



HAL
open science

Flocking models based on local communications : From theory to simulations

Alexandre Bonnefond

► **To cite this version:**

Alexandre Bonnefond. Flocking models based on local communications : From theory to simulations. Robotics [cs.RO]. INSA de Lyon, 2023. English. NNT : 2023ISAL0040 . tel-04286379

HAL Id: tel-04286379

<https://theses.hal.science/tel-04286379>

Submitted on 15 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° d'ordre NNT : 2023ISAL0040

Thèse de doctorat de l'Université de Lyon
opérée au sein de
Institut National des Sciences Appliquées de Lyon

École Doctorale 512
Informatique et Mathématiques

Spécialité / Discipline de doctorat :
Informatique

Soutenue publiquement le 14/06/2023, par :
Alexandre Bonnefond

**Flocking Models Based on Local Communications:
From Theory to Simulations**

Devant le jury composé de :

BREDECHE Nicolas	Professeur des Universités	Sorbonne Université ISIR Paris	Président
NATALIZIO Enrico	Professeur des Universités	Université Lorraine LORIA Nancy	Rapporteur
FANTONI Isabelle	Directrice de Recherche	CNRS LS2N Nantes	Examinatrice
SASKA Martin	Senior Researcher	CTU Prague	Examineur
SIMONIN Olivier	Professeur des Universités	INSA Lyon CITI	Directeur
GUERIN-LASSOUS Isabelle	Professeure des Universités	Université Lyon 1 LIP	co-Directrice

Référence : TH0973_Alexandre BONNEFOND

L'INSA Lyon a mis en place une procédure de contrôle systématique via un outil de détection de similitudes (logiciel Compilatio). Après le dépôt du manuscrit de thèse, celui-ci est analysé par l'outil. Pour tout taux de similarité supérieur à 10%, le manuscrit est vérifié par l'équipe de FEDORA. Il s'agit notamment d'exclure les auto-citations, à condition qu'elles soient correctement référencées avec citation expresse dans le manuscrit.

Par ce document, il est attesté que ce manuscrit, dans la forme communiquée par la personne doctorante à l'INSA Lyon, satisfait aux exigences de l'Etablissement concernant le taux maximal de similitude admissible.

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	<u>CHIMIE DE LYON</u> https://www.edchimie-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	M. Stéphane DANIELE C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne directeur@edchimie-lyon.fr
E.E.A.	<u>ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE</u> https://edeea.universite-lyon.fr Sec. : Stéphanie CAUVIN Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	M. Philippe DELACHARTRE INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 philippe.delachartre@insa-lyon.fr
E2M2	<u>ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION</u> http://e2m2.universite-lyon.fr Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr	Mme Sandrine CHARLES Université Claude Bernard Lyon 1 UFR Biosciences Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX sandrine.charles@univ-lyon1.fr
EDISS	<u>INTERDISCIPLINAIRE SCIENCES-SANTÉ</u> http://ediss.universite-lyon.fr Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr	Mme Sylvie RICARD-BLUM Institut de Chimie et Biochimie Moléculaires et Supramoléculaires (ICBMS) - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 sylvie.ricard-blum@univ-lyon1.fr
INFOMATHS	<u>INFORMATIQUE ET MATHÉMATIQUES</u> http://edinfomaths.universite-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	M. Hamamache KHEDDOUCI Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 hamamache.kheddouci@univ-lyon1.fr
Matériaux	<u>MATÉRIAUX DE LYON</u> http://ed34.universite-lyon.fr Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	M. Stéphane BENAYOUN Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 stephane.benayoun@ec-lyon.fr
MEGA	<u>MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE</u> http://edmega.universite-lyon.fr Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	M. Jocelyn BONJOUR INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69621 Villeurbanne CEDEX jocelyn.bonjour@insa-lyon.fr
ScSo	<u>ScSo*</u> https://edsciencessociales.universite-lyon.fr Sec. : Mélina FAVETON INSA : J.Y. TOUSSAINT Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	M. Bruno MILLY Université Lumière Lyon 2 86 Rue Pasteur 69365 Lyon CEDEX 07 bruno.milly@univ-lyon2.fr

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Abstract

Swarms of aerial robots are no longer science-fiction. They are now used for diverse purposes such as delivery and surveillance. However, controlling a large number of drones, or unmanned aerial vehicles (UAVs), is still very challenging. In this thesis, we study the flocking strategy which is directly inspired from nature and relies on a decentralized model where the UAVs interact locally with each others thanks to communication. By analyzing the performances of existing flocking models in highly constrained environments, we identify the potential limits of these models and adapt them in order to be robuster. We consider wireless communication to be the unique channel for sharing information among the UAVs, thus we integrate a realistic communication model into the existing simulator of Virágh et al. [1] that we exploit along this work.

The analysis of the constraints within the swarms leads us to develop a new flocking model with asymmetric features, capable of evolving in highly cluttered environments without producing a single collision. This new model is named *asymmetric pressure regulation* (APR). Our last contribution consists of a distributed leader follower algorithm allowing each follower to become a leader. This characteristic greatly improves the cohesion of the swarm and foster the success of missions including the traversal of complex environments such as tunnels.

Résumé

Les flottes de robots aériens sont désormais utilisées pour de multiples applications telles que la livraison ou encore la surveillance. Cependant, contrôler un grand nombre de drones ou UAVs (unmanned aerial vehicles) demeure un défi important. Dans cette thèse, nous étudions des stratégies de flocking, directement inspirées de la nature et reposant sur un modèle décentralisé où les entités concernées interagissent localement par le biais de communications. En analysant les performances de modèles de flocking existant dans des environnements très contraints (par des obstacles), nous identifions les potentielles limites de ces modèles et nous proposons de les adapter pour les rendre plus robustes. Dans ces travaux, nous considérons les communications sans fil comme étant le seul moyen d'accéder aux informations des voisins, ainsi nous intégrons un modèle de communication réaliste au simulateur de flocking de Viragh et al. [1].

L'analyse des contraintes de pression au sein des flockings nous conduit à développer un nouveau modèle introduisant des interactions asymétriques et capable de faire évoluer les agents (UAVs) dans des environnements très contraints sans générer de collisions, nommé APR (Asymmetric Pressure Regulation).

Notre dernière contribution consiste en un algorithme de type leader-follower distribué permettant à tout agent follower de devenir un leader. Cela renforce considérablement la cohésion de la flotte et donc favorise le succès de missions consistant à traverser des environnements complexes comme des tunnels.

