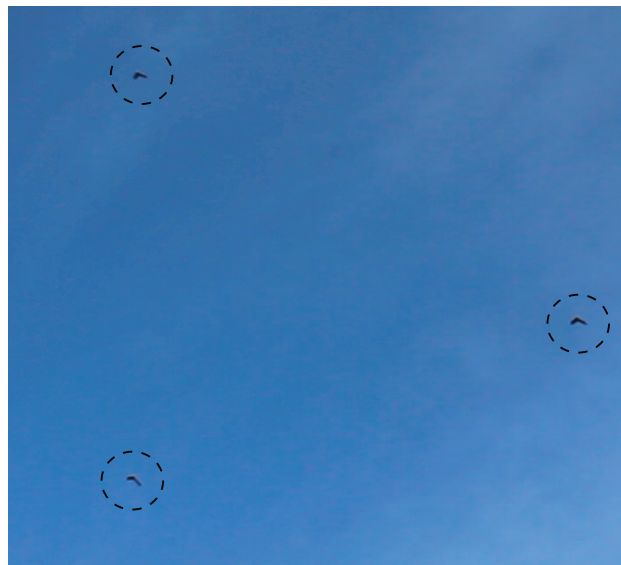


Figure 3.7: A snapshot of the field experiment showing 3 eBee drones, marked by dashed circles, keeping the distance of 30m between their neighbors.

Figure 3.8 shows the path of three drones for two different sequences of a field experiment, where a triangular formation of 30 meters and 50 meters is obtained. Drones are performing circular trajectories around their respective virtual agents.

Figure 3.9 shows the performance measures, mean distance and phase angle for the field experiment. We measure mean distance between each drone in the formation and between their respective virtual agents. At the beginning of the experiment, drones are not synchronized, virtual agents did not converge to their stable positions, hence there is a higher deviation in the mean inter-drone distance from the user specified distance. As experiment progresses, drones synchronize their phase and the deviation is within  $\pm 5m$  of the desired distance (30m). During

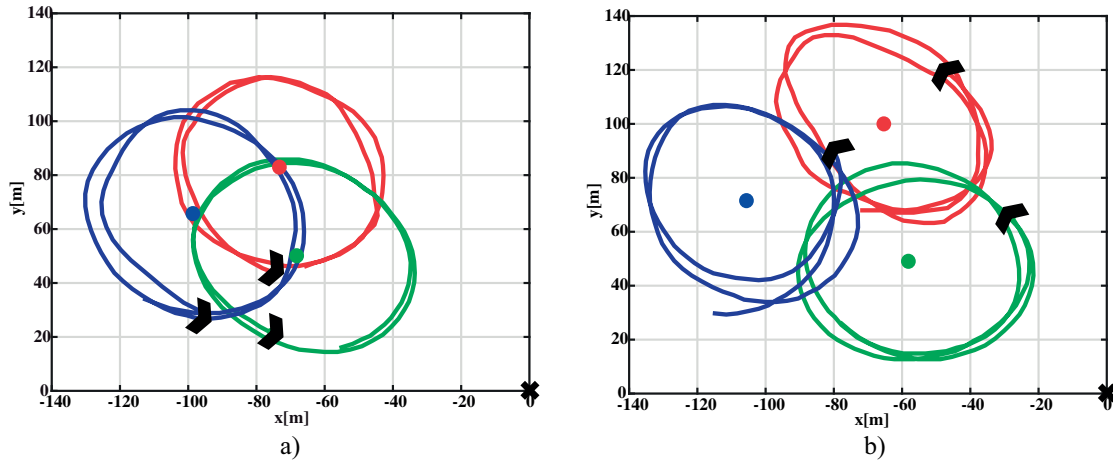


Figure 3.8: The motion paths of 3 drones in field experiment in which they keep distance of a) 30m and b) 50m with their neighbors. Motion paths are plotted relative to the launching point of the experiment (0,0) denoted by a black cross.

the drones transition from distance of 30m to distance of 50, drones do not desynchronize significantly. The higher deviation between user specified distance of 50m and mean drone distance in the interval [100,120s] can be explained by the fact that drones do not always follow ideal circular trajectories (due to wind) and such deviations can occur despite the fact that robots are synchronized.

### 3.4 Collision avoidance using altitude segregation

In multi-agent aerial swarms, an important problem to solve is a mid-air collision avoidance. Potential collisions can cause damage of the drones and jeopardize the whole mission. Approaches proposed in literature are based on predictive optimization algorithms, negotiation between agents, geometrical methods and probabilistic methods. Geometric collision avoidance methods usually define collision conditions for conflict in the position space [62] or in the velocity space [20, 5]. In probabilistic methods a probability of conflict is estimated [49] and used to control the drone. In predictive control, an objective function is defined respecting constraints such as command limits, limits on the displacement that a vehicle can carry out [14]. In this thesis, we propose reactive collision avoidance based on altitude segregation. Since the collision avoidance on the virtual agent level is guaranteed, we use a level of synchronization of fixed wing drones to define consensus algorithm on the inter-drone altitude.



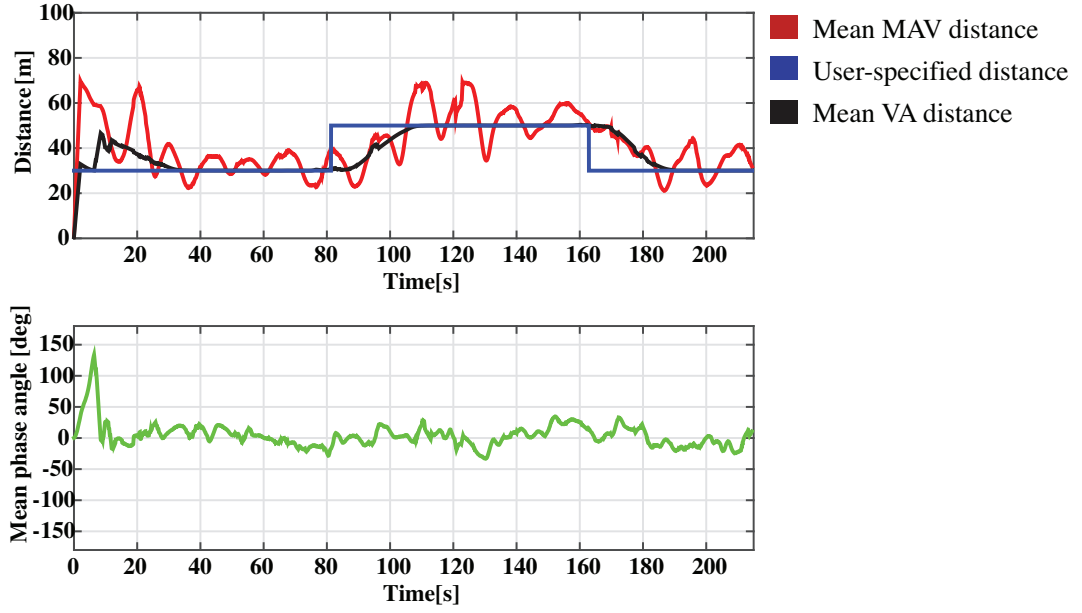


Figure 3.9: Mean distance and phase angle measurements for 3 drones in the field experiment. The desired distance profile is stored in drones memory before the experiment.

### 3.4.1 Proposed method

In terms of virtual agents collision avoidance it has been shown in [60] that equilateral triangular lattice can be created without agents collisions. When virtual agents create equilateral lattice, we can guarantee that loitering centers of drone motion will be on the user-defined distance.

In equilibrium case, if distance between virtual agents is greater than the drone loitering radius, it can be guaranteed that the drone paths will be collision free. In the case when distance between virtual agents is lower than the loitering radius, we should design collision avoidance strategy. If drones' phase angles are synchronized, we can guarantee that their 3D distance will be limited by the virtual agent's distance. In this case, drones can fly on the same altitude and have a collision free paths. In the non-synchronized case, the distance between drones can be lower than the virtual agent distance and collisions can occur. In this case, drones should fly on the different altitudes. To achieve this, we employ consensus algorithm on the altitude of the robots, based on the level of synchronization. The control input that commands desired altitude for  $drone_i$  is given by:

$$\Delta h_i = K_h \frac{1}{(n-1)\pi} \sum_{j \in C_i^{drone}} (\theta_j - \theta_i) \quad (3.23)$$

where  $n$  is number of drones within communication neighborhood.

### 3.4.2 Results

We run simulation experiment in which drones start flying on the same altitude and  $(x, y)$  starting positions are randomly defined within the radius of 30m from the initial point  $(0, 0)$ . In Figure 3.10 a maximum 3D distance between drones during the flight time is presented for formation algorithm with collision avoidance. It is observed that if collision avoidance algorithm is applied, the maximum 3D drone distance is always higher than 5m which is defined as safety distance.

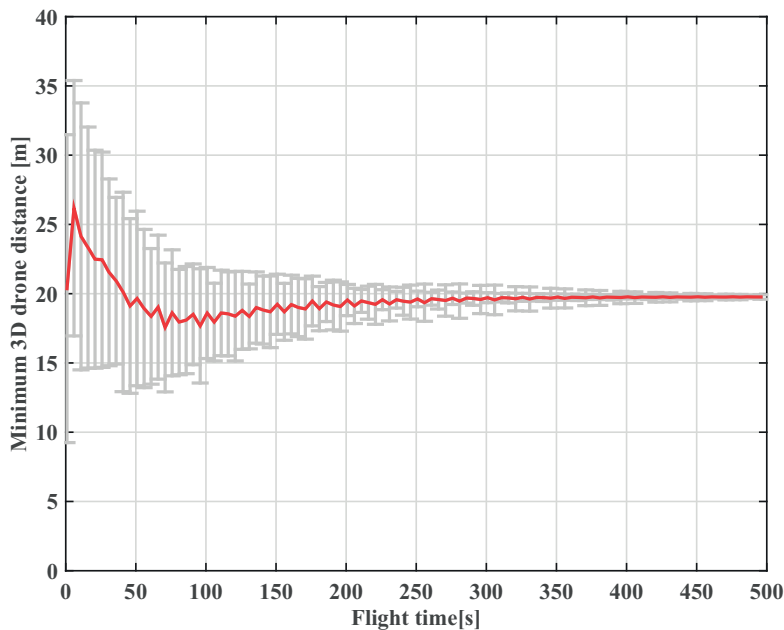


Figure 3.10: Minimum drone 3D distance, drones run collision avoidance algorithm and formation control algorithm in which they should keep 20m distance between their neighbors. Drones loitering radius is 30m. The experiment is conducted in 100 simulation runs with 5 drones.

## 3.5 Use case: Communication link adaptation

In the simulation experiments we examine how do drone teams regulate inter-drone distance and we measure how do they perform uniform coverage while making sure that communication link quality is above a predefined threshold value. The communication link quality is modeled based on the data collected in outdoor experiments. In collaboration with ETHZ TIK laboratory we ran experiments in which drones were flying away and towards each other, UDP

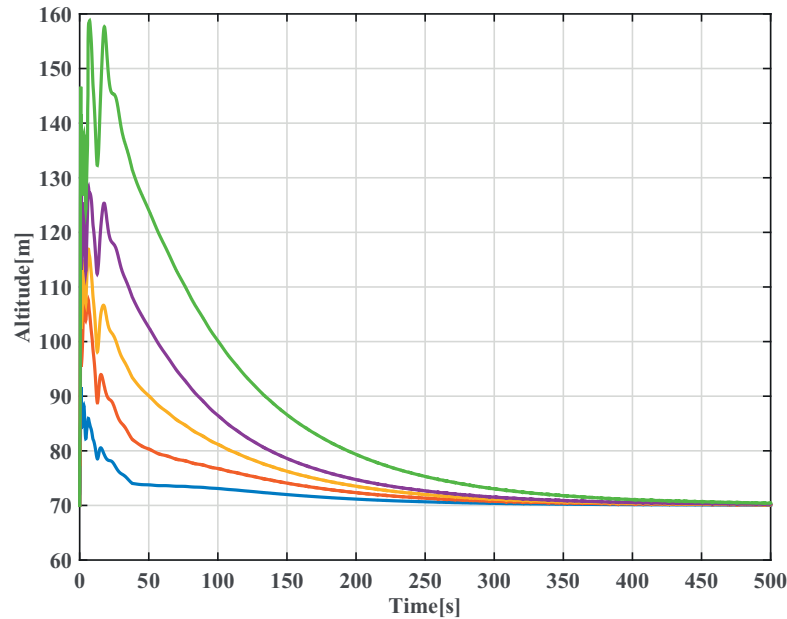


Figure 3.11: Altitude profiles of all drones during one simulation experiment

traffic was set to high rate 80Mb/s and *iperf* UDP bandwidth measurement tool [1] was used to measure Signal-to-Noise ratio (SNR) and Received-Signal Strength (RSSI). A polynomial regression model was then used to represent the obtained data. Since experimental data is noisy we superpose the white noise signal to the polynomial fit. The experimental data is given in Figure 3.12

### 3.5.1 Characterization of wireless link quality

A choice of the strategy for communication maintenance and area coverage in aerial networks highly depends on the technical properties of the mobile ad-hoc networks. Mobile ad-hoc networks (MANETs) are formally defined as an autonomous system of mobile nodes connected by wireless links which form a communication network modeled in a form of an arbitrary communication graph. MANET is peer-to-peer, multi-hop network in which information is flowing from a source to a destination node through intermediate nodes. A relative position, orientation and velocity of the nodes has high influence on the communication bandwidth and delays in delivering information from the source to the destination. In this section the performance measures will be introduced together with the factors that influence degradation of the communication performance in aerial and ground-aerial wireless ad-hoc networks.

In wireless networks, the information is carried by radio signals on typically 2.4 or 5 GHz frequency. To determine and optimize the communication performance, measures as re-

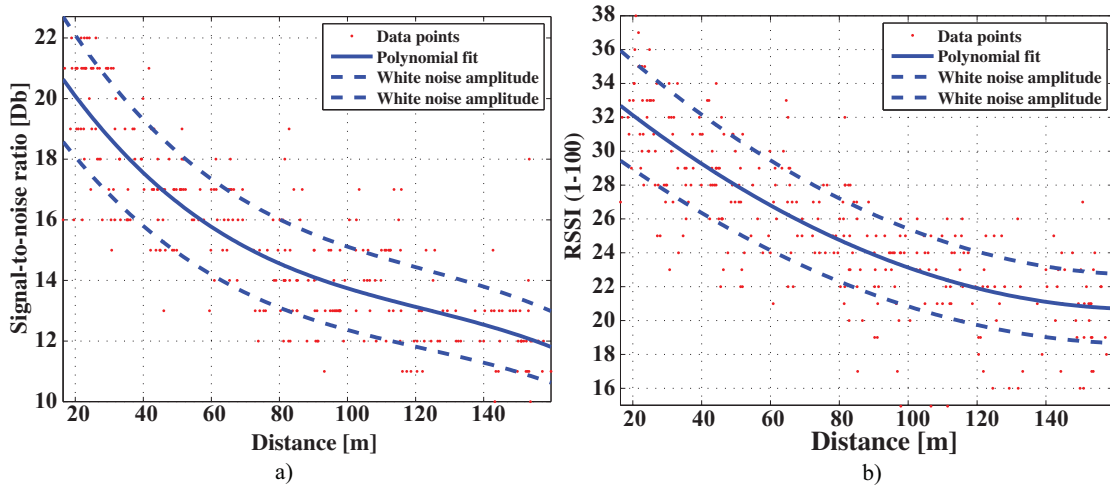


Figure 3.12: Link quality, RSSI and SNR measures obtained in field experiments with 2 fixed wing drones.

ceived signal strength (RSSI) , Signal-To-Noise ratio (SNR), packet loss, throughput, Expected Transmission Count (EXT) [23], Expected Transmission Time (ETT) [27] are usually included in the cost functions. In ideal free space propagation, the power of the transmitted signal drops with the square of the distance from the transmitter. The propagation of the signal is spherical. In urban and suburban environments, due to effects of scattering, reflection and diffraction from obstacles, the propagation models vary and depend on the specifications of the terrain. Besides environment specifications, in MANETs the communication performance also depends on the dynamics of the mobile nodes. Since we are considering fixed wing robots with highly restricted kinematic properties, two studies experimentally assessed the performance of the communication by measuring throughput and EXT/ETT in different flight experiments. The first study by Jimenez-Pacheco et al. [45] was conducted on two flying fixed wing robots and one static ground node in a single and multi-hop scenario for 802.11s mode (a mode for mesh networks). The results show that throughput degrades with distance and already around 100m from the ground station results show a big spread. There are many results achieving throughput values close to the maximal value, but there are around 25% of the results that had low throughput. It shows that there is a more complex dependency of the throughput on the flight parameters of the fixed wing robot, not just the distance. They also conducted the experiments to further investigate the dependency of the flying robot trajectory on the throughput. They showed that after certain distance, link behaves better if the plane is facing ground node then when it is flying away from it. This results could be explained by the position of the antenna on the left wing of the robot which affects antenna radiation pattern.