Contents

Abstract	iii
Résumé	iv
Abbreviations and Nomenclature	xii
1 Introduction	1
1.1 Context and Challenges	1
1.2 Thesis Organization	4
1.3 Publications	5
2 State of the Art	6
2.1 UAVs Swarm Navigation	6
2.2 Flocking Models	8
2.2.1 Reynolds' Boids	9
2.2.2 The Vicsek Model	12
2.2.3 Olfati-Saber's Model	15
2.2.4 Vásárhelyi et al.	21
2.3 Information Flow in Swarm of UAVs	25
2.3.1 Exchanging information in flocking models	25
2.3.2 The communication and mobility trade-off	27
2.3.3 Overview on communication in multi-robot simulators	28
3 Simulation of UAV Flocks and Degraded Communication	31
3.1 The Simulation Context	32
3.2 Aerial Robot Model	33
3.3 Simulating Degraded Communication by Obstacles	35
3.3.1 Log distance path loss model	37
3.3.2 Neighborhood definition	40
3.4 Environments and Settings	42
3.4.1 The physical environment	42
3.4.2 Initial conditions and configuration of an experiment	44
3.5 Metrics and Optimization Process	46

3.5.1	Measuring the performances of the flocking	46
3.5.2	Optimization based on a genetic algorithm	48
4	Extension of Vásárhelyi et al. and Olfati-Saber models	52
4.1	Impact of Obstacles on Collective Motion	53
4.2	Extending Flocking Models	55
4.2.1	Vásárhelyi and Attraction: VAT	56
4.2.2	Vásárhelyi and Olfati-Saber: VOS	57
4.3	Optimization and Comparison	58
4.3.1	Optimization process	58
4.3.2	Results and comparison	60
4.4	Exploiting the Equilibrium Distance	65
4.4.1	Context	65
4.4.2	Introduction of a new distance metric	66
4.4.3	Performances Analysis	67
5	Asymmetric Pressure Regulation	69
5.1	Motivations for a New Asymmetric Model	70
5.2	An Individual Pressure Measurement	70
5.3	Pressure Based Repulsion	72
5.4	Adaptive Neighborhood for Attraction	74
5.5	The Asymmetric Pressure Regulation Model (APR)	76
5.6	Experimentation & Results	78
5.6.1	Simulation settings	78
5.6.2	Metrics and parameters optimization	78
5.6.3	Results	80
5.6.4	Analysis of pressure dissipation	83
6	Incremental Leader-Follower Model	88
6.1	Context and Objectives	89
6.2	Incremental Leader Follower Strategy	90
6.2.1	Navigational feedback	90
6.2.2	Incremental leadership	91
6.3	Simulation and Comparison	95
6.3.1	APRT and VATT	96
6.3.2	ILF algorithm and classical leader follower algorithm	98
7	Conclusions	100
7.1	Synthesis	100
7.2	Perspectives	102

List of Tables

3.1	Parameters of the robot model	34
3.2	Parameters of the communication model	36
3.3	Parameters of the LDPL model	39
3.4	Parameters of the environment	44
3.5	Parameters of an experiment	45
3.6	Metrics for the performances of a flocking model	46
3.7	Fitness functions for the performances of a flocking model	48
4.1	Fitness functions for the performances of VAT and VOS	59
4.2	Parameters of the <i>generic</i> environment	59
4.3	VAT and VOS optimized parameters in GENERIC env. The columns correspond to the fitness functions F_{vel} , F_{col} , F_{clust} , F_{wall} , F_{corr} and F_{pow}	60
4.4	Parameters of the <i>forest</i> and <i>city</i> environments	61
4.5	VA performances extracted from [2].	62
4.6	FOREST	63
4.7	CITY	63
5.1	Parameters of the <i>generic</i> , <i>forest</i> and <i>city</i> environments	78
5.2	Metrics used for the APR model and the objective value	79
5.3	Fitness functions for the performances of the APR model	79
5.4	Performances of the APR model in the Generic environment.	81
5.5	Performances of the APR model in the Forest and City environments.	81
5.6	Performances of the VAT model in the Generic, Forest and City environments.	83
6.1	Performances of VATT and APRT with the incremental leader follower algorithm.	97
6.2	Performances of ILF and CLF algorithm with APRT model.	99

List of Figures

1.1	Flock of starling. Credits: NICK DUNLOP [3]	2
2.1	Intel Breaks Guinness World Records Title for Drone Light Shows in Celebration of 50th Anniversary. Credits: [4]	6
2.2	A flock of starlings flies in formation in the sky above Shapwick Heath National Nature Reserve in Somerset. Credits: [5]	7
2.3	Composite image of the aerial swarm in a bamboo forest. Credits: [6]	8
2.4	Batman Returns: Relighting of the Tree Ceremony scene. Credits: [7]	9
2.5	The three fundamental flocking interactions as defined by Craig Reynolds [8].	10
2.6	Projection of the boid onto the silhouette edge of the obstacle. Credits: [8] .	11
2.7	Velocity field for $N = 300$ self-propelled particles. The actual velocity is represented by a small arrow and the last 20 time-step is displayed as a short continuous curve before the arrow. Credits [9]	13
2.8	The order parameter v_a versus the noise on the left image and versus the density on the right image. Credits [9]	14
2.9	Agent i and its neighbors based on the definition of Eq. 3.9.	16
2.10	Representation of α -lattices. Credits [10]	16
2.11	Plot of the potential and action functions. Credits [10]	17
2.12	The fragmentation phenomenon. Credits: [10]	19
2.13	Projection of a β -agent on a spherical obstacle. Credits [10]	20
2.14	150 agents squeezing themselves to go through highly constraining obstacles. Credits [10]	21
2.15	Visual interpretation of the repulsion term in blue and the alignment term given by the maximum allowed velocity difference between two agents $v_{ij}^{frictmax}$ in green. Credits[2]	23
2.16	Agent i projecting a skill agent on an obstacle to perform the obstacle avoidance interaction.	24