The second study conducted by Asadpour et. al. [8] was performed on the IEEE 802.11n mode, to discover the effects in throughput between two mobile nodes. The authors showed that communication in 802.11n mode performs poorly in highly mobile scenarios and that throughput in a single hop case between 2 UAVs drops significantly below the theoretical value. Their key conclusion is that 802.11n cannot cope with highly mobile wireless channels since current wireless implementations do not take this phenomena into the account. Therefore, flying networks would benefit of flying in a formation and keeping the constant spatial arrangement of the network nodes.

#### 3.5.2 Adaptation algorithm

Sometimes it is of interest to expand this formation uniformly to cover the maximum possible area that can be achieved while having a desired link quality between the drones. To achieve this, drones measure the communication link quality between their local neighbors and increase their distance with them until they reach the desired link quality. The schematics of the link quality adaptation control algorithm is given in Figure 3.13.

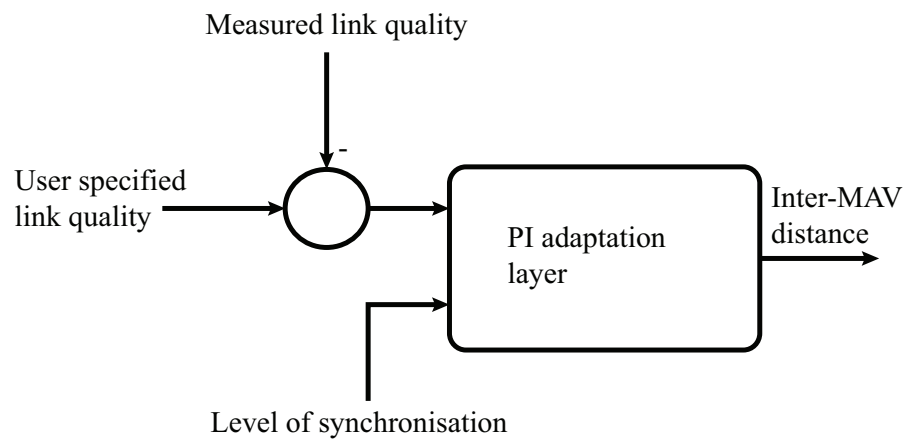


Figure 3.13: Schematics of the link quality adaptation layer.

Every drone measures its current communication link quality with its neighboring drones, compares it with a user-defined link quality and a controller outputs the desired distance that drone needs to keep with its neighbors.

$$d_{ij}(k) = d_{ij}(k-1) + \Delta d_{ij}(k) \quad (3.24)$$

where  $\Delta d_{ij}$  is defined as:

$$\Delta d_{ij}(k) = K_{adapt}(q_{ijdesired} - q_{ijmeasured}(k)) \quad (3.25)$$

where  $q_{ij}$  is the link quality measure between drone  $i$  and drone  $j$  and  $K_{adapt}$  is a tunable gain that determines the speed of adaptation. In this use case we use two link quality measures, signal to noise ratio (SNR) and received signal strength indicator (RSSI). Tunable gain is given by:

$$K_{adapt} = 1 - \frac{\sum_{j \in C^i} (\theta_i - \theta_j)}{\pi(n-1)} \quad (3.26)$$

where  $n$  is a number of robots within communication neighborhood of the drone  $i$ .

The speed of this control process is adapted to the state of the synchronization of each robot. If robots have not synchronized their phase angles with their neighbors, then the gain of the controller is continuously reduced to slow down the adaptation and to allow the robot to first synchronize and then to adapt the distance.

### 3.5.3 Simulation results

In the second simulation experiment, we combine the formation algorithm with communication link adaptation layer. We show how does such algorithm adapt to the user specified link quality between agents. The user requires from the swarm of drones to increase the distance between neighboring drones until the communication link quality reaches a threshold, as shown in Figure 3.14a,b. We measure average link quality between neighbors and compare it with the link quality threshold set by the user. Since drones start the experiment in the non synchronized state, adaptation layer measures the level of synchronization and decrease the gain of the adaptation. In this way adaptation layer of each robot continuously regulates the speed of adaptation and allows each robot first to synchronize with their neighbors and then to adapt the link quality. We can observe that mean phase angle reduces to less than 10 degrees. drones manage to increase their inter-drone distance in the the interval of 100s, which is still significantly shorter than the flight time of the platform (45 min for eBee platforms).

## 3.6 Conclusion

In this chapter, a formation algorithm that allows fixed wing drones to regulate the distance between their neighbors in a formation was presented. Drones can uniformly cover an area

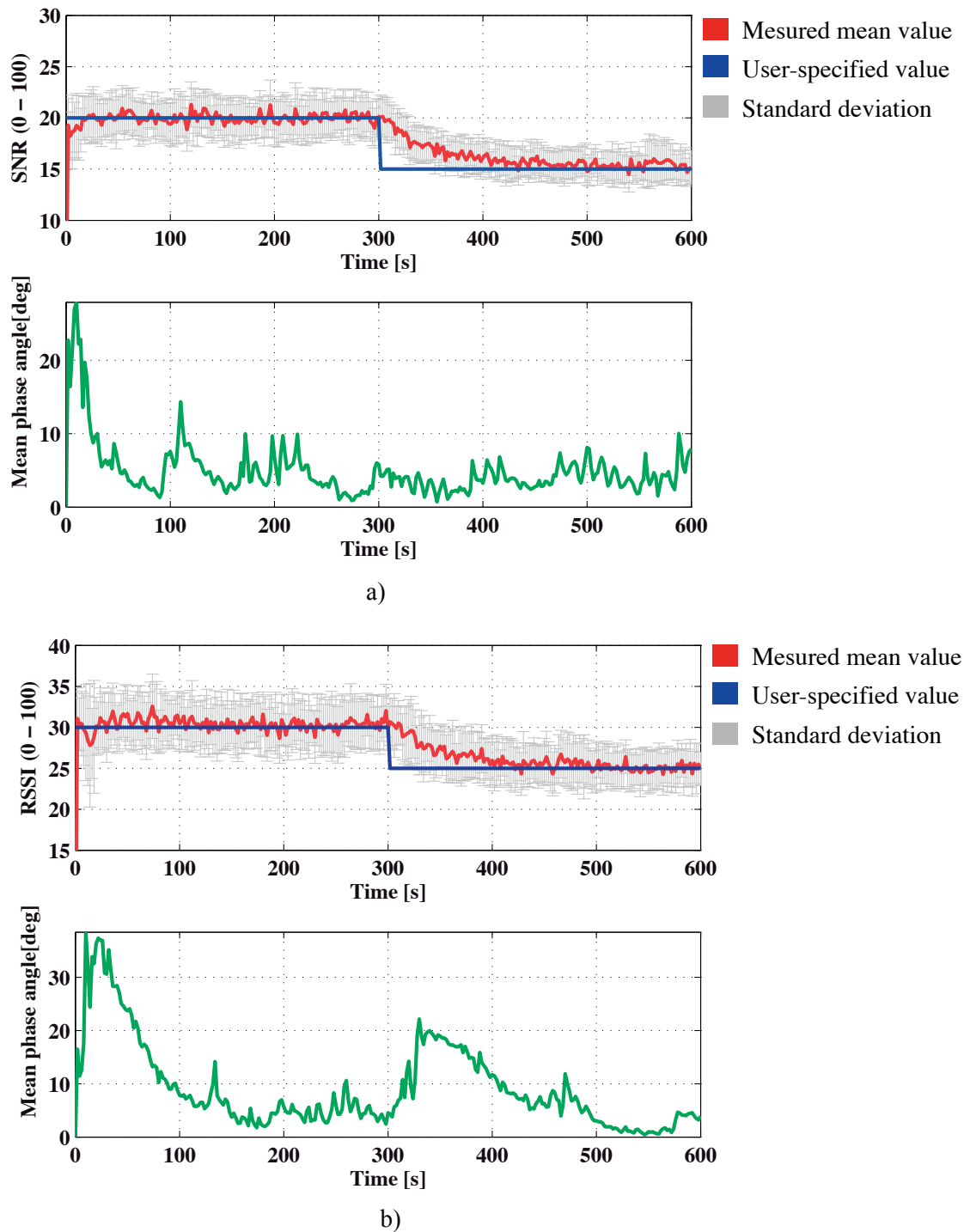


Figure 3.14: Fixed wing drones are adapting the quality of communication link between their neighbors. The desired communication link quality is set by the user. We show that drones are able to adapt the formation to achieve desired link quality. They run formation algorithm combined with adaptation layer. Adaptation layer controls the speed of adaptation to the desired link quality based on the level of synchronization between robots. If robots are not well synchronized, adaptation layer will slow down the adaptation to allow them to synchronize and then it will speed up the adaptation when robots reach desired level of synchronization.

### **Chapter 3. Formation control with fixed wing drones**

---

while keeping communication link quality above the threshold value.

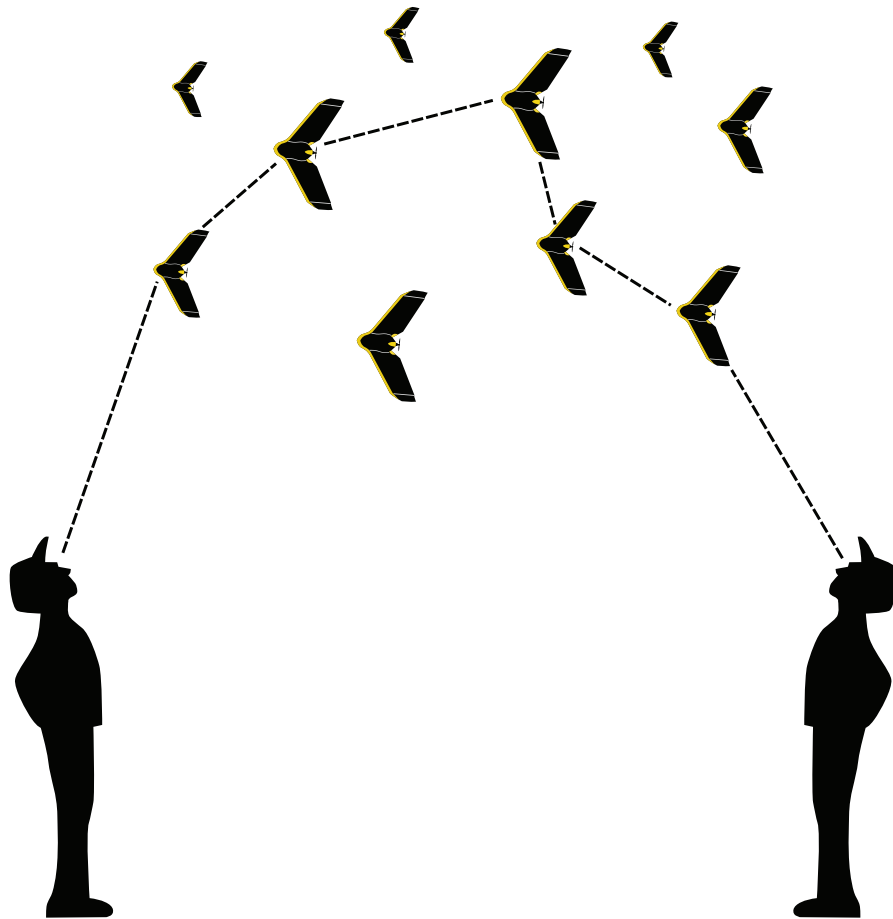
We observed that drones are able to follow instructions given by the user to increase or decrease distance only if they synchronize the phase angles with their neighbors. In this case, robots distributively create the same formation as their respective virtual agents. A collision avoidance algorithm is implemented and it is based on phase synchronization level and altitude segregation. It was shown that drones successfully avoided collisions while deploying in the formation.

In field experiments drones have to deal with asynchronous communication, delays in communication and wind disturbances. Despite these challenges, drones are capable to create desired formation and keep the desired distance between their neighbors.

Furthermore, it was shown that the method is well suited for maintaining uniform formation and controlling the communication link quality. By adding an adaptation layer that controls the dynamics of the synchronization of robots and dynamics of virtual agents, drones are able to successfully adapt the communication link quality. Hence such formation algorithm can be used in aerial ad-hoc wireless network where robots should adapt to the user requirements, either to cover larger areas with flying beacons while keeping good communication links or to increase the link quality if the transfer of a large amount of data (images or sound) is requested by ground users.



## 4 Connectivity maintenance



This chapter is based on following publications:

Varga, M, Basiri, M., Stovold, J., Heitz, G. & Floreano, D. Communication maintenance in drone networks. Under submission.

## 4.1 Introduction

Successful decentralized multi-drone operations rely on information exchange between drones. Drones within an aerial team need to retrieve information about behaviors, position, velocity and tasks of other drones. Some information, such as relative range and bearing can be determined locally with on-board exteroceptive sensors [11, 10, 69]. In addition to that, the detection range of on-board sensors is limited and communication network is needed to share positioning information among all individuals. Furthermore, contextualized information such as neighborhood sets or current tasks, need to be transferred by a communication network. Therefore, to achieve some autonomous collective behaviors, drones need to be connected.

Teams of drones that are connected by a wireless network could also serve as aerial communication relays. Drone networks are relevant in search-and-rescue operations where mission success depends on the communication infrastructure. Rescuers often need to share mission plans between rescue teams. They also need to transfer bulky data such as images, video or voice messages. Currently, rescue teams rely on GSM networks or hardly portable WiMAX antennas. In mountains regions, signal coverage for such networks is poor due to terrain occlusions. Search and rescue teams would benefit from their own drone network, which could provide line of sight communication and relay nodes that fly above the obstacles. Furthermore, in disastrous situations such as massive earthquakes, fires or floods, communication infrastructure can be heavily damaged. In such situations a network of drones can serve as a rapidly deployable communication network allowing communication and monitoring the area.

To stay connected, drones need to run distributed connectivity maintenance control algorithm. The task of such algorithm is to derive a decentralized control input that takes into the account the limited sensing range of each agent so that the communication graph remains connected. It means that communication path between every two agents in the network always exists, but initial links do not have to be necessarily maintained [25].

Connectivity maintenance methods can be divided in three groups, methods based on potential fields control, graph theory methods and optimization methods. Potential field based connectivity maintenance method uses local observations to derive potentials and gradient based control to guide the agents to stay connected with their neighbors. In [25], Dimarogonas et. al. proposed a method to maintain all the links in already established communication network by defining a bounded potential field function. The control scheme (i) maintains the edges of the graph which were set in initial conditions and (ii) drives agents to a com-

mon, known position in state space. Another application of maintaining communication in heterogeneous systems using a gradient based approach is proposed by Hsieh et. al. [42]. Authors propose aerial-ground robots network where both robot groups perform a gradient based control to maintain communication. The authors assume that the ground robots network topology is preserved throughout the experiment. In [83], Tardioli et. al. used a virtual forces approach to maintain connectivity between a fixed node and moving ground robots. They used spring forces model to maintain fixed topology while performing an exploration task. Potential field based methods are more suited in applications where network topology changes rapidly, in heterogeneous swarms where agents differ in mobility and controllability and robots should react in a reactive mode to the changes in the environment.

Graph theory based methods rely on global properties of the network and they use mathematical formalisms such as a weighted Laplacian matrix and Fiedler values to derive next agent's actions. The goal is to define agent's actions which will not violate network connectivity. In [56] Michael et. al. presented an approach which relies on estimates of the network topology. They use gossip algorithms [15] and the distributed market-based control [19] that allow link deletions without violating connectivity. This paper also investigates practical issues with implementing this methodology on a real system of homogeneous ground robots. Similar approach was presented by Stump et. al. [79] in which authors developed an algorithm to control a team of robots to maintain the communication bridge between stationary robot in the environment with obstacles. Their approach relies on calculating a Fiedler value of the weighted Laplacian matrix which describes communication interactions between all robots and they use a k-connectivity matrix to detect which robots can interact through k or less intermediary robots. In approach presented by Cortez et. al. [22], authors are defining motion constraints of agents in a heterogeneous group consisted of sensing robots and relay nodes. By using distance based connectivity properties of the network, they define the control of relay nodes to extend and maintain communication with sensing robots. The main advantages of the graph theory based methods is the fact that they provide valuable insight in the network topology and allow the usage of known distributed search algorithms on graphs to identify communication bottlenecks in the network.

In the paper proposed by Gil et. al. [33], authors use distributed optimization to maintain connectivity. Quadrotor drones are used in this method to improve the communication quality between static ground nodes. Their cost function is defined by Signal to Interference Ratio between neighboring robots and control algorithm follows distributed gradient descent on the cost function. This approach is demonstrated in reality using few static nodes and 3-5

quadrotors in an indoor experiment.

In all presented approaches, authors assume that ground nodes are static or they are cooperative and their motion can be controlled. However, in this work we address the problem of connectivity maintenance between mobile users on the ground, where their motion cannot be controlled. They are independently moving nodes. Similar problem has been addressed by Gil et. al. [32]. Authors use centralized optimization algorithm to place relay nodes to maintain connectivity with independently moving ground nodes. Algorithm assumes all agent positions are updated in each time step. In contrast to the approach proposed by Gil et. al., we propose two decentralized methods to solve connectivity maintenance problem between drones and independently moving ground nodes.

The first method is based on formation control algorithm proposed in Chapter 3. Drones create equilateral triangular lattice using only local information. Then, they move the formation's center of mass and adapt distances between neighbors based on the information they receive from ground nodes. In the second method, drones also exchange their interaction topology matrices and using graph theory methods break interaction links to transform the interaction graph to a spanning tree. We implement methods in hardware in the loop emulator and test the performance in field experiments. We compare methods for different number of ground nodes.

This chapter is organized as follows: Section 4.2 introduces important mathematical concepts used in this chapter. The first connectivity method, potential field based method is presented in Section 4.3. In Section 4.4, we introduce second method based on market-based approach. Finally, Sections 4.5 and 4.6 conclude the chapter by providing a conclusion and future work on this topic.

## 4.2 Theoretical background

Theoretical framework in this chapter relies on the graph theory concepts. In this section, the definitions which will be used throughout the chapter are presented. Similarly, as in the previous formation control method, it is assumed that fixed-wing drones are controlled using virtual agents approach and Lyapunov guidance vector field method. Therefore, we present interaction rules between virtual agents and we assume that drones are following them by performing orbital motion. Every virtual agent and ground agent is presented with one vertex in the graph. The distinction is made between interaction and communication graph. An interaction graph  $\mathcal{G}_{int}$  is defined as the pair  $\mathcal{G}_{int} = (V, \mathcal{E}_{int})$  where  $V$  is vertexes set and  $\mathcal{E}_{int}$  is

## Chapter 4. Connectivity maintenance

---

an edge set defined as:

$$\mathcal{E}_{int} = \{v_i v_j | v_i \in I_j^{drone}, v_j \in I_i^{drone}, i \neq j\} \quad (4.1)$$

where  $I_i^{drone}, I_j^{drone}$  are interaction neighborhoods of drone  $i, j$  respectively. Analogously, a communication graph  $\mathcal{G}_{comm}$  is defined as the pair  $\mathcal{G}_{comm} = (V, \mathcal{E}_{comm})$  where  $V$  represents a vertexes set and  $\mathcal{E}_{comm}$  is an edges set defined as:

$$\mathcal{E}_{comm} = \{v_i v_j | v_i \in C_j^{drone}, v_j \in C_i^{drone}, i \neq j\} \quad (4.2)$$

where  $C_i^{drone}, C_j^{drone}$  are communication neighborhoods of drone  $i$  and  $j$ .

Among diverse graph structures, tree graphs and graph cycles are relevant for the proposed method. These structures are defined using properties of the graph paths. A path of graph  $\mathcal{G}$  of length  $m$  is defined as a sequence of distinct vertices

$$v_0, v_1, \dots, v_{m-1} \quad (4.3)$$

such that any  $k \in 0, 1, \dots, m-1$  vertices  $v_k$  and  $v_{k+1}$  are adjacent.

If  $v_0$  and  $v_{m-1}$  are defined as end vertices and  $v_1, \dots, v_{m-2}$  as inner vertices then a path that has all vertices distinct except the end vertices is called a cycle path. If a graph does not have cycles it is called a forest. A subgraph of the forest graph is a tree graph. A tree is an undirected graph in which every two vertices are connected with the path without cycles. For a tree graph  $\mathcal{G}$ , following properties follow:

- $\mathcal{G}$  is connected
- If one edge is added to the graph  $\mathcal{G}$  it will form a cycle
- $\mathcal{G}$  becomes disconnected if only one edge is removed from the graph
- End vertices have degree 1 and inner vertices have degree 2.
- $\mathcal{G}$  has  $m-1$  edges

End vertices of a tree graph are called leaves. A spanning tree of an undirected graph  $\mathcal{G}$  is a tree graph that contains all vertices of the graph  $\mathcal{G}$ .

In our connectivity maintenance methods, ground nodes and virtual agents formation are presented as interaction graphs. A communication network between ground nodes and

drones is described as a communication graph between virtual agents and ground nodes. Findings in the algebraic graph theory are used to analyze properties of these graphs. Algebraic graph theory uses matrices to represent graphs and defines algebraic methods to analyze the properties of these matrices. Moving from the combinatorial interpretation of a graph to an algebraic interpretation, the dynamical properties of the graph can be analyzed.

A degree matrix is used to represent the number of neighbors for each vertex. It is defined as a diagonal matrix:

$$\Delta(\mathcal{G}) = A_{m,n} = \begin{pmatrix} d(v_1) & 0 & \cdots & 0 \\ 0 & d(v_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d(v_{m-1}) \end{pmatrix} \quad (4.4)$$

Adjacency matrix is used to encode adjacency relationships in the graph. It is a symmetric and quadratic matrix defined as:

$$[A(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{if } v_i v_j \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (4.5)$$

A Laplacian matrix is defined using degree matrix and adjacency matrix . It plays an important role in dynamical analysis on graphs and in connectivity analysis. Laplacian matrix is given by:

$$L(\mathcal{G}) = \Delta(\mathcal{G}) - A(\mathcal{G}) \quad (4.6)$$

It is symmetric positive semi-definite matrix. This implies that its real eigenvalues can be ordered as

$$\lambda_1(\mathcal{G}) \leq \lambda_2(\mathcal{G}) \leq \cdots \leq \lambda_n(\mathcal{G}) \quad (4.7)$$

From this follows one important result in algebraic graph theory:

**Theorem 1** *The graph  $\mathcal{G}$  is connected if and only if  $\lambda_2(\mathcal{G}) > 0$ .*

Second smallest eigenvalue,  $\lambda_2(\mathcal{G})$  is also called Fiedler value in the literature and it measures algebraic connectivity. If the graph is complete; there is one-hop path between every pair of vertices, then algebraic connectivity is equal to number of vertices. Otherwise, if the graph is

not connected, algebraic connectivity is less or equal to 0. This measure is used throughout this chapter to show the level of connectivity while the drones are adapting to the motion of ground nodes.

### 4.3 Potential field based method

As seen in Chapter 3, fixed wing drones fly in an equilateral triangular formation and change distance to their neighbors based on the user requirements. If drones receive information about position of other drones and ground nodes, they can locally decide which distance they should maintain with their neighbors to keep the aerial-ground network connected. The formation should be able to stretch and contract its shape and translate with the ground nodes motion. To keep the lattice formation, drones implement interaction rules between virtual agents, follow virtual agents and synchronize their phases.

#### 4.3.1 Proposed method

In the potential field based method ground nodes send messages  $MSG_{gn}$  which consist of:

$$MSG_{gn} = \{t_{sent}, \mathbf{p}_{gn}, drone_{closest}\} \quad (4.8)$$

where  $t_{sent}$  is a timestamp when the message is sent,  $p_{GA}$  is a ground node current GPS position and  $drone_{closest}$  is the closest drone to the ground node. Drones synchronize their clocks using server time on the mission computer and ntp algorithm in the Linux operating system implementation. The communication range of each agent in the team depends on the technical properties of the wireless dongle. In this thesis we use wireless dongles with communication radius of 250m [70]. Ground nodes are passive agents in the team. This means that each ground node sends and receives messages from drones within the communication range and calculates the distance to the drones but does not send control commands to drones.

Drones are active relay agents which fly in the formation and adapt to the position of the ground nodes. They communicate both with other drones and ground nodes within their communication range and send messages  $MSG_{drone}$  which consist of:

$$MSG_{drone} = \{t_{sent}, \mathbf{p}_{va}, list_{va}, list_{gn}\} \quad (4.9)$$

where  $t_{sent}$  is a timestamp when the message is sent,  $\mathbf{p}_{va}$  is a virtual agent current GPS position,



$list_{va}$  is a list of all neighboring virtual agents and  $list_{gn}$  is a list of all neighboring ground nodes. Using this information each drone builds ground nodes position map and virtual agent position map. The maps are updated in each control algorithm cycle or when a new message is received from neighboring agents. In each update, current timestamp is compared with the timestamp of the received message and map is updated with newer information. The virtual agent and ground agents' positions are deleted from the map if data is older than  $t_{erase}$ . In this way, formation algorithm is robust to changing number of agents. The time threshold  $t_{erase}$  is determined based on the highest measured delay in the wireless network.

In the potential field based method, drones have two roles. They can be paired with ground nodes or they can be relay agents. Drones become paired when they receive the message from the ground node in which  $drone_{closest} = drone_{id}$ . It means that this drone is the closest agent to the ground node. A drone can be paired only with one ground node. If drone is already paired with the ground node and receives message from that ground node in which  $drone_{closest} \neq drone_{id}$ , the drone becomes a relay node.

Interaction rules, attraction/repulsion, alignment and navigational forces are acting on the virtual agents' motion. Drones need to calculate the distance they need to keep with their neighbors and navigation point which will determine the area where drones need to fly. Since drones store position information in the virtual agents' map and ground nodes positions in the ground nodes' map, they can analyze spatial arrangement of the virtual agents and ground nodes. Virtual agents center of mass is defined as:

$$\mathbf{p}_{cm}^{va} = \frac{1}{n} \sum_{i \in va} (\mathbf{p}_i^{va}) \quad (4.10)$$

The maximum distance between any virtual agent and virtual agent's center of mass is given by:

$$d_{cm}^{va}(k) = \max(\|\mathbf{p}_{cm}^{va}(k) - \mathbf{p}_i^{va}(k)\|), \forall i \in va \quad (4.11)$$

Similarly, the maximum distance between any ground agent and virtual agent's center of mass is given by:

$$d_{cm}^{gn}(k) = \max(\|\mathbf{p}_{cm}^{va}(k) - \mathbf{p}_i^{gn}(k)\|), \forall i \in va \quad (4.12)$$

## Chapter 4. Connectivity maintenance

---

Then, the desired distance is calculated as:

$$d(k) = d(k-1) + K_d \Delta d \quad (4.13)$$

where  $K_d$  is adaptation gain and  $\Delta d$  is defined as

$$\Delta d(k) = d_{cm}^{va}(k) - d_{cm}^{gn}(k) \quad (4.14)$$

The aerial flock will expand or contract by applying this rule. If ground nodes are translating at the same time, robots need to apply navigational force which will make them follow the ground nodes. Therefore drones apply navigational force given by:

$$\mathbf{F}_{nav_i}^{va}(k) = -c_1 f_\sigma(\mathbf{p}_i^{va}(k) - \mathbf{p}_{cm}^{gn}) \quad (4.15)$$

where  $f_\sigma$  is sigmoidal navigation function defined as 3.4,  $\mathbf{p}_{cm}^{gn}$  is the ground nodes center of mass position and  $c_1$  is navigational constant. Navigational constant is tuned experimentally to allow the drone formation to attract to the ground nodes center of mass and not to be weaker than attraction/repulsion force. The attraction/repulsion force depends on the desired distance and is defined as:

$$\mathbf{F}_{att/rep_i}^{va}(k) = \sum_{j \in I_i} \Phi(\|\mathbf{p}_j^{va}(k) - \mathbf{p}_i^{va}(k)\| - d(k)) \delta\left(\frac{\|\mathbf{p}_j^{va}(k) - \mathbf{p}_i^{va}(k)\|}{1.2 * d(k)}\right) \mathbf{n}_{ij} \quad (4.16)$$

In addition to these rules, paired agent adds attraction force towards their ground nodes pairs, given by:

$$\mathbf{F}_{attr/gn_i}^{va}(k) = -c_{gn} f_\sigma(\mathbf{p}_i^{va}(k) - \mathbf{p}_j^{gn}) \quad (4.17)$$

Then, the total force acting on the paired virtual agent is defined as:

$$\mathbf{F}_i^{va} = \mathbf{F}_{att/rep_i}^{va}(k) + \mathbf{F}_{align_i}^{va} + \mathbf{F}_{nav_i}^{va}(k) + \mathbf{F}_{attr/gn_i}^{va}(k) \quad (4.18)$$

To fly in a coherent lattice while adapting inter agent distance, each robot has to have at least two neighbors in the interaction neighborhood. In this way drones keep triangular formations with agents in their interaction neighborhood. Therefore, drones run the neighborhood selection algorithm presented in Algorithm 1.

**Algorithm 1** Neighborhood selection

---

```

while  $size(messageBuffer) \neq 0$  do
   $j := message.ID;$ 
  if  $\|p_j^{va} - p_i^{va}\| < k_{int}d_{ij}$  then
    Add virtual agent to  $I_i^m$ 
  else
    Remove virtual agent from  $I_i^m$ 
  end if
  if  $C(I_i^m) = 0$  then
    Sort all  $va \in C_i^{mm}$  by distance
    Add two closest  $va$  to  $I_i^m$ 
  else
    if  $C(I_i^m) = 0$  then
      Sort all  $va \in C_i^{mm}$  by distance
      Add the closest  $va$  to  $I_i^m$ 
    end if
  end if
end while

```

---

**4.3.2 Results**

In simulation experiments we examine how drones adapt their formation to the motion of ground nodes. Experiments are run in EMANE emulator described in Appendix A. Ground node's speed  $v_{gn}$  is set to  $1.5m/s$ , which is an average human walking speed [35, 44]. Starting point coordinate is set to  $(0,0)$  in each experiment. Ground nodes and drones positions are randomly initialized from uniform distribution within radius of 30m from the starting point. Each drone uses UDP communication protocol, which implies that agents do not get the acknowledgment whether the messages are received by their neighbors. Routing is not implemented and agents broadcast their messages to their single-hop neighbors. Drones broadcast 10 messages per second. At the beginning of the experiment, all agents create complete communication graph and their clocks are synchronized. We run 10 emulator runs per experiment.

There is a large number of possible ground nodes' motion patterns. We choose to examine the performance of the algorithm on two cases of ground nodes' motion. Firstly, it is examined how does aerial team follow ground nodes moving in the same direction. Each ground node's motion is given by:

$$\begin{aligned}
 v_x^{gn} &= v^{gn} \cos(\phi) \\
 v_y^{gn} &= v^{gn} \sin(\phi) \\
 \phi &= \phi_0 + U(-1, 1) \cdot \phi_{noise}
 \end{aligned} \tag{4.19}$$

## Chapter 4. Connectivity maintenance

where  $v^{gn} = 1.5m/s$  is ground node's speed,  $\phi$  is ground node's orientation and  $\phi_{noise} = 10\pi/180$  is maximum absolute value of uniform noise added to ground node's orientation. Secondly, it is examined how do drones adapt their shape if ground nodes move in separate directions. Each ground node's motion is given by:

$$\begin{aligned} v_x^{gn} &= v^{gn} \cos(\phi) \\ v_y^{gn} &= v^{gn} \sin(\phi) \\ \phi &= \frac{2\pi}{n_{gn}}(n_{gn} - ID_{gn}) + U(-1, 1) \cdot \phi_{noise} \end{aligned} \quad (4.20)$$

where  $n_{gn}$  is number of ground nodes,  $ID_{gn}$  is ground node's id  $v^{gn} = 1.5m/s$  is ground node's speed,  $\phi$  is ground node's orientation and  $\phi_{noise} = 10\pi/180$  is maximum absolute value of uniform noise added to ground node's orientation.