2.17	Probability distribution of the communication outages as a function of distance. The database was gathered from a 5-min section of a general flight with 32 drones in a remote open-air setting. Each drone logged a 5-Hz sampling of the elapsed time since the reception of the last status packet from all other drones. This value was matched later with the position of the drones recorded accurately by each drone onboard. Color indicates timeout probability in each bin for a given distance. Average timeouts with standard deviation and with the number of data points are indicated on the right for 50m distance gap, whereas the black line on the plot indicates average and std of timeout for each distance bin of 10m. [2]	27
2.18	Snapshots of different robots simulators.	29
2.19	GAZEBO [11] (our flocking simulation)	29
2.20	Snapshots of different network simulators.	30
3.1	Main window of the simulator displaying the agents evolving in the arena.	33
3.2	Received power for a given scenario with varying size of obstacle.	38
3.3	Representation of the directed graph with 4 nodes and their respective communication neighborhood (CN) represented by the gray circles. The full arrows represent an effective communication link while the dashed arrows represent an uncertain communication as there is an obstacle in the middle.	41
3.4	Representation of the active neighborhood with 4 agents. Here $N_{max} = 1$, <i>i.e.</i> each agent exploits only information from its closest neighbor (black directed edges correspond to these data exchanges). Gray ellipses illustrate the topological neighborhood, that we call the active neighborhood (AN).	42
3.5	Obstacle file and its equivalent in the simulator.	43
3.6	Representation of the mapping functions with $\alpha = 3$, $\beta = 4$ and $l = 7$, $\kappa = 3$	47
3.7	Evolution of the fitness functions during the optimization.	50
3.8	Evolution of two flocking parameters during the optimization.	51
4.1	20 agents being separated by a convex obstacle. The agents are moving from the right to the left of the environment.	53
4.2	3 agents interacting together at two different time. On the left, at $t = 0s$, the interaction ranges represented by the colored disks, are focused on agent 1. On the right, at $t = 5s$, the focus is on agent 2. The plain black arrows represent an interaction while the dashed arrows represent an uncertain interaction due to the obstacle.	54
4.3	20 agents heading in the corner of a concave obstacle.	55
4.4	Representation of the different parameters depending on the distance between two agents.	57
4.5	Evolution of the metrics during optimization using degraded communication in the genetic environment (VAT in red, VOS in blue).	60
4.6	Snapshots of the testing environments.	61

4.7	Inter-agent distance measured between the agents and their closest neighbor. The red, blue and green lines correspond respectively to the average, the maximum and the standard deviation of the measure. The black line corresponds to the average value over the whole experiment and we have $d = 1200cm$. . .	65
4.8	Superposition process leading to the modification of the observed equilibrium.	66
4.9	Optimization process with the distance and standard deviation metrics. . . .	67
4.10	Inter-agent distance measured between the agents and their closest neighbor. The red, blue and green lines correspond respectively to the average, the maximum and the standard deviation of the measure. The black line corresponds to the average value over the whole experiment and we have $d = 2100cm$. . .	68
5.1	Representation of the pressure with 3 agents depending of the inter-agent distance.	71
5.2	Measure of the pressure in a narrow path. The pressure is represented by the yellow circles around the agents. The larger the circle, the greater the pressure measured.	72
5.3	Comparison of two models with 4 agents. On the left we use a classical repulsion and a pressure repulsion on the right.	73
5.4	Representation of three agents at the equilibrium two by two in Phase 1 and once they are stabilized in Phase 2 (VAT model).	74
5.5	Representation of 5 agents with $N_{max} = 2$. The thick black arrows represent the current velocities of the agents.	75
5.6	Representation of the attraction in red based on the neighbors filter.	76
5.7	Attraction forces among a 15 agents swarm represented by the red arrows. The actual velocity of each agent is not represented here.	77
5.8	Optimization process of the APR model with the NSGA-III algorithm. . . .	80
5.9	Initialization of the experiment with 15 agents.	84
5.10	Measure of the pressure (y axis) over time (x axis). The red, green, blue and yellow lines respectively represent the average, the standard deviation, the minimum and the maximum of the pressure among all the agents.	85
5.11	Measure of the pressure (in cm) over time (in sec). The red, green, blue and yellow lines respectively represent the average, the standard deviation, the minimum and the maximum of the pressure among all the agents.	85
5.12	Behavior of the models towards obstacles	87
6.1	Graphical representation of two leadership approaches: on the left the classical one and on the right the hierarchical one.	89
6.2	Two examples of HL -flocks and one example of a non- HL -flock. The arrow $i \rightarrow j$ represents that agent i is led by agent j . Credits: [12].	92
6.3	Representation of the graph with the incremental leader follower algorithm. The colors represent the leading rank with $lr = 0, lr = 1, lr = 2, lr = 3$ respectively for the red, green, blue and yellow colors.	94

6.4	Swarm of 16 agents applying the ILF algorithm in a narrow corridor. The colors represent the different leading ranks.	95
6.5	S-shaped environment.	96
6.6	APRT with incremental leader follower	97
6.7	VATT with incremental leader follower	97
6.8	Swarm of drones at the entrance of the corridor (leader in red color)	98
6.9	APRT with ILF algorithm.	99
6.10	APRT with CLF algorithm.	99
7.1	a. Snapshot of 5 IntelAero drones flocking together with the VAT model in the GAZEBO simulator b. Experiment of a 3-drones flocking by the Chroma team on April 2023 (CITI lab.)	101

Abbreviations and Nomenclature

Nomenclatures

- UAV Unmanned Aerial vehicle.
VAT Vásárhelyi and Attraction.
VATT Vásárhelyi and Attraction and Tracking.
VA Vásárhelyi.
VM Vicsek Model.
VOS Vásárhelyi and Olfati-Saber.
APR Asymmetric Pressure Regulation.
APRT Asymmetric Pressure Regulation and Tracking.
ILF Incremental Leader Follower.
CLF Classical Leader Follower.
UWB Ultra-Wide-Band.
 N Number of agents.
 PL Path loss.
 γ Path loss exponent.
 \mathcal{N}_i Communication neighborhood of agent i .
 \mathcal{N}_i^+ Active neighborhood of agent i .
 \mathcal{R}_{rep} Repulsion area.
 \mathcal{R}_{att} Attraction area.
 p_i Pressure of agent i .
 J_{ij} Similarities coefficient between i and j
 LA Leader Assignment algorithm.