A performance measure is the distance between the center of mass of ground team and the center of mass of aerial team. The center of mass of the aerial team is defined as a center of mass of the virtual agents. In Figures 4.1 and 4.2 we show results from the experiment in which drones translate their formation with the ground nodes. In the Figure 4.1 we show the center of mass distance and it can be observed that drones' formation follows the ground nodes with the average error of less than 10m. Besides measuring the distance between centers of mass, it

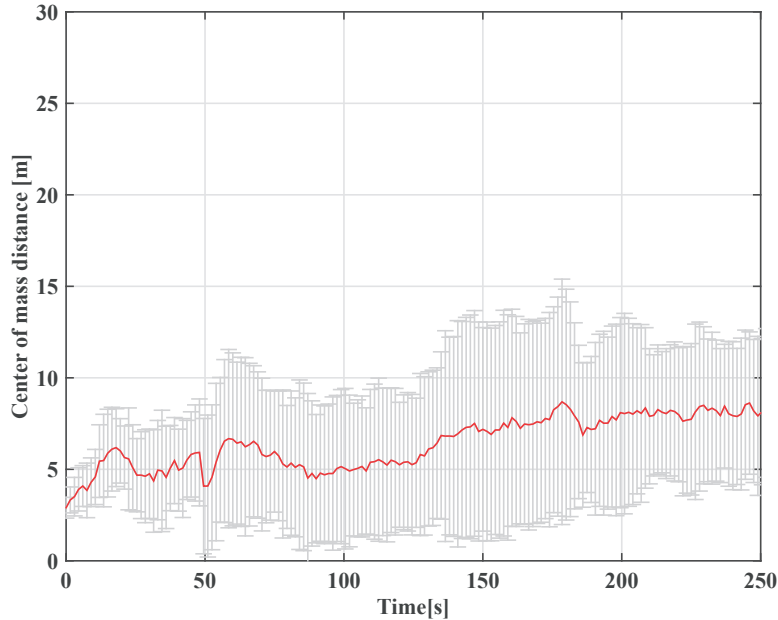


Figure 4.1: Distance between ground node's center of mass and virtual agent's center of mass. The experiment is run in Emane emulator, 10 runs. Drones formation follows the ground agents with average of 10m distance between centers of mass.

is examined if drones are flying in a coherent formation while they follow ground nodes. The Figure 4.2 shows results of a single emulation experiment. Tracks of drones, virtual agents and ground agents and all agents' positions in three time instances are shown. It can be observed that virtual agents keep the formation coherent and follow by keeping equilateral triangular formation. Drones are successfully following virtual agents and their phases are synchronized. Hence, drones fly in the same formation as their respective virtual agents. We examine the

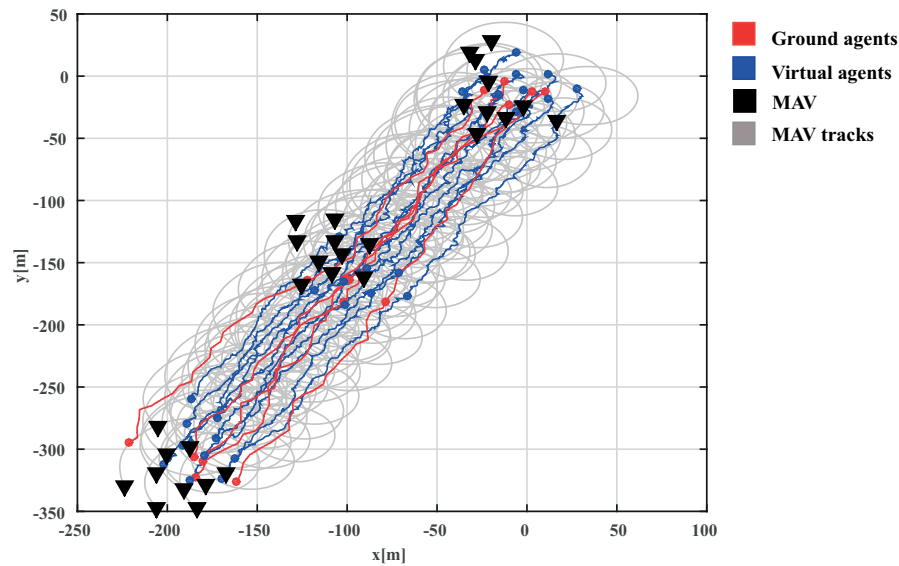


Figure 4.2: Tracks of virtual agents, ground nodes and drones in a single emulation experiment. Agents positions and drone positions are shown in three experiment time instances. Virtual agents closely follow group of ground nodes, drones synchronize their phase and create the same formation as their respective virtual agents.

same scenario in a field experiment. The field experiment setup consists of two humans carrying ground nodes and four fixed wing eBee platforms following them. The experimental setup is shown in the Figure 4.3.

Ground nodes consist of GPS sensor and WiFi module connected to a custom made extension board. Extension board is needed to access serial ports of Gumstix Linux board. The peripheral devices are connected with Linux Gumstix board. The communication algorithm runs on the Gumstix board, and broadcasts communication messages to aerial team. Ebee fixed wing drones are equipped with the same hardware as in the field experiments in Chapter 3.

The experiments starts by launching 4 eBee drones. After eBees settle in the equilateral triangular lattice and synchronize their phases, ground nodes start to broadcast their position and start to move in the same direction. Figure 4.4 shows the distance between centers of mass. The distance measure in field experiments is well aligned with the result obtained in

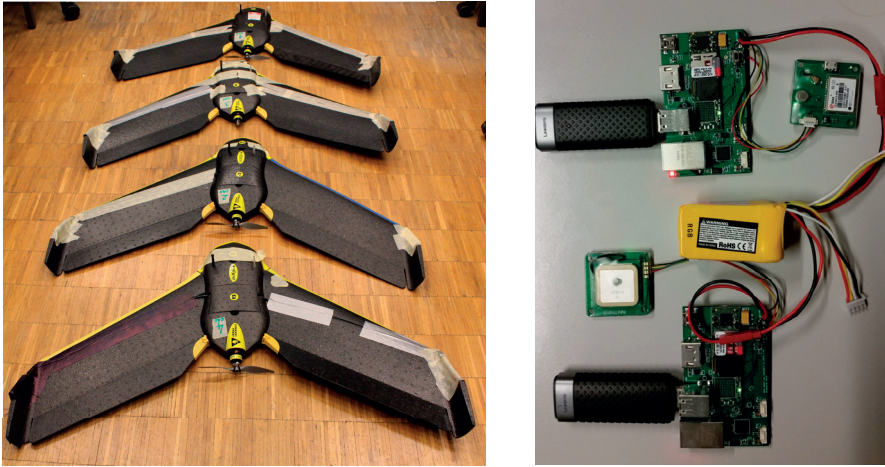


Figure 4.3: Tracks of virtual agents, ground nodes and drones in a single emulation experiment. Agents positions and drone positions are shown in three experiment time instances. Virtual agents closely follow group of ground nodes, drones synchronize their phase and create the same formation as their respective virtual agents.

the emulations. The drones, ground nodes and virtual agents tracks are shown in Figure 4.5. The field experiment results align well with the emulation experiments. The tracks show that virtual agents follow ground users in the equilateral triangular formation and that drones are following virtual agents in the same formation and synchronize their phases.

Secondly, it is examined how does aerial team keeps ground users connected if they walk in separate directions. We run each emulation experiment until the connectivity of the ground-aerial communication network is violated. Algebraic connectivity is a performance measure of how well the network is connected and it is calculated as second smallest eigenvalue of the Laplacian matrix.

Figure 4.6 shows algebraic connectivity of the ground-aerial team in relation to maximum distance between any two ground nodes. It can be observed that as ground nodes are moving in the separate directions, maximum distance between ground nodes is increasing and connectivity of the network is decreasing. If the algebraic connectivity is equal to the number of agents in the connectivity graph, it means that the connectivity graph is complete, which means that every agent has direct communication link with other agents in the network. The connectivity in the network is maintained up to 750m with 10 drones and 5 ground nodes on average which is three times higher than the single agent's communication radius. Figure 4.7 shows consecutive snapshots of the virtual agents and ground nodes formation connected in a connectivity graph  $\mathcal{G}_{conn}$ . As ground nodes are moving apart, the number of the connectivity graph edges decreases and the edges elongate. Virtual agents are symmetrically adapting the

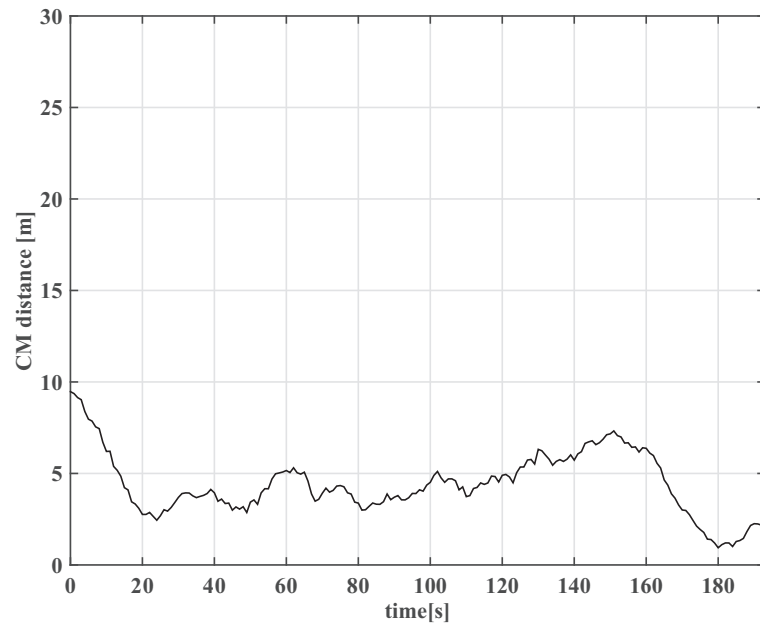


Figure 4.4: Distance between ground node's center of mass and virtual agents center of mass in field experiment. The experiment is run with four drones and two ground nodes. The drone formation follows the ground agent formation with less than 10m distance between centers of mass.

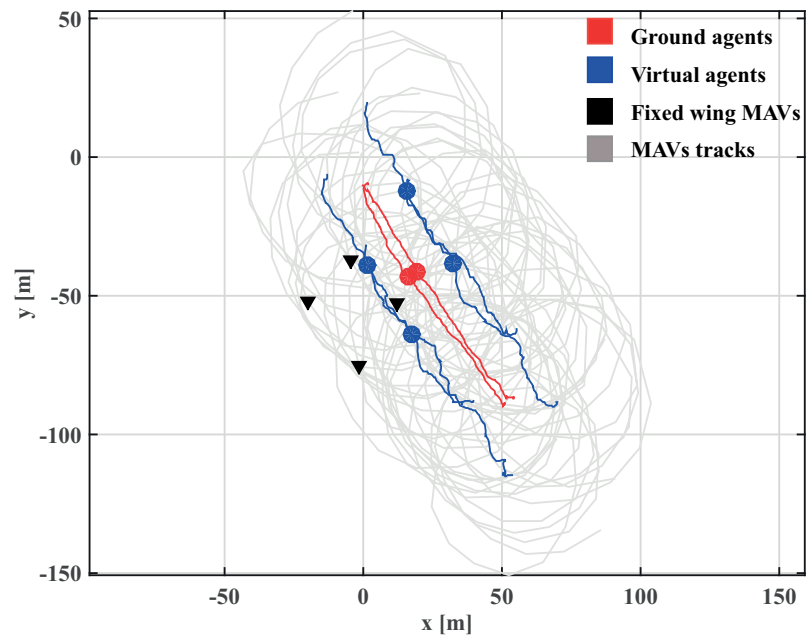


Figure 4.5: Tracks of virtual agents, ground nodes and drones in a field experiments. The field experiment results show that drones can successfully follow a group of ground nodes and translate the formation.

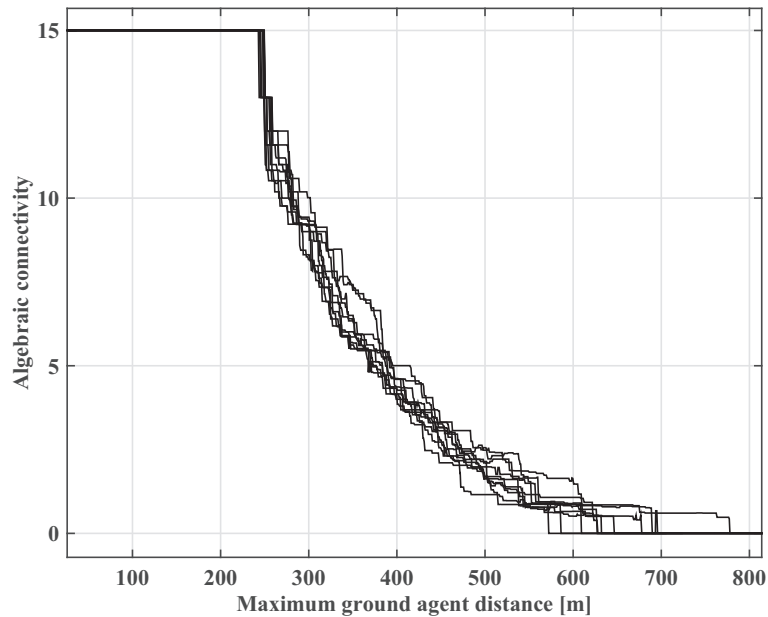


Figure 4.6: Algebraic connectivity of ground-aerial connectivity graph in relation to the maximum distance between any pair of ground nodes. As ground nodes are spreading, they move apart, interaction links are elongating. Connectivity is maintained up to 750m which is three times higher distance than communication radius.

equilateral triangular formation to the position of the ground nodes. In this way robots keep the equal quality of the communication links, keep the spatial arrangement and uniformly cover the area. In Fig 4.7d, one ground agent is connected with a single edge to the rest of the network. Further motion of this node will violate the connectivity despite that other edges are not fully elongated up to the sensing range.

In the field experiments, it is examined how drones spread their formation if ground nodes move in separate directions. The size of the experimentation field is 150x150m. The experiment starts in the center of the experimentation field. Drones run the control algorithm with the same parameters as in the emulation experiment. Four fixed wing eBee platforms are launched, they settle in the equilateral triangular formation with inter agent distance of 10m. Two ground nodes move in the separate directions towards the testing field edges. Figure 4.8 shows consecutive snapshots of the field experiment. It can be observed that fixed wing drones successfully adapt the inter-agent distance and spread to the ground nodes motion. Drones synchronize their phases successfully and create equilateral triangular lattice. Virtual agents are closely following ground nodes and adapting the formation shape. Information on positions of all agents is successfully spread within the formation.