1.1 Context and Challenges

Collective motion is ubiquitous in nature, from the microscopic scale to the macroscopic one, from a few particles up to half a million birds. It is a spectacular phenomenon where the will of a fraction of a group can impact the entire population. In Fig. 1.1, we can see this amazing choreography performed by thousands of starlings and captured by NICK DUNLOP. This invisible power relies on local interactions which propagate among all the reachable individuals. The flocking behavior is the perfect example of collective motion. If the reasons why some multi-cellular organisms flock together are still not fully understood by scientists, these latter yet have a pretty good understanding of how it is performed. It is ruled by three local interactions which are the attraction, the repulsion and the alignment allowing respectively to stay together, to avoid collision, and to move in the same direction. The flocking has even sparked inspiration into many researchers work, initially as a way to simulate the flocking behavior of birds for cinematographic use, in the seminal work of Craig Reynolds [8]. More recently, thanks to the technological breakthroughs in terms of communication and robotics, it is now possible to integrate this bio-inspired movements directly into swarm of UAVs communicating together [2]. The challenge is to elaborate autonomous flights and manage complex missions while taking advantage of the flocking model.

Such a fascinating behavior depends on an efficient information transmission process that has to be correctly designed in order to ensure cohesion. In nature, this process takes different forms and mostly relies on visual observation for birds [13] or fish [14] for example but can be also induced by other means such as the magnetic field if we consider ferromagnetic particles [15] for instance. When it comes to mobile robots and UAVs, different sensors can also be used in order to acquire and share information, from vision-based sensors like cameras [16] and Ultra-Violet sensors [17] to communication sensors based on various technologies such as cellular network [18], Wi-Fi [2], Ultra-Wide-Band [19] or LoRa [20]. These different methods can also be combined to improve the accuracy of the system. The different sensors are yet not flawless and their reliability is highly dependent of the environment.

The main topic that will be discussed in this thesis regards how the information that has



Figure 1.1: Flock of starling. Credits: NICK DUNLOP [3]

been acquired through the different sensors we mentioned previously is being “transformed” into motion. This conversion can be divided into two main parts which are the selection of this information and the definition of the control law based on this information and resulting in a desired velocity. The selection of the information is mostly equivalent to the definition of the neighborhood of an individual, also called agent. Basically, it defines the set of agents whose information will be used to compute the control law. Once again, nature taught us different ways of doing it [21]. Whether it is based on a metric, usually a distance [22] where an agent is considered to interact with all its neighbors within a given distance, or a topological neighborhood [23], only selecting a fixed number of neighbors among the nearest ones. The topological neighborhood is also sometimes based on a *Voronoi* diagram [24] thus breaking free of any metric. This non-exhaustive list of methods have their own advantages and limitations even though it has been shown that, regarding flocking, topological neighborhood seems to be more efficient at bringing order [25]. Once the neighbors have been identified, there is also a filtering action consisting of getting rid of the information which is not useful for the control law.

The control law is the central point of the flocking behavior, its nervous system, taking information as input and outputting a new information which can be directly used as a command (if we talk about mechanical systems) and hence produce motion. This control law defines the interactions that are wished between an agent and its neighborhood. As we mentioned before, the interactions for flocking are the attraction, the repulsion and the alignment. There is an infinite number of ways to define these interactions but the overall control law should respect some conditions such as the feasibility (in terms of hardware) or the stability to name but a few of them. In most of the cases, the interactions are reciprocal

and symmetric [26, 1] but there are also asymmetric interactions [27, 28], it usually depends on the organization of the swarm but also the available sensors. For example, UAVs with a front camera cannot interact with neighbors behind them and it is the same for birds as they do not have an omnidirectional vision. The locality of the interactions is also a crucial aspect of the definition of the control law [29]. In other words, should the strength of an interaction be limited to a fixed distance or not and how it should evolve accordingly? In this thesis we seek to address this question which is at the core of the model definition.

Of course, defining a flocking model would not be possible without the right tools to evaluate its performances. Thanks to the pioneer work of Vicsek et al. [9] and the introduction of statistical physics models for collective motion, it is now possible to measure the emergence of some features specific to collective behaviors. They are studied in the context of non-equilibrium continuum dynamical models applied to large groups of biological organisms (flock of birds for example) [30]. In this thesis, we place ourselves in the same context, considering swarms of self-propelled UAVs flocking together. Numerical models will be presented and evaluated using a dedicated simulator [1]. An optimization strategy will also be introduced because of the complexity of the models, ie. depending on several parameters, in order to allow a fair comparison between them.

This thesis is funded by the DGA/INRIA **DYNAFLOCK**¹ project which lies into the distributed AI field, and more precisely into swarm intelligence, applied here to decentralized multi-agent systems. It aims at studying and experimenting flocking models with simulated UAVs and real robots equipped with wireless communication. These different aspects of the project directly impact the various hypotheses that we consider in this thesis. In particular, communication will be the only way for agents to know (estimate) the position and velocity of their neighbors. Regarding the perception of obstacles, we consider that any UAV is capable of sensing an obstacle within a given range.

In this thesis, we aim at coping with most of the challenges we mentioned above. Hence, we address multiple issues that can be summarized into the following items:

1. Many flocking models available in the literature are usually studied in a *free-flocking* context which means that there are no obstacles. In addition, the working assumptions are not always clear or coherent with realistic conditions. Defining an appropriate testing framework independent of the particular objectives sought by authors would be a major advance in order to fairly evaluate the performances of the models and thus identify their advantages and their limits.
2. Among the large amount of flocking models in the literature, only a few deal with the impact of obstacles over the communication and the overall motion of the swarm (for

¹Flocking dynamique fondée sur la qualité de liaison pour la coordination décentralisée de flottes de drones, O. Simonin, I. Guérin-Lassous, 2020-2023

the flocking models based on communication). This aspect will be considered all along the thesis when presenting flocking models and developing new ones.

3. Flocking models are usually established on three local interactions based on state variables such as position and velocity. Even though this formulation has proven to be efficient, it is not the only one. As we get access to richer and more accurate information, it could be interesting to study if we can go beyond this classical approach and infer new information allowing the design of flocking models more robust.
4. Last but not least, given the performances of the flocking, could we use it to perform complex tasks such as following a target while evolving in highly constrained environments such as tunnels or forests?

1.2 Thesis Organization

The rest of this thesis is organized as follows.

In Chapter 2, after presenting some examples of actual UAVs swarm applications, we survey existing research on flocking models in a chronological way. The idea is to observe the evolution of the control laws as well as the evaluation of the performances. Keeping that same order, we analyze how the information is being accessed and transmitted in these different models and finally point out the limitations of actual simulation tools resulting in unwanted behaviors when performing real experiments.

Chapter 3 lays the foundation of the framework we aim at developing starting with the simulation environment and the model of the UAV we use in terms of hardware, including flaws and stochasticity, but also the equation of motion resulting in the equation of an acceleration. We then present the communication model we define and the impact of obstacles on the communication quality. This eventually lead us to a new definition of the neighborhood based on the ability of an agent to communicate. The next layer of this framework consists of the “physical” environment and the settings of a simulation. This aspect is fundamental in order to be able to repeat an experiment and thus validate or compare the performances of a model. Finally, we present the methods and tools we use in order to evaluate the performances of a model and optimize it.

In Chapter 4, we focus on two models we presented in Chapter 2 which are the one of Olfati-Saber [10] and the one of Vásárhelyi et al. [2]. Within our framework, we identify their limits and come up with two extended models, namely **VAT** and **VOS**, which are supposed to prevent the main drawbacks of their original versions which are the fragmentation and the collision. The performances of the models are evaluated in different environments after going through an optimization process. As we decided to integrate the communication quality within the optimization strategy, we faced some unwanted phenomena that made us

rethink the optimized parameters.

In chapter 5, we develop a new flocking model based on asymmetric interactions and using a new measure called *pressure* that concatenates multiple information at once. We also define a *neighborhood filter* that identifies the agents that are most likely to be separated from the swarm. The new model called **APR** for *asymmetric pressure regulation* is compared to the previous VAT model and its performances are analyzed after an optimization process.

Given the encouraging performances of the APR model, we present in chapter 6 a new algorithm that can be applied to the APR model in order to perform *leader-follower* navigation. We propose a new algorithm called **ILF** for *incremental leader follower* algorithm that “distributes” the leadership and hence reduces the risks of fragmentation. We evaluate the performances of this algorithm in a winding tunnel environment pushing the limits of both the ILF algorithm and the APR model.

Chapter 7 concludes this manuscript and discusses the different perspectives that we have identified.

1.3 Publications

- **IROS 2021** IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Pages 9139-9145, Prague, Czech Republic, September 2021. IEEE. ISBN 978-1-66541-714-3. doi: 10.1109/IROS51168.2021.9635944. A. Bonnefond, O. Simonin, I. Guérin-Lassous, “Extension of Flocking Models to Environments with Obstacles and Degraded Communications”
- **ROIA** Revue Ouverte d’Intelligence Artificielle (à paraître, accepté en 2022), A. Bonnefond, O. Simonin, I. Guérin-Lassous ”Modèles de Flocking Adaptés aux Environnements avec Obstacles et Communications Dégradées”
- **JFSMA 2021** Journées Francophones sur les Systèmes Multi-Agents, A. Bonnefond, O. Simonin, I. Guérin-Lassous ”Extension des Modèles de Flocking aux Environnements avec Obstacles et Communications Dégradées”

2.1 UAVs Swarm Navigation

Swarms of drones are no longer science fiction. Over the past decades, it has gained interest among many communities. Nowadays, drones are used in various contexts from entertainment to industrial applications. Thanks to recent technological breakthrough, drones are becoming lighter, faster and less prompt to failures. Improvements in terms of sensors and communication are also at the heart of drones collaboration as they are cheaper, smaller and more easily integrable into aerial robots.



Figure 2.1: Intel Breaks Guinness World Records Title for Drone Light Shows in Celebration of 50th Anniversary. Credits: [4]

One of the most impressive example is the drone light show performed by the company Intel in 2018 for its 50th anniversary. They broke the Guinness World Record by flying not less than 2,018 drones during a spectacular airborne performance as it can be seen in Fig. 2.1. Such a performance is made possible thanks to the use of *artificial potential field* as

it is explained in [31]. With this method, the user can accurately define local interactions (desired force) to apply depending on the position of the drone. This system is entirely centralized and most of the computation is done by a ground control station correcting the positions of the drones regarding some predefined paths. Even if this thesis will be more focused on decentralized models, the mathematical methods applied to centralized models for path planning give us interesting insights for our work. The main difference is the access to the information of all the entities involved and the knowledge of the environment.

Other applications include *Swarm-based Drone-as-a-Service* (SDaaS) where a swarm of drones is used for delivery [32]. Once again this application is centralized and includes a complex architecture with different intermediate actors like the consumer, the provider and the infrastructure for example.

Of course swarm of drones are also used for research purposes to investigate the emergence of a collective motion regardless of any exterior objective. This topic is the one we are interested in in this thesis and especially the flocking behavior which is mostly inspired from nature like flocks of starling birds as an instance. In Fig. 2.2, a “murmuration” of starlings can be seen. This type of dramatic formation can gather up to 750,000 individuals.



Figure 2.2: A flock of starlings flies in formation in the sky above Shapwick Heath National Nature Reserve in Somerset. Credits: [5]

In their wish to mimic nature, researchers have developed flocking algorithms which are entirely decentralized as it is done in the seminal work of Craig W. Reynolds [8]. Recent articles also deal with realistic constraints leading to real outdoor experiments as in [1]. In [2], Vásárhelyi’s et al. manage to fly up to 32 drones in outdoor conditions. However, the system was subject to multiple communication failures which forced the authors to extrapolate neighbors positions based on global information. In 2022, Zhou and al. [6], develop miniature and fully autonomous drones with an highly efficient trajectory planner

based on limited information of onboard sensors. They manage to fly multiple drones in cluttered environments such as a bamboo forest as it can be seen in the composite image in Fig. 2.3.



Figure 2.3: Composite image of the aerial swarm in a bamboo forest. Credits: [6]

In the same fashion, Soria et al. [33] achieved rapid and safe collective motion with dense obstacles by using potential fields into model predictive control (MPC). The high computational cost of this method requires a centralized architecture. Also, the experiments are done indoor using a motion capture system facilitating the localization of the drones. This performance was also conducted in a decentralized way in the great article of Ahmad et al. [34].

Flocking control is not the only field of research when it comes to swarm of drones. Other topics such as consensus [35], coverage [36], connectivity preservation [37, 38, 39, 40, 41], and pattern formation [42] are also widely studied in and applied to various types of autonomous agents. The overall goal is the development of distributed control laws in order to reach a global objective.