To examine performance with different numbers of ground nodes, we run additional set of



### 4.3. Potential field based method

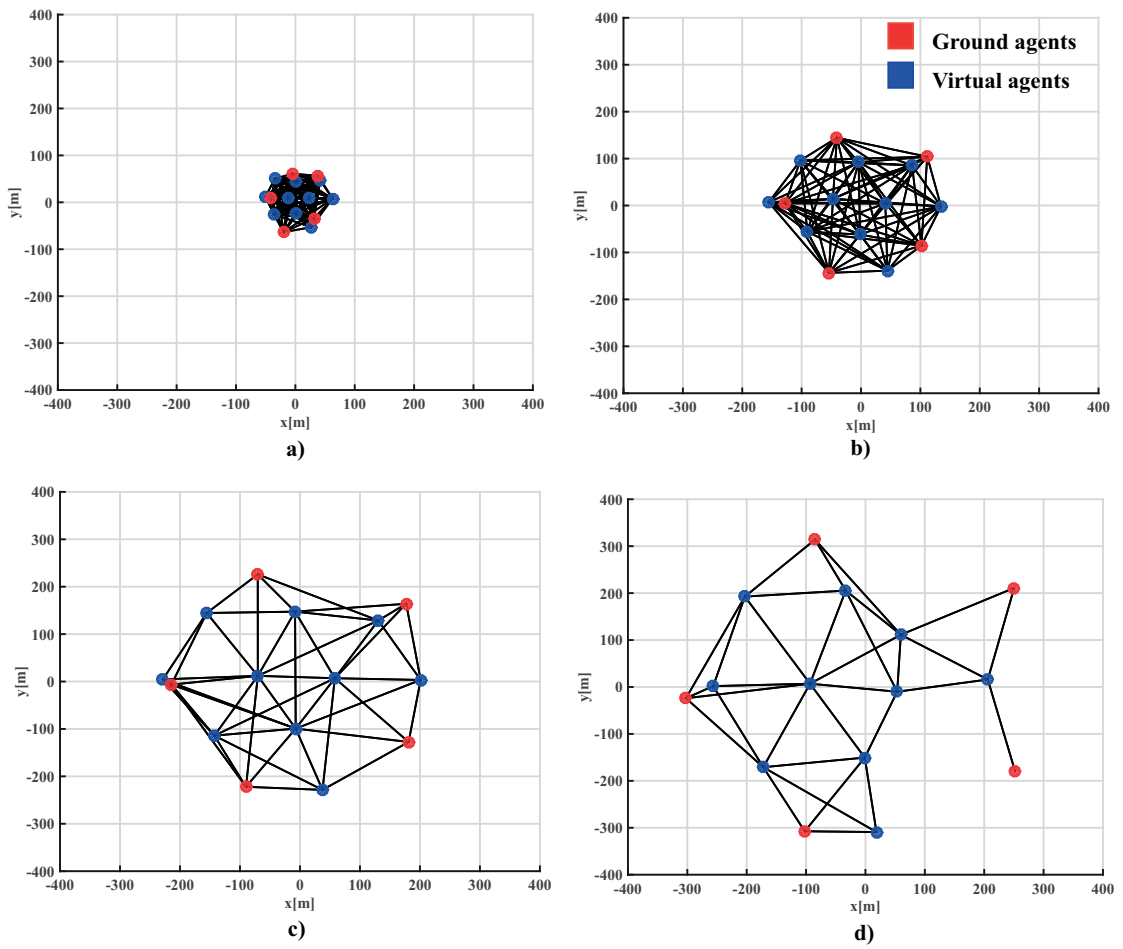


Figure 4.7: Consecutive snapshots of the topology of ground-aerial connectivity graph. Virtual agents are symmetrically adapting the inter-agent distances to spread the formation and adapt to the motion of ground nodes.

Chapter 4. Connectivity maintenance

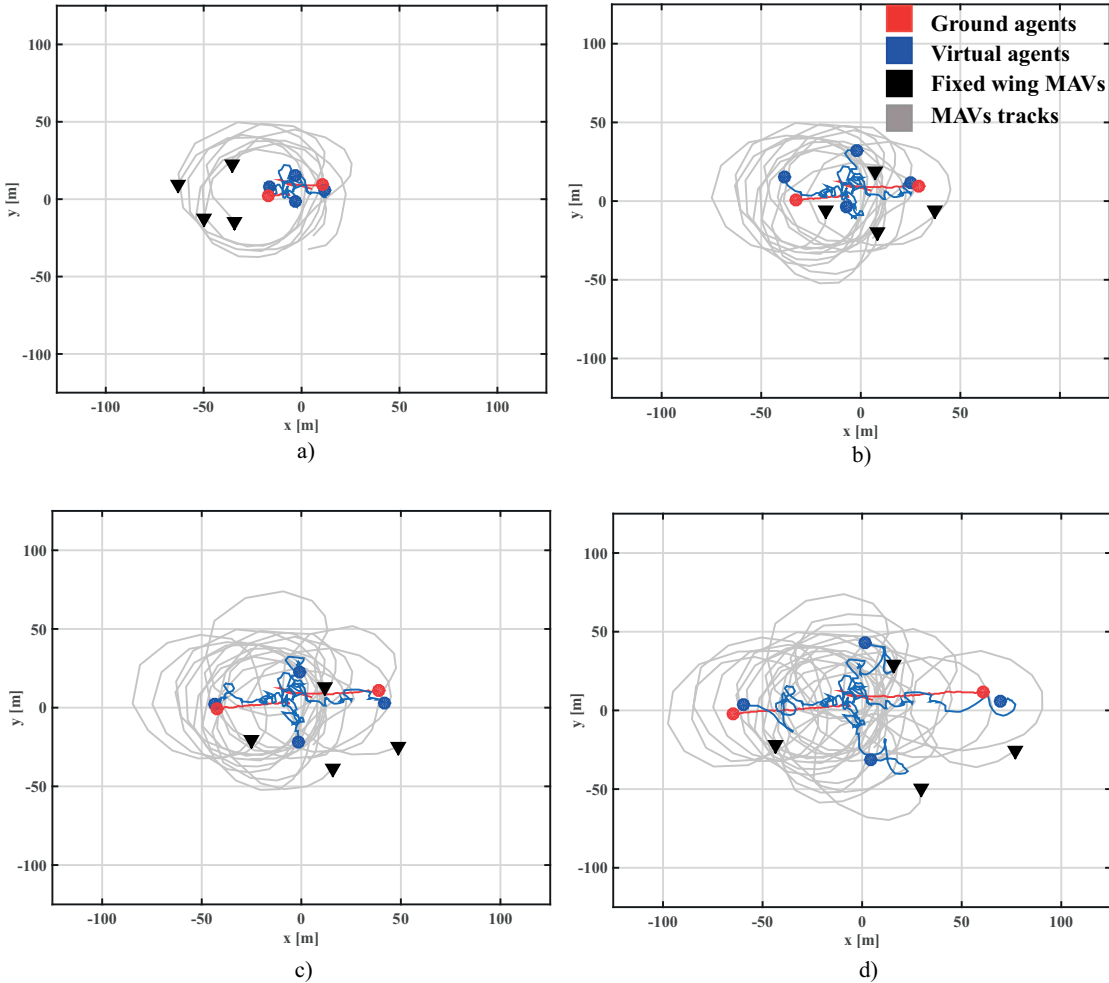


Figure 4.8: Field experiment with four drones and two ground nodes. Ground nodes move apart to the edges of the testing field. Swarm of drones receives their positions and increases inter-agent distance to spread the formation. Drones successfully synchronize their phases and create equilateral triangular formation.

emulation experiments. The drones number is kept constant to 10 drones. The number of ground nodes is altered from 2 to 6 ground nodes. In Fig 4.9 it can be observed that performance of the algorithm is almost invariant to the changing number of ground nodes. It mostly depends on the number of drones involved. While stretching the network, drones try to keep the equilateral triangular lattice and in that way they keep cycles in the interaction graph. If cycles in the interaction graph could be broken and interaction graph could be expanded to a tree graph, the coverage would be higher. This led us to develop the second communication maintenance method which we present in following subsection.

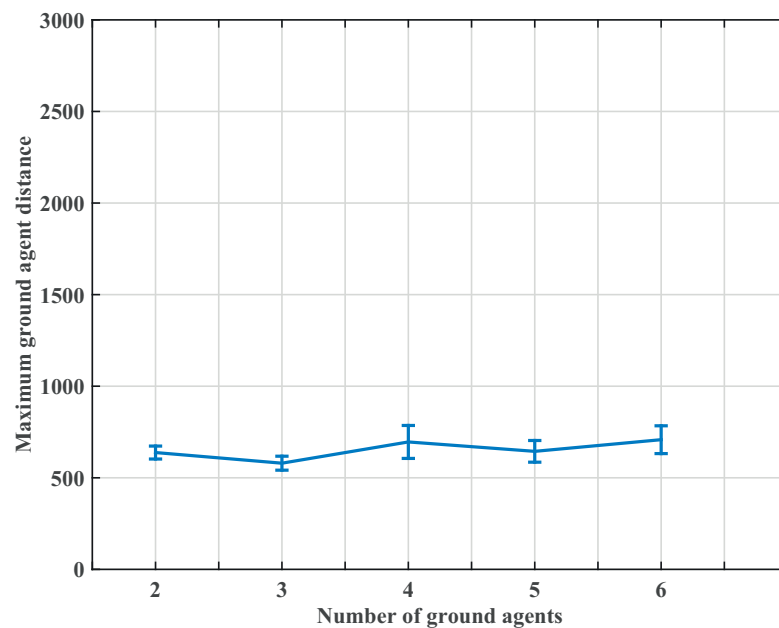


Figure 4.9: Maximum distance between any pair of ground nodes for different number of ground agents. The drone number is fixed to 10. We show that performance of algorithm is invariant on the number of ground agents, since links are expanding symmetrically.

#### 4.4 Distributed market-based method

In the potential field based method, drones expand and translate equilateral triangular lattice formation to adapt to the ground nodes' motion. The interaction graph of such formation consists of multiple triangles that create cycles within the graph. If some of the links in the interaction graph are broken, such structures can be extended to tree graphs. Transforming the interaction graph to a tree graph can significantly increase the maximum distance that the aerial agents can cover.

#### 4.4.1 Proposed method

We propose a method that extends the potential field based method with link deletions and heuristic rules that transform the equilateral triangular lattice to a tree graph. In addition, links deletions are performed such that the ground nodes are the leaves of the tree interaction graph.

*A ground-aerial connectivity graph  $\mathcal{G}_{conn}$  and ground-aerial interaction graph  $\mathcal{G}_{int}$  consist of two types of vertices, ground nodes  $v_{ga}$  and virtual agents  $v_{va}$ . We need to define a discrete control input  $u_d$  that adds and deletes links of the  $\mathcal{G}_{int}$  and a continuous control input  $u_i$  which controls drones motion such that resulting interaction graph is a tree  $\mathcal{G}_{tree}$  of the ground-aerial interaction graph and all leaves of  $\mathcal{G}_{tree}$  are ground nodes.*

The link deletion in the interaction graph is a nontrivial task, since interaction link breakages can lead to the formation fragmentation. Only virtual agents can break the interaction links since ground nodes are only passive nodes that do not directly influence the control of the aerial nodes. Virtual agents need an information about the topology of the interaction graph to know whether certain link deletion will violate the connectivity of the interaction graph. We modify the approach proposed by [90] based on market-based negotiations to delete interaction links.

Each virtual agent  $i$  receives message  $MSG_{auction}$  defined as:

$$MSG_{auction} = \{A_j, B_j, T_j, \mathbf{p}_j^{va}\} \quad (4.21)$$

which consists of a current interaction graph topology estimate  $A_j$ , a received bids list  $B_j$  and a received token vector  $T_j$ ,  $VA_j$  position  $\mathbf{p}_{va_j}$ . Then  $VA_i$  merges its topology matrix  $T_i$  with the received topology matrix  $T_j$ . If agent  $j$  is not in the interaction neighborhood and it satisfies the condition to enter the interaction neighborhood, link between agent  $i$  and agent  $j$  is added to the topology matrix  $T_i$ .

Concurrently with the topology update automaton, agents are running auction automaton. When the auction starts,  $VA_i$  calculates a safe neighbors set  $S_i$  defined as:

$$S_{drone_i} = \{va_j | \lambda_2(\mathcal{E}(\mathcal{G}_{int}) / v_i v_j) > 0\} \quad (4.22)$$

Among the safe neighbors, agent  $VA_i$  chooses a link to delete and places a bid based on the

equation:

$$[B_i]_i = \|\mathbf{p}_i^{va} - \mathbf{p}_j^{va}\| + (\|\mathcal{G}_{int}\|^2 \cdot c) \quad (4.23)$$

Virtual agent  $VA_i$  initializes token vector  $T_i = [0 \ 0 \ \dots \ 1 \ \dots \ 0 \ 0]$ , sets its bid vector  $B_i$  and enters the update phase of the auction automaton. A bid value depends on the distance between virtual agent  $VA_i$  and its safe neighbor and a number of neighbors of virtual agent  $VA_i$ . Therefore, agents who have more neighbors and longer links with safe neighbours will have higher bids. In the update phase, agent receives messages from other agents, updates bid list and token list. When agent receives messages from all agents in the network, and all tokens are received, agent decides on the maximum bid from its bid list and deletes that link from the topology matrix.

If agents apply this link deletion algorithm, the resulting ground aerial interaction graph will be a tree graph but leaves of the graph can be both ground nodes and aerial agents. To achieve a tree graph with ground nodes as leaves, we allow aerial agents to be in two states, a chain state and a swarm state. If agent has more than  $n_{toggle}$  neighbors it is in the chain state and if it has more than  $n_{toggle}$  neighbors, agent is in the swarm state. The value of  $n_{toggle}$  is determined empirically and it is set to 5. This number depends on the total number of flying agents in the interaction graph and it is set experimentally to the  $n_{toggle} = n_{drones}/2$ . If agent is in the swarm state its continuous control law is given by:

$$\mathbf{u}_i = \mathbf{F}_{att/rep_i}^{va} + \mathbf{F}_{align_i}^{va} + \mathbf{F}_{nav_i}^{va} \quad (4.24)$$

If agent is in the chain state it moves towards the furthest neighbor, and velocity of the virtual agent is directly controlled by:

$$\mathbf{u}_i = \mathbf{v}_{chain} = \langle v_i, \theta_i \rangle, \theta_i = \arctan(\mathbf{p}_{furthest} - \mathbf{p}_i^{va}) \quad (4.25)$$

This behavior allows drones which are close to the ground nodes to move toward them as they get further away. This will encourage the formation to stretch as stresses are imposed on the formation by moving ground nodes.

#### 4.4.2 Results

In simulation experiments we examine how virtual agents adapt their formation if ground nodes spread radially from the starting point. The starting point coordinate is set to (0,0).

## Chapter 4. Connectivity maintenance

---

Ground nodes perform motion patterns given by Eq. 4.20. Experiments are run in Netlogo simulator described in Appendix A. In each time step of the simulation, agents are synchronously updating their communication and interaction neighborhoods. The information about the topology of the communication and interaction graph spreads uniformly within one simulation step. Ground nodes and virtual agents positions are randomly initialized from uniform distribution within radius of 30m from the starting point.

Figure 4.10 shows algebraic connectivity of the ground-aerial team in relation to the maximum distance between two ground nodes. As maximum distance between ground nodes is increasing, the algebraic connectivity is decreasing because the number of link in the connectivity graph is decreasing. The dynamics of the decrease is similar as for the potential field method based approach (Figure 4.7). In the first iterations of the formation control, the interaction graph becomes a spanning tree and it spreads as ground nodes are spreading. Up to the communication graph elongation of 600m both potential field based approach and market based approach have similar dynamics. It can be observed that using market-based approach, drones can cover larger distances between ground nodes than in the potential field based approach. The distance covered is around 1200m which is significantly larger than 750m distance for potential field based method.

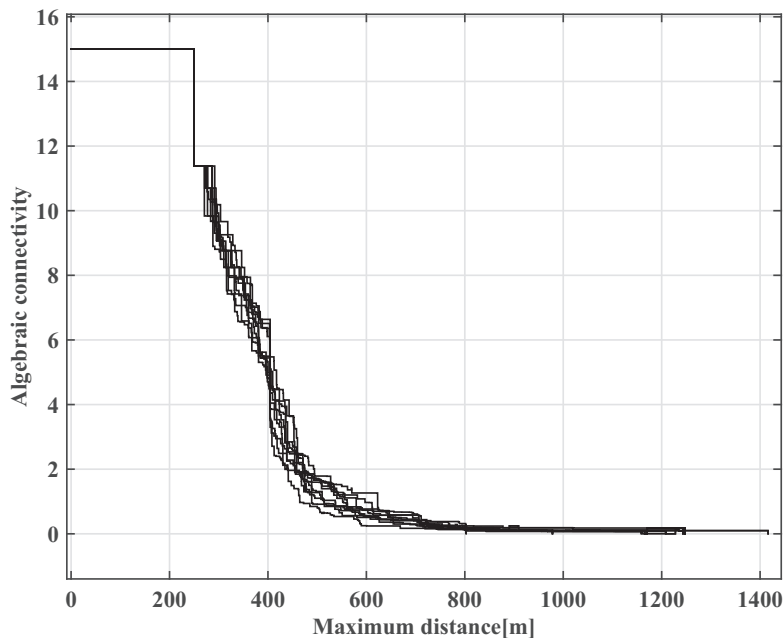


Figure 4.10: Algebraic connectivity for the auction based method. The results are shown for 10 simulation experiments, with 10 drones and 5 ground agents. By changing topology of the interaction graph, drones can extend the aerial network and cover larger distances between ground nodes.