In this thesis, we focus on flocking models suited for outdoor applications with multiple obstacles. This implies different hypothesis regarding perception, communication and localization that will be discussed throughout this thesis.

2.2 Flocking Models

In this thesis, we focus on the collective motion among swarm of self-propelling UAVs without any global objective. The self-propelling characteristic, ensuring persistent motion, is necessary to the emergence of an order among the swarm [43]. Moreover, it prevents the system from being stuck in an equilibrium state. The overall objective is to find efficient strategies in order to increase the performances of flocking models while taking into account various constraints such as the environment as well as the communication. So as to understand the various aspects of this subject, we organize this section as a chronological survey

outlining different flocking strategies and their properties. Even though our work is focused on autonomous agents evolving freely, we also consider in this section, models that include group objectives or navigational feedback as they are essential to understand the challenges ahead.

2.2.1 Reynolds' Boids

The “Boids model” is perhaps the first well known flocking model. It was published by Reynolds in 1987 [8] and aimed at simulating swarm of “bird-oid” objects also called “boids” even when they represent other sorts of creatures such as schooling fish. Computer graphics was the first motivation of this work and led to realistic simulations of large flocks like, for example, in 1992 with the Tim Burton movie *Batman Returns*. We can see this in Fig. 2.4 where hundreds of bats are flocking together around the Christmas tree. This rendering has been done thanks to Reynolds work.



Figure 2.4: Batman Returns: Relighting of the Tree Ceremony scene. Credits: [7]

The model defined by Reynolds is deterministic and relies on three fundamental interactions that are the separation, the alignment and the cohesion. These interactions are represented in Fig. 2.5. Given these local interactions, the simulated boids should steer to avoid crowding local flockmates (separation), steer towards the average heading of local flockmates (alignment) and steer to move toward the average position of local flockmates (cohesion). We understand here that the motion of a boid is entirely determined by its peers in its neighborhood. The *collision avoidance* (separation) and the *velocity matching* (alignment) are complementary as the first one establishes a minimum distance between the agents while the last one tends to maintain this distance. The alignment is the predictive version of the collision avoidance and is only based on the velocity. Indeed, by aligning the agents velocity together, it prevents them from having crossing trajectories and thus anticipates potential collisions. Together, these two forces ensure a safe flocking. The *flock centering*

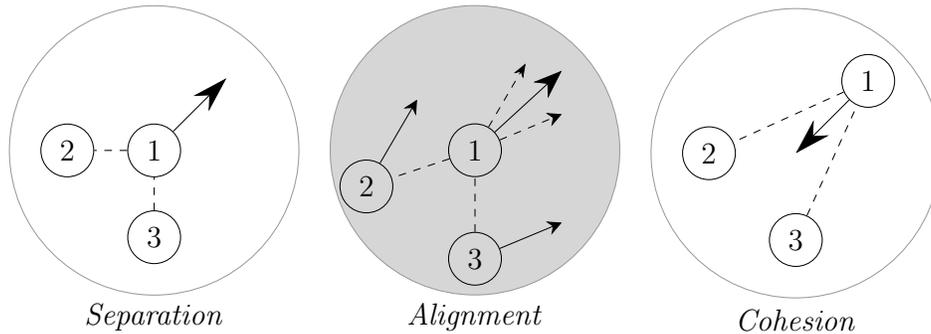


Figure 2.5: The three fundamental flocking interactions as defined by Craig Reynolds [8].

(cohesion), as we mentioned above, causes the boids to steer toward denser regions (in terms of boids population) and appears necessary when for instance the flock is separated around an obstacle. However, because the interactions are local, if a flock is too widely separated, it will not be able to merge back together. In that sense, the author assumes that his model is better to represent a school or a herd as the animals involved have a relatively short-range and limited perception of their environment in comparison with birds.

Regarding the control of the boids, the three forces produce independent *acceleration requests* that are then passed into a navigation module which goal is to prioritize, combine and adjust these requests with regard to the situation (risk of collision or obstacles ahead). Indeed, one may be tempted to average the strength and heading of the accelerations without checking if the resulting motion is dangerous. The accelerations could cancel out leaving the motion of the boid unchanged leading potentially into an obstacle. To avoid this situation, Reynolds uses an approach called the *prioritized acceleration allocation*. It prioritizes the requests in terms of magnitude and stack them into an accumulator. While the sum of the accumulated magnitudes is not larger than a *maximum acceleration* value proper to the boids, it keeps stacking. The last request is truncated to compensate for the excess of accumulated magnitude. Doing so, the most pressing requests are satisfied sometimes at the expense of others. For instance, the acceleration resulting from the cohesion could be ignored in favor of a maneuver to avoid a collision with a flockmate or an obstacle.

So far, we assumed that the boids could “acquire” information freely but this process is perhaps the most important step yielding to a collective motion. In his model, Reynolds’ boids have direct access to all the local geometric information of the environment thus including the position, orientation and velocity of neighboring boids. In real world, this information results of a cognitive process following a perception step itself strongly impacted by imperfect senses and environment conditions. The *simulated perception* of Reynolds consists of a spherical zone of sensitivity centered at the boid local origin. An unbounded perception of the environment would be unrealistic but could also lead to harmful behaviors (considering that the interactions would be applied to this entire perception field). It is quite easy to see that applying a cohesion force to the entire flock would lead all the boids to converge to the same point which is not a good idea. It is even stated that “the aggregate

motion that we intuitively recognize as “flocking” (or schooling or herding) **depends** upon a limited, localized view of the world” [8]. This concept will be very important for the rest of this thesis.

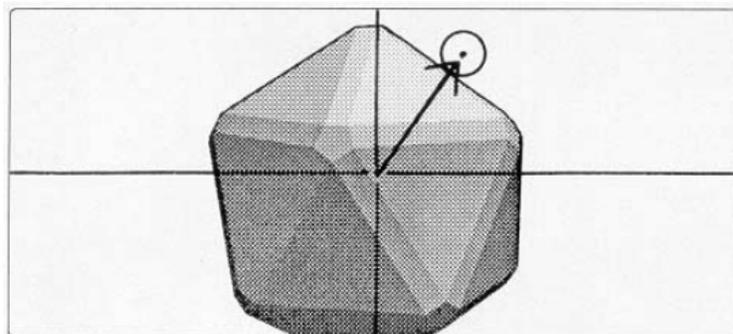


Figure 2.6: Projection of the boid onto the silhouette edge of the obstacle. Credits: [8]

Last but not least, the author also deals with the interaction of the flock with other objects in the environment. Indeed, as we mentioned earlier, the main purpose of this model is to suit computer graphics animations which include some scenes with different kind of obstacles. These obstacles can be geometrically represented and simplified to suit the simulation framework (the complexity of the shape for rendering would be too costly to use). Interacting with the environment appears to be an actual challenge of the flocking as the obstacles can be seen as constraints without which the flock tends to reach a steady state. Two strategies of environmental collision avoidance are implemented in this paper. The first one is based on *force fields* and the second one is called *steer-to-avoid*. The force field concept suggests the existence of a repulsion force coming out of the obstacle which strength is stronger as a boid gets closer to that obstacle. This method is easy to implement but has some limitations such as the particular case of a boid moving toward an obstacle with a heading collinear to the force field (but opposite directions). The force field would only slow the boid down but not induce any change of heading which is not a good strategy. Also, boids moving alongside an obstacle should not be repelled, only those flying toward it. To overcome these limitations, the *steer-to-avoid* strategy considers only obstacles directly in front of the boids and computes the appropriate acceleration in order to steer the boids in a safe region along the obstacle ahead. This acceleration vector is based on a projection of the boid onto the silhouette edge of the obstacle closest to the point of eventual impact (see Fig. 2.6).

This pioneer work of Reynolds is a necessary step in understanding flocking in general. In order to go deeper in the field of collective motion we will now talk about the Vicsek model [9] whose approach relies more on statistical physics.

2.2.2 The Vicsek Model

The Vicsek Model, referred to as “VM” from now on, was developed in 1995 by Vicsek et al. [9] in order to study the transition from disordered to ordered states among homogeneous polar¹ self-propelled particles. This study is conducted around two main variables which are the density of particles and the noise introduced in the model. The model itself is very simple and is based on a unique rule: “at each time step a given particle driven with a constant absolute velocity v_0 assumes the average direction of motion of the particles in its neighborhood of radius r with some random perturbation added” [9]. This rule can be translated by two equations applied to N particles evolving in a 2D square arena of size L . The first one regards the motion at a constant speed:

$$\mathbf{q}_i(t + 1) = \mathbf{q}_i(t) + \mathbf{v}_i(t)\Delta t \quad (2.1)$$

where Δt is the time-step, $\mathbf{q}_i(t)$ represents the position of particle i at time t and $\mathbf{v}_i(t) = v_0\mathbf{s}_i(t)$ its velocity. $\mathbf{s}_i(t)$ corresponds to the orientation of the particle velocity which can be defined by an angle $\theta_i(t)$ and we have $\mathbf{s}_i = (\cos \theta_i, \sin \theta_i)$. Hereafter, bold notation will be used to define vectors.

And the second equation concerns the update of the direction based on the particle neighborhood of radius r :

$$\theta_i(t + 1) = Arg \left[\sum_j \mathbf{v}_j(t) \right] + \eta\xi_i(t) \quad (2.2)$$

where $Arg(x)$ measures the angle of x in radians, j represents all the particles in the neighborhood of particle i , $\xi_i(t)$ is a delta-correlated noise uniformly distributed in $[-\pi, \pi]$ and η is the noise strength. Equation 2.2 corresponds to the alignment interaction based on the measure of the average velocity direction of the agent neighborhood.

We can see that the only parameters of the model are the absolute velocity v_0 , the noise strength η and the density of particles $\rho = N/L^2$ (the number of particles in the surface of the arena).

This model is very popular as it is simple and appears to give rise to some different patterns depending on the noise applied and the density. For example we can see in Fig. 2.7 that the image on the left has relatively high density and noise resulting in a poorly correlated motion (correlation here refers to the alignment among the agents). As a comparison, the image on the right has a higher density and almost no noise and we can see that the particles are almost all aligned. We can already see here the impact of these parameters over the global behavior of the particles as a group.

These different patterns, also called phases² by Vicsek, are at the heart of his article [9] and especially the transition from one phase to another. A typical example of phase transition is the freezing of a fluid switching from a liquid state to a solid one. These transitions are

¹Polar self-propelled particles have distinct heads and tails, and are propelled head-first

²In mechanical systems, the phase usually consists of a given position and momentum of a particle but can be extended here to all the particles.

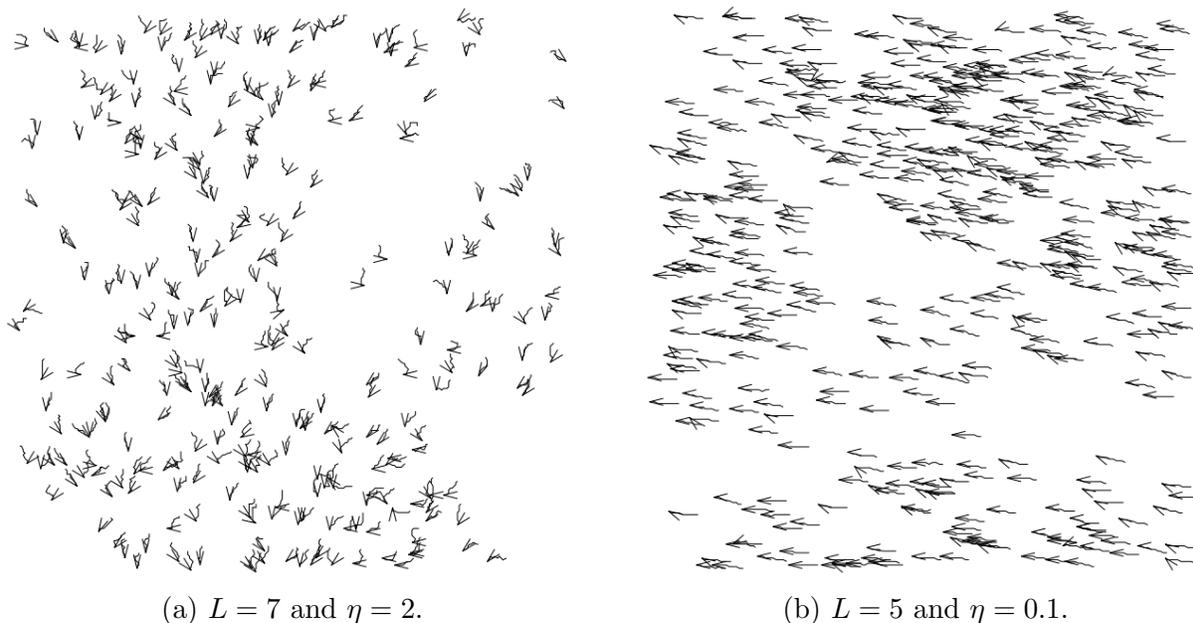


Figure 2.7: Velocity field for $N = 300$ self-propelled particles. The actual velocity is represented by a small arrow and the last 20 time-step is displayed as a short continuous curve before the arrow. Credits [9]