Figure 4.11 shows consecutive snapshots of the connectivity maintenance experiment running auction based method. As ground nodes are spreading, the number of the connectivity graph edges decreases and the edges elongate. Virtual agents organize in the spanning tree and enter the chain state. In this state they try to move towards their furthest neighbor. This causes reactive links elongation, without prior knowledge of the positions of all other agents in the interaction graph. As a result of this behavior, all agents are keeping equal distances between their neighbors. Such configuration of the interaction graph will also allow that ground nodes spread more than in the potential field approach.

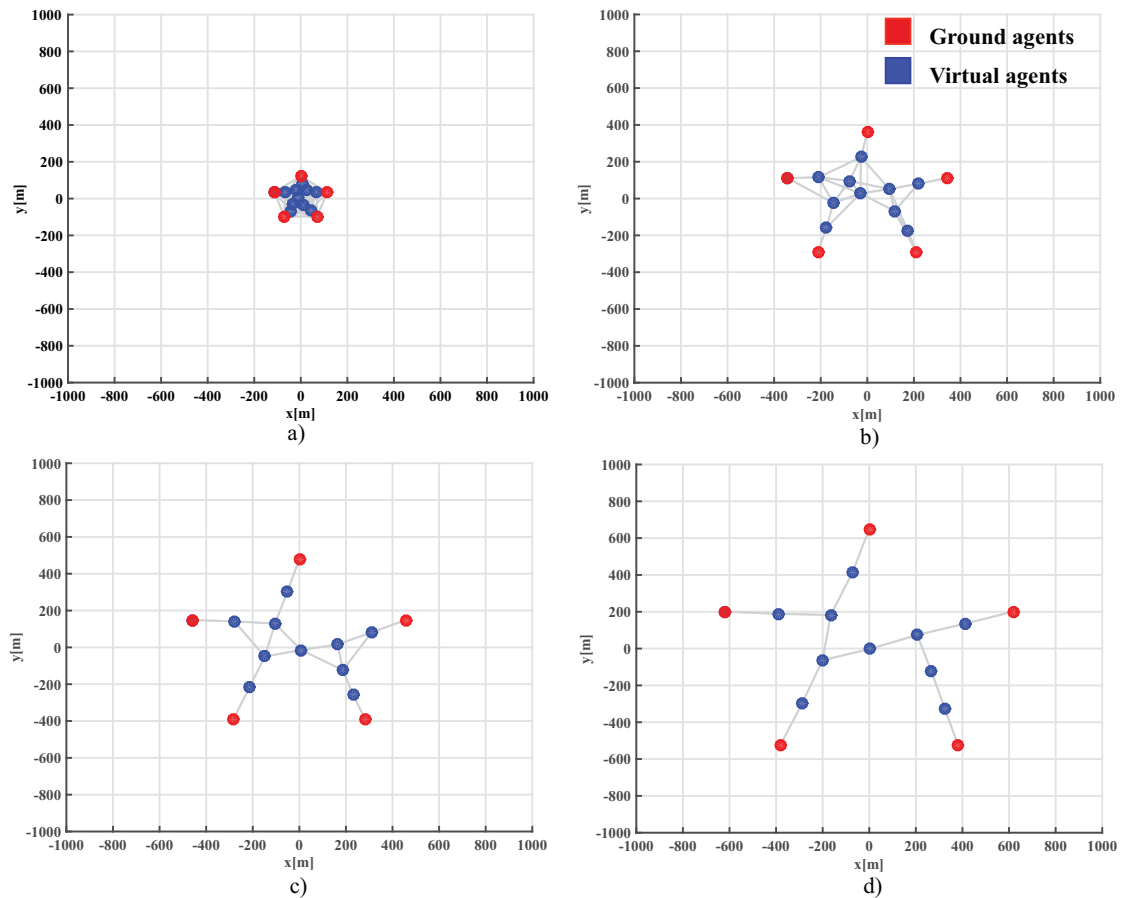


Figure 4.11: Consecutive snapshots of the simulation experiments for auction based method. As ground nodes are moving apart, the interaction topology is transforming to a spanning tree, and links elongate to stretch the network. Such interaction topology allows higher distances between ground nodes than the equilateral triangular lattice topology.

To examine how the performance of the algorithm changes with the number of ground nodes we run multiple experiment in which the number of ground agents is changed from 2 to 6 nodes and number of flying agents is kept constant at 10 agents. Figure 4.12 shows comparison of potential field based method and auction based method for communication maintenance.

## Chapter 4. Connectivity maintenance

On the x-axis we show number of agents in each experiment and on y-axis we show maximum distance between any two ground nodes. Each experiment is performed 10 times. It can be observed that auction based method performs significantly better than potential field based method for teams with 2 and 3 ground agents. For the higher number of agents, the auction method still performs better than potential field based method but the performance is decreasing.

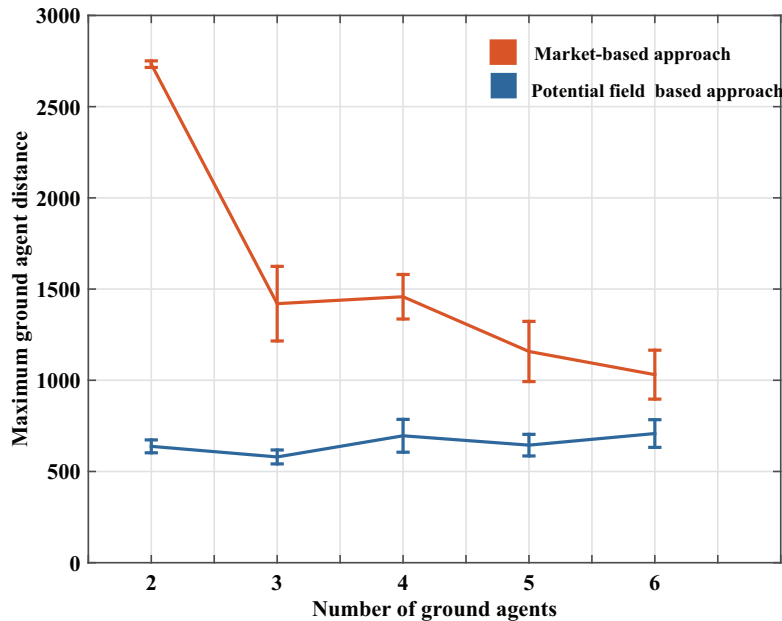


Figure 4.12: Comparison between potential field based method and auction based method.

We run additional emulation experiment to observe how does the market based approach perform if communication is not synchronous and if network topology updates are not synchronous among all agents. We ran experiment in EMANE emulator with 4 flying agents and two ground agents to observe whether flying team will extend to a line if ground nodes are far apart. In Figure we show consecutive snapshots of emulation experiment with 4 drones and two ground nodes.

We can observe that for 4 drones and two ground agents, the drone formation extends to the chain and interaction links are deleted between drones. When we increase number of drones in the network, we observed in multiple emulations that due to the asynchronous update of the topology matrices, drones entered the state in which some drones did not have any link to delete and did not place a bid while other drones had different topology matrix and could place a bid. This led to the auction deadlock and interaction links could not be further deleted. One possible solution to this problem would be to combine auction process with the



#### 4.4. Distributed market-based method

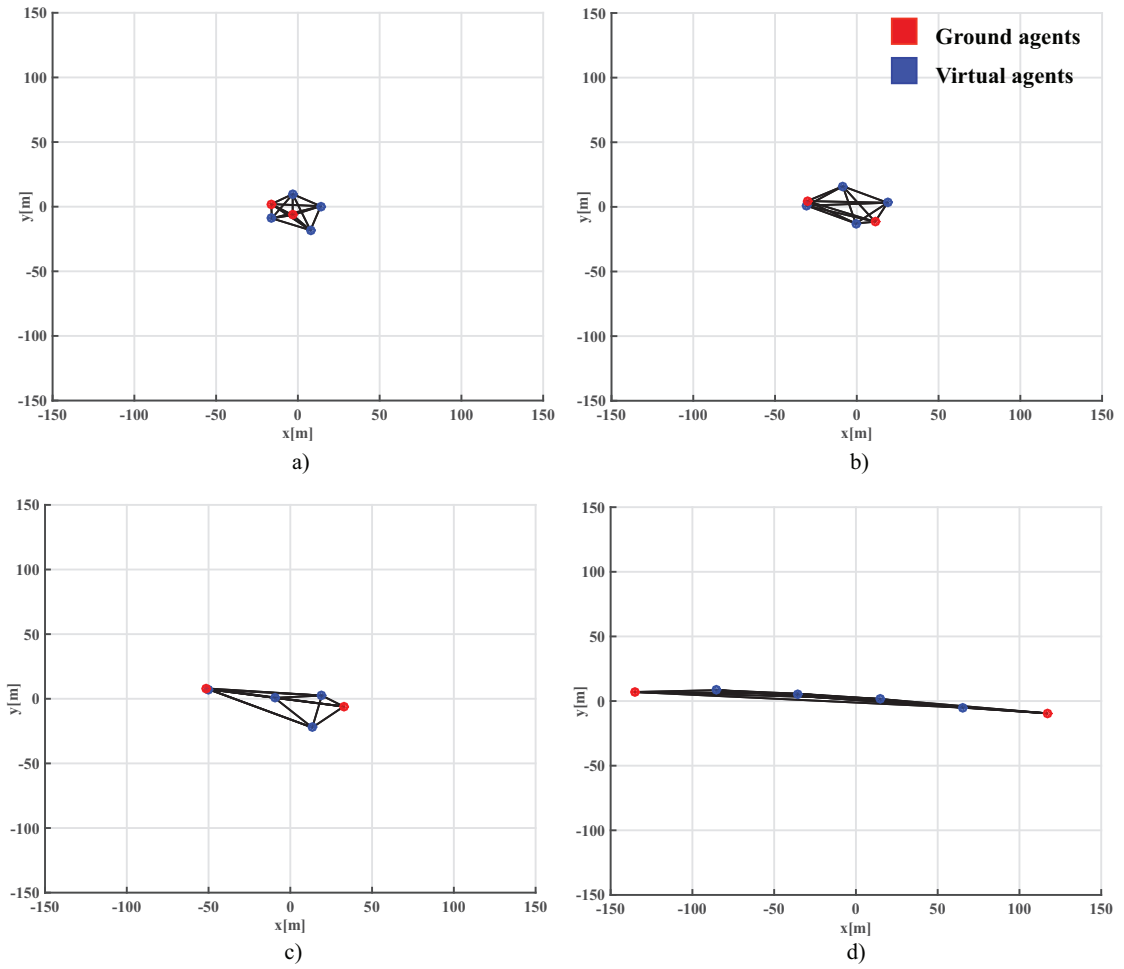


Figure 4.13: Emulation experiment for auction based method with 4 drones and 2 ground nodes. We show 4 consecutive snapshots a) 3s b) 66s c) 83s and d) 133s. In the first two snapshots, drones are running potential field based method and stretching links of equilateral triangular lattice. In the third snapshot, the auction starts, interaction links are deleted and drone formation extends to a chain. In fourth snapshot, drones are in chain state, they are moving towards furthest neighbour and gradually extending communication links.

agreement on the topology. Agents should not place bids in current auction before they all agree on the topology matrices.

### 4.5 Extensions

Potential field based method is invariant to growing number of ground nodes, its performance depends on number of flying robots engaged in the mission. Furthermore, this method does not require robots to negotiate to delete links, hence it is less demanding on the communication channel. Theoretically it can work also for larger groups of ground agents, larger than number of flying agents because it depends on the center of mass of the ground node formation and maximum distance between center of mass and furthest ground node. Since this method does not have the notion on the interaction topology, flying agents do not know which links could be deleted and when to potentially spread the network beyond increasing links in the equilateral triangular lattice.

Therefore we explore another method that allows flying robots to delete interaction links and change the interaction topology to a spanning tree in which tree leaves are ground agents. This method requires that each agent exchanges its neighborhood list with others and calculate the overall topology of the network. Then agents negotiate over the network using auction based approach to delete links that could extend the formation and will not violate the connectivity of the interaction graph and cause fragmentation of the team. Such method performs much better than potential field based method, in the case of small number of ground agents compared to the number of flying nodes. There is also a theoretical number of relation between flying nodes and ground robots for which this method can work. Beyond this number it is not possible to create a tree with ground nodes as tree leaves.

There are two problems that still need to be solved and will be part of the future work on this topic. Firstly, asynchronous updates of the topology should be addressed to prevent deadlocks in the link deletion. Before agents place bids they should check whether they all participate in the same auction and whether they all see the same topology.

These two methods can be merged to solve the scenario in which large group of ground agents are moving in the field. We additionally analyze the ground node positions to determine whether ground nodes organize in sub clusters. For that we run k-means algorithm and we measure distance between each center. This distance can trigger switching between potential field based method and market-based method.

## **4.6 Conclusion**

A solution to the problem of connectivity maintenance in ground-aerial team was presented. A team of drones is adapting the topology of its interaction network to independent moving ground nodes. Two methods were proposed, potential field based method and auction based method. The potential field based method relies on the data collected from single hop neighbors. Each agent updates its neighbors about the knowledge on his neighborhood and his position. After few communication cycles, agents have a rough estimate of the position of all agents in the network. This position map is then used to calculate the distance flying robots should maintain in the equilateral triangular lattice. The interaction topology is changing in the continuous domain, and we only change the distance agents keep with their neighbors but drones still try to maintain equilateral triangular lattice. Such approach is good for maintaining communication and covering the area at the same time with on-board cameras, it is not sensitive to delays and asynchronous communication and works for larger groups of ground agents. The second method, auction based method uses market based negotiation approach to delete interaction topology links and to allow the flying network to stretch more and allow longer multi-hop communication links between ground nodes. This approach relies on the fast communication between agents, adds an additional information that should be shared in the communication channel, it is not prone to the changing number of robots, and works well for smaller number of ground nodes. The best solution would be a hybrid approach which would consist of potential field based approach for smaller distances between agents, detection step using data mining techniques such as k-means clustering which would detect cluster of agents and their distances, and if distance between clusters is over the threshold, agents start to delete some links to spread the network.



## **5 Conclusion**



## **5.1 Main accomplishments**

Networked drone teams can be used for search and rescue, surveillance or to establish dedicated wireless networks in urban environments. The main advantage of such drone teams is that they can rapidly deploy, fly above the obstacles and obstructed terrains and provide line-of-sight communication. In such teams, drones create wireless ad-hoc networks and adapt their position to the ground users motion and to the desired link quality specifications. The main objective of this thesis has been to develop networked fixed wing drone team that will adapt its spatial arrangement to keep ground users connected in a wireless ad-hoc network.

In drone missions, energy consumption plays an important role. Fixed wing drone autonomy is usually limited to 30-40 minutes of flight and energy-demanding behaviors can increase the energy consumption up to 20%. In this thesis, a new measure, energy expenditure, was introduced. Using this measure and multi-objective optimization, it was identified which strategy will track the target with a minimal distance and a minimal energy expenditure. Strategies are systematically compared in simulation and in field experiments. We showed that orbital trajectory motion provides the best results both in simulation and field experiments, and this strategy will be used to bridge nonholonomic constraints of fixed wing robots when they fly in teams.

Furthermore, a formation algorithm that allows fixed wing drones to regulate distance between their neighbors was developed. The algorithm is based on flocking rules, virtual agent following and fixed wing phase angles synchronization. Drones were able to adapt the distances to their neighbors both in simulation and field experiments where drones experienced asynchronous communication and wind disturbances. By applying this formation control algorithm, fixed wing drones were able to converge to an equilateral triangular lattice. In the use case, it has been shown that this method is well suited for communication link adaptation. The adaptation layer which controls the drones synchronization dynamics has been added and drones were able to adjust their distance to achieve desired communication link quality.

Since fixed wing drones should serve as communication relays for mobile ground users, a distributed connectivity maintenance methods was developed. Drone teams are adapting their topology to the independently moving ground users. Two methods were presented, potential field based method and distributed market-based approach. With potential based method, drones maintain connectivity by locally deciding about desired distance they should keep with their neighbors. Their decision is based on the the rough estimate of the ground agents and drones position. The second method extends the potential field based method

by deleting interaction links in the drone team to extend the formation and to cover larger distances. We show that this extension enables longer multi-hop communication links for ground users.

Strategies for communication maintenance and formation control are tested in field experiments. Having real world constraints, such as asynchronous communication, message delays, wind disturbances and unmodeled drone dynamics, the robustness of the algorithms was shown and results aligned well with the results obtained in simulation.

### 5.2 Potential applications

Adaptive drone networks will allow quick deployment of dedicated communication networks for ground users. Such networks could be used in many scenarios where communication links are broken or communication should be improved. The communication loss mostly occurs in disaster situations. Rescuers depend on reliable communication links to conduct successful and fast rescue operations. They usually operate in foreign countries and have no access to the cellular networks or the cellular network infrastructure is damaged. Such teams would benefit of their dedicated drone network which is easily portable, can be quickly deployed and adapts to their motion in the rescue field. In urban environments, communication infrastructure can be also damaged or can be of poor quality. Telecommunication operators could use such drone networks to replace some access points due to the infrastructure damages.

There is also a recent trend to provide Internet access to remote and unpopulated areas. In such areas, cellular network infrastructure usually does not exist. In Google Loon project [36] stratosphere balloons are used to allow Internet connectivity. Researchers in Facebook Connectivity lab [30] explore technologies to provide Internet access in remote areas with low cellular network coverage. The approach proposed in this thesis combined with decentralized algorithms for long duration missions could also be used for such applications.

Fixed wing drones can carry cameras and provide aerial overview of the terrain. If drones fly in teams in formations, the team camera coverage can be adapted to guarantee persistent coverage. Using such drone teams can guarantee larger camera coverage than single robot and hence can detect rare events. Constant event surveillance information can be streamed and directly communicated to the users. This can be used for monitoring concerts or rapidly changing urban environments.



## 5.3 Future directions

In spite of the recent improvements in drones automation, there are still no commercially available large scale drone systems on the market. With the drone industry expanding, it is assumed that the single drone cost will decrease and drone teams will become affordable and more present in future applications.

It is expected that drone teams stay in the air for longer period of time, sometimes even for days, to perform surveillance and provide connected communication network. The battery life of a single drone is still much lower compared to the mission duration. Therefore, it is important to develop effective scheduling strategies for drone teams, that will allow drone recharging without network quality disruption. This could be one promising research direction for large scale drone systems.

The drone network should be also less dependent on the gps signal sent from ground users. The drones in the network should be able to track the ground users with their cameras or use ground user wireless signals to track their position. Tracking the wireless signal with single drone has been proposed by [37, 18] and visual tracking has been proposed by [84]. Drone teams should merge received sensing data to perform better ground users localization than with single drone. In addition, to improve the autonomy of drone teams, drones wind compensation and aerodynamic properties should be improved to allow them to fly in rough weather conditions.

The interaction between drone network and human users should not put high cognitive load on human users since they should be able to concentrate on their primary tasks (for example, search for the victims in search and rescue operations) [4, 59, 76]. Therefore, one promising research direction would be to develop intuitive, gesture based control and monitoring of the drone swarm.

One further development of collective drone systems could go in the line of integrating distributed drone data acquisition with the cloud computing. Fixed wing drone teams which distributively fly in a formation can upload their collected data to the cloud and partially coordinate their actions. In such cases, cloud could serve as mission manager and provide high level tasks to the swarm. They could connect to the cloud and upload their data on the cloud and cloud mission manager could determine the next task for the drones based on the received data and drone status.



# A Appendix - Materials and experimental setup

## A.1 Matlab and Netlogo simulator

We examine the performance of algorithms in multi-agent simulators built in Matlab and Netlogo simulation environment. In simulations, drones are modeled as discrete Dubin's vehicles:

$$\begin{aligned}x_i^{drone}[k] &= x_i^{drone}[k-1] + T_d v_i^{drone} \cos(\phi[k-1]) \\y_i^{drone}[k] &= y_i^{drone}[k-1] + T_d v_i^{drone} \sin(\phi[k-1]) \\ \phi_i[k] &= \phi_i[k-1] + T_d \omega_i^{drone}\end{aligned}\tag{A.1}$$

Virtual agents and ground nodes are simulated as discrete point masses. Communication range is defined as a circular field around the virtual agent position (for the simplicity, here we neglect drone's loitering radius).

In Matlab simulator, agents' dynamics is simulated in multi-agent Simulink tool AV-lib [57]. Agents exchange data synchronously, by accessing joint information matrix. In this matrix, information about agents' position and speed is stored and each agent synchronously accesses this information in discrete time steps  $T_d$ .

To simulate negotiation algorithm in market-based connectivity maintenance algorithm, we use Netlogo simulator. Matlab based simulator is effective if agents only exchange messages with few numerical value. In the case of market-based approach, agents exchange their neighborhood lists which becomes impractical in Matlab simulator. Therefore, we decided to simulate this algorithm using Netlogo numerical simulator [85]. Netlogo's programming language is a member of Lisp family [77] and it supports agents and concurrency. Mobile agents move over a grid of patches. All agents can interact with each other and perform tasks

concurrently. The numerical part of the simulator is similar to the Matlab environment, but in contrast to Matlab simulator, Netlogo has built in functions to send messages between agents and it handles connectivity links automatically based on the distance between agents.

### A.2 EMANE network emulator

Testing platforms for drone's algorithms should be as close as possible to the field experiments. Such advanced simulators allow thorough debugging before running field experiments. Frequently, hardware-in-the-loop emulators are used to test the robot control algorithms. In addition, there is a necessity to easily port the code from simulation environment to real hardware.

In the case of networked robots, besides hardware and physics simulations, communication network performance should be simulated. Communication network model is a layered structure in which each layer takes care of specific aspects of the networking and feeds its results to the subsequent layers [81].

Swarm emulator used in this thesis consists of three main modules, communication network emulator EMANE, linux board emulator and drone and ground agents' physics emulator. Such emulation setup will allow us to detect failures in algorithm logics, non stable behaviors due to communication delays, give us opportunity to analyze turn rate and forward speed signals and provide realistic message exchange between drones.

Network emulator EMANE implements mobile network systems in a real-time. It models link and physical network layer so that the software on higher network layers are subjected to the same setup as in the field experiments. Furthermore, each Gumstix microcomputer is represented in the emulation setup as virtual Linux container (LXC). Then these nodes are connected on the MAC layer and physical layer over EMANE emulator. On each virtual Gumstix node we run routing algorithm and UDP based message exchange between drones. Drone motion is emulated in real time and drone positions are sent to the EMANE emulator using Linux pipeline. The schematics of the emulation is given in Figure A.1.

### A.3 Ebee fixed wing robots

EBees fixed wing drones, produced by Sensefly, are used in the field experiments in this thesis. The flight autonomy of the Ebee drones is 45 minutes and they are powered with LiPo batteries. The total drone weight is 450g and the wingspan is 96cm. Drones are built from EPP foam

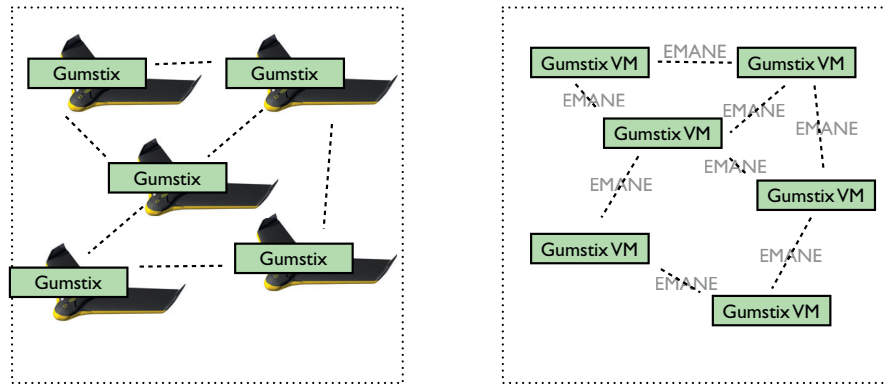


Figure A.1: Comparison of real hardware setup and emulation setup

with carbon structure and they have detachable wings, which makes them easily portable. Drone's altitude and turn rate is controlled by elevons, movable control surfaces that serve both as elevator and aileron control surface [52]. Thrust force is produced by an electric motor mounted at the end of the drone middle part. Drone's maximum speed is 14 m/s and it can resist wind speeds up to 10 m/s. This property makes them applicable in high wind conditions, which especially relevant for search and rescue operations. The electronics is mounted in the middle part of the robot and is completely enclosed in the foam, which makes this drone prone to humidity. The Ebee platform is shown in Figure A.2.



Figure A.2: Ebee platform, wings can be detached from the main body and it can be folded for easy transport.

A drone's control hardware consists of the autopilot, a Gumstix microcomputer and an extension board. A built-in autopilot, developed by senseFly, controls drone's attitude, airspeed and altitude. Gumstix microcomputer is connected to the extension board and communicates over serial link with the autopilot. An extension board, developed at LIS by Gergoire Heitz, enables the user to access serial ports of the Gumstix microcomputer. A dedicated API library, GAPI, was developed for this project to interface the autopilot and Gumstix microcomputer. This library enables safe serial transfer of control commands from Gumstix to the autopilot

## Appendix A. Appendix - Materials and experimental setup

---

and sensor data from the autopilot to the Gumstix microcomputer. The hardware schematics is shown in Figure A.3. Ebee drone is equipped with IMU unit and compass which provide



Figure A.3: Hardware setup consists of Gumstix Linux microcomputer and extension board which has an access to the autopilot data and Gumstix ports.

attitude information. The GPS sensor provides global position and ground speed of the drone. To measure wind speed and the drone's airspeed, Ebee is equipped with a pitot tube. At the bottom of the middle part, Ebee is equipped with an optic flow sensor. This sensor is used to achieve save and precise autonomous linear landing.

Drones establish ad-hoc wireless communication network using usb wireless dongles. The Linksys WiFi dongles are used and they operate in 5 GHz frequency range to prevent interferences with frequently used wireless networks in 2.4 GHz range. The dongles are set in 802.11n operation mode.

During the experiment, drone's states, position, battery level and speed is monitored on the base computer station. Drones connect to the eMotion software [3] over long range XBee communication and periodically send their information. This is crucial for safe operation, since single operator can quickly detect failures or non stable behaviors and send commands for drone's safe landing.

### A.4 Field experiments setup

Field experiments start by switching on the drones and connecting them to the base computer. Operator needs to connect to the Gumstix board of each drone using ssh protocol and examine whether drone is connected to the drone network. When drone receives GPS signal and runs safety checks it is ready to be launched. This procedure takes around 5 minutes. Drone's altitudes are separated by 10m to avoid collisions. Due to this, drones are launched in the sequence, starting from the highest drone.

Launching is performed by slightly shaking the drone which signalizes to the autopilot to switch on the engine. When the drone is ready to fly and the engine is on, the drone is thrown in the air. Drones fly to the initial home waypoint and wait to start the collective behavior. When all drones are launched, operator sends start signals to drones to start collective behavior and Gumstix microcomputer takes over the control by sending desired turn rate, altitude and airspeed to the drone's autopilot. When the experiment is over, operator sends stop signal to signalize the autopilot to take over the control and send drone to the initial home waypoint. Then the land signal is launched and emotion calculates schedule for collective landing.

To allow safe operations, each drone has its safety radius, which is defined from the launching spot. If drone hits the border of the safety radius, autopilot takes over the control and drone is immediately returned to the home waypoint. Drones also measure wind speed and if measured wind is higher than safety threshold, robots return to the home waypoint. Transitions between different flight modes and abrupt changes of the turn rate and speed are smoothed out at the autopilot level to prevent abrupt changes in the drone's attitude.





# Bibliography

- [1] Iperf. <https://iperf.fr/>, 2015. [Online; accessed 25-February-2015].
- [2] Sensefly eBee. <http://www.sensefly.com/drones/ebee.html>, 2015. [Online; accessed 23-February-2015].
- [3] Sensefly eMotion. <https://www.sensefly.com/drones/emotion.html>, 2015. [Online; accessed 23-February-2015].
- [4] Julie A Adams, Curtis M Humphrey, Michael A Goodrich, Joseph L Cooper, Bryan S Morse, Cameron Engh, and Nathan Rasmussen. Cognitive task analysis for developing unmanned aerial vehicle wilderness search support. *Journal of cognitive engineering and decision making*, 3(1):1–26, 2009.
- [5] Dominique Alejo, JA Cobano, G Heredia, and A Ollero. Optimal reciprocal collision avoidance with mobile and static obstacles for multi-uav systems. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 1259–1266. IEEE, 2014.
- [6] Brian DO Anderson, Barış Fidan, Changbin Yu, and Dirk Walle. Uav formation control: theory and application. In *Recent advances in learning and control*, pages 15–33. Springer, 2008.
- [7] R. Anderson and D. Milutinovic. A stochastic approach to dubins feedback control for target tracking. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3917–3922. IEEE, 2011.
- [8] Mahdi Asadpour, Domenico Gustiniano, Karin Anna Hummel, and Simon Heimlicher. Characterizing 802.11n aerial communication. In *to be submitted*, 2013.
- [9] S. S. Baek, H. Kwon, J. A. Yoder, and D. Pack. Optimal path planning of a target-following fixed-wing uav using sequential decision processes. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2955–2962. IEEE, 2013.

## Bibliography

---

- [10] Meysam Basiri, Felix Schill, Dario Floreano, and Pedro Lima. Audio-based relative positioning system for multiple micro air vehicle systems. In *Robotics: Science and Systems RSS2013*, number EPFL-CONF-191181, 2013.
- [11] Meysam Basiri, Felix Schill, Pedro U Lima, and Dario Floreano. Robust acoustic source localization of emergency signals from micro air vehicles. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4737–4742. IEEE, 2012.
- [12] R. W. Beard. A class of flight trajectories for tracking ground targets with micro air vehicles. In *Control & Automation, 2007. MED'07. Mediterranean Conference on*, pages 1–6. IEEE, 2007.
- [13] R.W. Beard, T.W. McLain, D.B. Nelson, D. Kingston, and D. Johanson. Decentralized cooperative aerial surveillance using fixed-wing miniature uavs. *Proceedings of the IEEE*, 94(7):1306–1324, 2006.
- [14] Eric Boivin, André Desbiens, and Eric Gagnon. Uav collision avoidance using cooperative predictive control. In *Control and Automation, 2008 16th Mediterranean Conference on*, pages 682–688. IEEE, 2008.
- [15] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Gossip algorithms: Design, analysis and applications. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1653–1664. IEEE, 2005.
- [16] Andreas Breitenmoser, Mac Schwager, J-C Metzger, Roland Siegwart, and Daniela Rus. Voronoi coverage of non-convex environments with a group of networked robots. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4982–4989. IEEE, 2010.
- [17] D. Brockmann, L. Hufnagel, and T. Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, 2006.
- [18] Mattia Carpin, Stefano Rosati, Mohammad Emtiyaz Khan, and Bixio Rimoldi. Uavs using bayesian optimization to locate wifi devices. *arXiv preprint arXiv:1510.03592*, 2015.
- [19] Scott H Clearwater. *Market-based control: A paradigm for distributed resource allocation*. World Scientific Publishing Company, 1996.
- [20] Parker Conroy, Daman Bareiss, Matt Beall, and Jur van den Berg. 3-d reciprocal collision avoidance on physical quadrotor helicopters with on-board sensing for relative positioning. *arXiv preprint arXiv:1411.3794*, 2014.

- [21] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *Robotics and Automation, IEEE Transactions on*, 20(2):243–255, 2004.
- [22] Randy Andres Cortez, Rafael Fierro, and John Wood. Connectivity maintenance of a heterogeneous sensor network. In *Distributed Autonomous Robotic Systems*, pages 33–46. Springer, 2013.
- [23] Douglas SJ De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4):419–434, 2005.
- [24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.
- [25] Dimos V Dimarogonas and Karl H Johansson. Bounded control of network connectivity in multi-agent systems. *IET control theory & applications*, 4(8):1330–1338, 2010.
- [26] X.C. Ding, A.R. Rahmani, and M. Egerstedt. Multi-uav convoy protection: an optimal approach to path planning and coordination. *Robotics, IEEE Transactions on*, 26(2):256–268, 2010.
- [27] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128. ACM, 2004.
- [28] L.E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516, 1957.
- [29] A. M. Edwards, R. A. Phillips, N. W. Watkins, M. P. Freeman, E. J. Murphy, V. Afanasyev, S. V. Buldyrev, M. da Luz, E. P. Raposo, H.E. Stanley, et al. Revisiting lévy flight search patterns of wandering albatrosses, bumblebees and deer. *Nature*, 449(7165):1044–1048, 2007.
- [30] Facebook. Facebook Connectivity Lab. <https://info.internet.org/en/story/connectivity-lab/>, 2015. [Online; accessed 01-December-2015].
- [31] E.W. Frew, D.A. Lawrence, C. Dixon, J. Elston, and W.J. Pisano. Lyapunov guidance vector fields for unmanned aircraft applications. In *American Control Conference, 2007. ACC'07*, pages 371–376. IEEE, 2007.

## Bibliography

---

- [32] Stephanie Gil, Dan Feldman, and Daniela Rus. Communication coverage for independently moving robots. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4865–4872. IEEE, 2012.
- [33] Stephanie Gil, Mac Schwager, BJ Julian, and Daniela Rus. Optimizing communication in air-ground robot networks using decentralized control. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1964–1971. IEEE, 2010.
- [34] Christopher David Godsil, Gordon Royle, and CD Godsil. *Algebraic graph theory*, volume 8. Springer New York, 2001.
- [35] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.
- [36] Google. Google Loon Project. <http://www.google.com/loon/>, 2015. [Online; accessed 01-December-2015].
- [37] G. Gu, RR Chandler, CJ Schumacher, A. Sparks, and M. Pachter. Optimum cooperative uav sensing based on cramer-rao bound. In *American Control Conference, 2005. Proceedings of the 2005*, pages 4090–4095. IEEE, 2005.
- [38] Meng Guo, Michael M Zavlanos, and Dimos V Dimarogonas. Controlling the relative agent motion in multi-agent formation stabilization. *Automatic Control, IEEE Transactions on*, 59(3):820–826, 2014.
- [39] Sabine Hauert, Jean-Christophe Zufferey, and Dario Floreano. Evolved swarming without positioning information: an application in aerial communication relay. *Autonomous Robots*, 26(1):21–32, 2009.
- [40] Tomohisa Hayakawa, Takuji Matsuzawa, and Shinji Hara. Formation control of multi-agent systems with sampled information. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4333–4338, 2006.
- [41] Andrew Howard, Maja J Matarić, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Systems 5*, pages 299–308. Springer, 2002.
- [42] M Ani Hsieh, Anthony Cowley, James F Keller, Luiz Chaimowicz, Ben Grocholsky, Vijay Kumar, Camillo J Taylor, Endo, et al. Adaptive teams of autonomous aerial and ground robots for situational awareness. *Journal of Field Robotics*, 24(11-12):991–1014, 2007.

- 
- [43] M Ani Hsieh, Anthony Cowley, Vijay Kumar, and Camillo J Taylor. Maintaining network connectivity and performance in robot teams. *Journal of Field Robotics*, 25(1-2):111–131, 2008.
- [44] Bin Jiang, Junjun Yin, and Sijian Zhao. Characterizing the human mobility pattern in a large street network. *Physical Review E*, 80(2):021136, 2009.
- [45] Alberto Jimenez-Pacheco, Denia Bouhired, Yannick Gasser, Jean-Christophe Zufferey, Dario Floreano, and Bixio Rimoldi. Implementation of a wireless mesh network of ultra light mavs with dynamic routing. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 1591–1596. IEEE, 2012.
- [46] Eric W Justh and PS Krishnaprasad. Equilibria and steering laws for planar formations. *Systems & Control Letters*, 52(1):25–38, 2004.
- [47] Y. Kang and J. K. Hedrick. Linear tracking for a fixed-wing uav using nonlinear model predictive control. *Control Systems Technology, IEEE Transactions on*, 17(5):1202–1210, 2009.
- [48] Richard Kershner. The number of circles covering a set. *American Journal of Mathematics*, 61(3):665–671, 1939.
- [49] Kwang-Yeon Kim, Jung-Woo Park, and Min-Jea Tahk. Uav collision avoidance using probabilistic method in 3-d. In *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*, pages 826–829. IEEE, 2007.
- [50] Andrew Kwok and S Martinez. Coverage control with unicycles via hybrid modeling. In *American Control Conference, 2008*, pages 2672–2677. IEEE, 2008.
- [51] E. Lalish, K.A. Morgansen, and T. Tsukamaki. Oscillatory control for constant-speed unicycle-type vehicles. In *Decision and Control, 2007 46th IEEE Conference on*, pages 5246–5251. IEEE, 2007.
- [52] S. Leven, J.C. Zufferey, and D. Floreano. A minimalist control strategy for small uavs. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2873–2878. IEEE, 2009.
- [53] Severin LEVEN. *Enabling large-scale collective systems in outdoor aerial robotics*. PhD thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2011.
- [54] Stuart Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

## Bibliography

---

- [55] Joshua A Marshall, Mireille E Broucke, and Bruce A Francis. Pursuit formations of unicycles. *Automatica*, 42(1):3–12, 2006.
- [56] Nathan Michael, Michael M Zavlanos, Vijay Kumar, and George J Pappas. Maintaining connectivity in mobile robot networks. In *Experimental Robotics*, pages 117–126. Springer, 2009.
- [57] Damjan Miklic, Stjepan Bogdan, and Rafael Fierro. Avlib: A simulink r® library for multi-agent systems research. In *Control & Automation (MED), 2012 20th Mediterranean Conference on*, pages 1043–1048. IEEE, 2012.
- [58] Nima Moshtagh and Ali Jadbabaie. Distributed geodesic control laws for flocking of nonholonomic agents. *Automatic Control, IEEE Transactions on*, 52(4):681–686, 2007.
- [59] Carl Nehme, Jacob W Crandall, and ML Cummings. An operator function taxonomy for unmanned aerial vehicle missions. In *12th international command and control research and technology symposium, 2007*.
- [60] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *Automatic Control, IEEE Transactions on*, 51(3):401–420, 2006.
- [61] Reza Olfati-Saber, Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [62] Jung-Woo Park, Hyon-Dong Oh, and Min-Jea Tahk. Uav collision avoidance based on geometric approach. In *SICE Annual Conference, 2008*, pages 2122–2126. IEEE, 2008.
- [63] S.A.P. Quintero, F Papi, D.J. Klein, L. Chisci, and J.P. Hespanha. Optimal uav coordination for target tracking using dynamic programming. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 4541–4546. IEEE, 2010.
- [64] E Rafi, S. Khan, K. Shafiq, and M. Shah. Autonomous target following by unmanned aerial vehicles. In *Defense and Security Symposium*, pages 623010–623010. International Society for Optics and Photonics, 2006.
- [65] Wei Ren and Yongcan Cao. Convergence of sampled-data consensus algorithms for double-integrator dynamics. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 3965–3970. IEEE, 2008.
- [66] Alessandro Renzaglia, Lefteris Doitsidis, Agostino Martinelli, and Elias B Kosmatopoulos. Cognitive-based adaptive control for cooperative multi-robot coverage. In *Intelligent*

- Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3314–3320. IEEE, 2010.
- [67] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 25–34. ACM, 1987.
- [68] I. Rhee, M. Shin, S. Hong, S. J. Lee, K. and Kim, and S. Chong. On the levy-walk nature of human mobility. *IEEE/ACM Transactions on Networking (TON)*, 19(3):630–643, 2011.
- [69] James F Roberts, Timothy S Stirling, Jean-Christophe Zufferey, and Dario Floreano. 2.5 d infrared range and bearing system for collective robotics. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3659–3664. IEEE, 2009.
- [70] Stefano Rosati, Karol Kruzelecki, Gregoire Heitz, Dario Floreano, and Bixio Rimoldi. Dynamic routing for flying ad hoc networks. 2014.
- [71] A.V. Savkin and H. Teimoori. Bearings-only guidance of an autonomous vehicle following a moving target with a smaller minimum turning radius. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4239–4243. IEEE, 2008.
- [72] K. Savla, F. Bullo, and E. Frazzoli. The coverage problem for loitering dubins vehicles. In *Decision and Control, 2007 46th IEEE Conference on*, pages 1398–1403. IEEE, 2007.
- [73] Mac Schwager, Brian J Julian, Michael Angermann, and Daniela Rus. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561, 2011.
- [74] Mac Schwager, Brian J Julian, and Daniela Rus. Optimal coverage for multiple hovering robots with downward facing cameras. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3515–3522. IEEE, 2009.
- [75] William M Spears, Diana F Spears, Rodney Heil, Wesley Kerr, and Suranga Hettiarachchi. *An overview of physicomimetics*. Springer, 2005.
- [76] PN Squire and R Parasuraman. Effects of automation and task load on task switching during human supervision of multiple semi-autonomous robots in a dynamic environment. *Ergonomics*, 53(8):951–961, 2010.
- [77] Guy L Steele Jr and Richard P Gabriel. The evolution of lisp. In *acm sigplan Notices*, volume 28, pages 231–270. ACM, 1993.

## Bibliography

---

- [78] R.F. Stengel. *Flight dynamics*. Princeton University Press, 2004.
- [79] Ethan Stump, Ali Jadbabaie, and Vijay Kumar. Connectivity management in mobile robot teams. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1525–1530. IEEE, 2008.
- [80] Sensefly Swinglet. Sensefly Swinglet. <https://www.sensefly.com/drones/swinglet-cam.html>, 2015. [Online; accessed 12-May-2015].
- [81] Andrew S Tanenbaum. *Computer networks*, 4-th edition. ed: *Prentice Hall*, 2003.
- [82] Herbert G Tanner, Ali Jadbabaie, and George J Pappas. Flocking in teams of nonholonomic agents. In *Cooperative Control*, pages 229–239. Springer, 2005.
- [83] D Tardioli, AR Mosteo, L Riazuelo, JL Villarroel, and L Montano. Enforcing network connectivity in robot team missions. *The International Journal of Robotics Research*, 29(4):460–480, 2010.
- [84] P. Theodorakopoulos and S. Lacroix. A strategy for tracking a ground target with a uav. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1254–1259. IEEE, 2008.
- [85] Seth Tisue and Uri Wilensky. Netlogo: A simple environment for modeling complexity. In *International conference on complex systems*, pages 16–21. Boston, MA, 2004.
- [86] V. I. Utkin, J. Guldner, and J. Shi. *Sliding mode control in electromechanical systems*, volume 9. CRC press, 1999.
- [87] G. M. Viswanathan. Ecology: Fish in lévy-flight foraging. *Nature*, 465(7301):1018–1019, 2010.
- [88] R. Weron. On the chambers-mallows-stuck method for simulating skewed stable random variables. *Statistics & probability letters*, 28(2):165–171, 1996.
- [89] R.A. Wise and R.T. Rysdyk. Uav coordination for autonomous target tracking. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 3210–3231, 2006.
- [90] Michael M Zavlanos and George J Pappas. Distributed connectivity control of mobile networks. *Robotics, IEEE Transactions on*, 24(6):1416–1428, 2008.



**Maja Varga**  
11.07.1984.  
Place du Vallon 6  
1005 Lausanne  
+41 78 799 98 20  
[maja.z.varga@gmail.com](mailto:maja.z.varga@gmail.com)

#### Research interests:

- Aerial robotics
- Distributed systems, multi-agent systems
- Swarm robotics



## Education

---

**2011. – 01.2016.** PhD in Robotics at EPFL, Laboratory of Intelligent Systems, supervisor: Prof. Dario Floreano  
**2003. - 2008.** Master in Automation and Electrical engineering at University of Zagreb, Croatia

## Professional experience

---

- 2014. – 2016.** **SMAVNET II project, EPFL** - Worked on distributed algorithms for drone networks that connect human users on the ground and adapt the communication links to their motion, applicable for urban communication networks, funded by ArmaSuisse
- 2011. - 2014.** **SWARMIX project, EPFL** - Worked on control algorithms for drones in search and rescue operations, developed hybrid controllers for drones, did a research on interaction between drones and rescue dogs, developed image-processing algorithms to detect objects on the ground
- 2011. – 2016.** **Teaching assistant at EPFL** in courses: Bio-Inspired Artificial Intelligence, Mobile Robots, Robotics practical
- 2010.** **Laboratory of Intelligent Systems, EPFL (9-months internship)** - Research stay funded by ESKAS Swiss government excellence scholarship, did a research on swarms of drones
- 2008. - 2010.** **LARICS laboratory, University of Zagreb, Croatia** - Worked on a research project on bio-inspired distributed algorithms for ground robots, developed mathematical models of honeybees' behavior, directly applicable for swarms of miniature robots

## Honors and Awards

---

"**Swiss government scholarship for foreign students**" – one year research project at EPFL, Laboratory of Intelligent Systems, 2010.

"**Top scholarship for Top students**" under the auspices of the **Croatian President and Croatian Government** awarded to the best 20 students in Croatia, 2008.

**University of Zagreb Rector Award**, 2007. , Zagreb, Croatia

**The City of Zagreb Scholarship**, awarded to best students of the University of Zagreb, 2006.

**Best student award** presented by Faculty Council, 2006. , Zagreb, Croatia

**Best student award** presented by Faculty Council, 2005. , Zagreb, Croatia

## Languages

---

<b>Croatian</b>	Native language
<b>English</b>	Fluent in speaking and writing, Cambridge certificate in Advanced English, C1 level
<b>German</b>	Intermediate, B1 level
<b>French</b>	Intermediate, B1 level

## Interests and extracurricular activities

---

Sports	skiing, rock climbing, yoga
Other	food and nutrition research, food photography, guitar and piano

## Publication list

---

Varga, Maja, Meysam Basiri, Grégoire Hilaire Marie Heitz, and Dario Floreano. "Distributed Formation Control of Fixed Wing Micro Aerial Vehicles for Uniform Area Coverage." In *IEEE/RSJ International Conference on Intelligent Robots and System (IROS), 2015*

Varga, Maja, Jean-Christophe Zufferey, Gregoire Hilaire Marie Heitz, and Dario Floreano. "Evaluation of control strategies for fixed-wing drones following slow-moving ground agents." *Robotics and Autonomous Systems* 72 (2015): 285-294.

Hauert, Sabine, Severin Leven, Maja Varga, Fabio Ruini, Angelo Cangelosi, Jean-Christophe Zufferey, and Dario Floreano. "Reynolds flocking in reality with fixed-wing robots: communication range vs. maximum turning rate." In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 5015-5020. IEEE, 2011.

Varga, Maja, Stjepan Bogdan, Marija Dragojević, and Damjan Miklić. "Collective search and decision-making for target localization." *Mathematical and Computer Modelling of Dynamical Systems* 18, no. 1 (2012): 51-65.

Varga, Maja, Zvonimir Pišković, and Stjepan Bogdan. "Multi-agent swarm based localization of hazardous events." In *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, pp. 1864-1869. IEEE, 2010.

Puljiz, David, Maja Varga, and Stjepan Bogdan. "Stochastic Search Strategies in 2D Using Agents with Limited Perception." In *Robot Control*, vol. 10, no. 1, pp. 650-654. 2012.

Varga, Maja, and Stjepan Bogdan. "Fuzzy-Lyapunov based quadrotor controller design." In *Control Conference (ECC), 2009 European*, pp. 928-933. IEEE, 2009.