usually defined by some changes of some specific parameters of the system named *order parameters* and are correlated to the symmetry characterizing a phase [43]. In the case of flocking here, and as we can see in Fig. 2.7, the different phases can be differentiated by the average normalized velocity. As a result, Vicsek defines the order parameter v_a as follows:

$$v_a = \frac{1}{Nv_0} \left| \sum_{i=1}^N \mathbf{v}_i \right| \quad (2.3)$$

Thanks to this definition, it is now possible to quantify the phase of the system. If the particles are highly disordered, the velocities are in random directions and hence the order parameter is close to zero. On the contrary, if there is order in the motion then all the particles are on average moving in the same direction and the order parameter is one.

Thus, the order parameter can be used as an accurate indicator of the **phase transition** by monitoring its value while some parameters vary. In Fig. 2.8a we can see how the order parameter and hence the global symmetry of the particles are impacted when the noise increases at a fixed density for various number of particles. When the noise strength is high, the order parameter is very low which makes sense as the particles heading tend to cancel out. Also one can show that the sum of N randomly oriented unit vectors has a modulo of \sqrt{N} [44]. Hence, in the disordered phase, we have $v_a = \sqrt{N}/N = 1/\sqrt{N}$ which explains why the higher the number of particles, the lower the order parameter is.

As the noise strength goes lower a certain threshold η_c , the system goes through a *spontaneous symmetry breaking phase transition* to finally reach an ordered phase. This is due

to the synchronization of the particles heading. To sum up, we can say that the noise has a really noxious effect on the collective motion and even more when the number of particles is high (at fixed density).

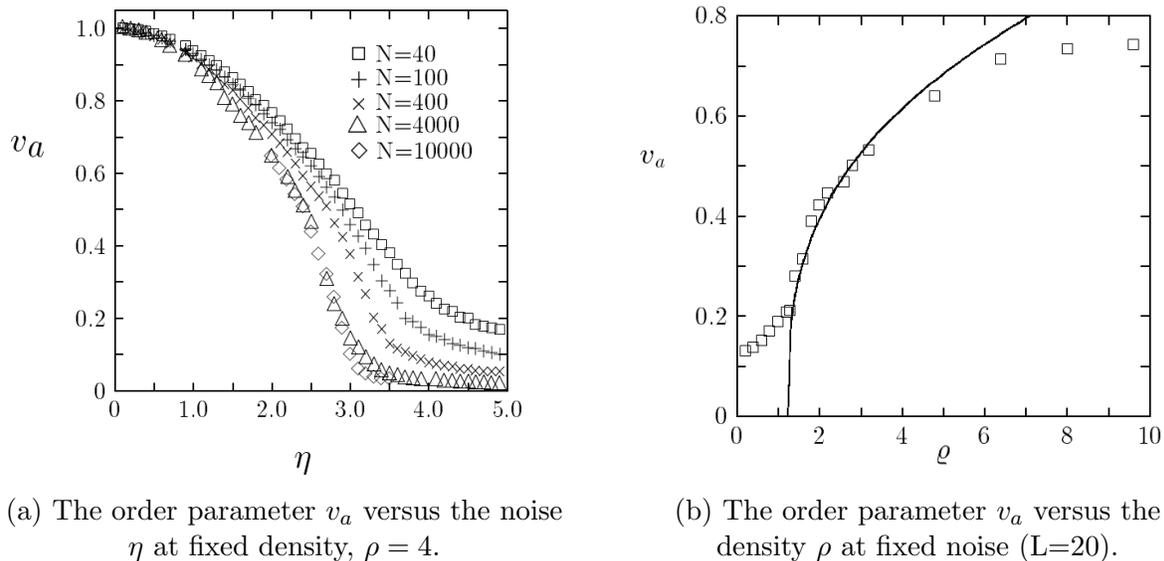


Figure 2.8: The order parameter v_a versus the noise on the left image and versus the density on the right image. Credits [9]

However, at a fixed level of noise, the higher the density, the better the order parameter is as it can be seen in Fig. 2.8b. Indeed, a higher density means more particles in a given neighborhood synchronizing their headings. The information sharing process, through the alignment rule, is thus more efficient which explains the trend of the curve. Once again it is possible to define a threshold ρ_c indicating the phase transition.

The phase transition is a big part a Vicsek's work and the analogy with the continuous phase transition of equilibrium systems is assumed in order to find the rate of the vanishing of the order parameter.

Because of its interesting characteristics, the Vicsek Model (noted VM) is still largely used to model physical systems such as ferromagnetic particles or bacteria. However, some other versions have also been developed in order to suit different situations like for example the paper of Grégoire et al. [45] which uses the VM with a Lennard-Jones-type body force for the attraction/repulsion interaction. This model is interesting when collisions are undesirable and cohesion is required. Another model inspired by Vicsek can be found in the work of Ginelli and Chaté [24] which considers topological interactions meaning that an agent will interact with the n_{topo} closest neighbors (for example). This method, motivated by observations of starling flocks [23], does not suffer from density fluctuations as the number of interactions is fixed. Vicsek himself also developed some variations of his model like for instance in the recent work of Jia and Vicsek [46] where a hierarchical flocking model is designed. It relies on leader-follower interactions which are hierarchical in the sense that

the leader and the followers don't have the same impact on each others. In this work, the stability of the model is investigated in the presence of noise which is quite similar to Vicsek's initial work.

Some of his articles have even led to real outdoor experiments such as the work of Vásárhelyi et al. [2] or the one of Virágh et al. [1] but we will not detail them in this section as they play a major part in this thesis and an entire section will be dedicated to them.

2.2.3 Olfati-Saber's Model

In this section, we are going to talk about the work of Olfati-Saber [10] which was released in 2006. It is probably one of the most general theoretical framework for design and analysis of distributed flocking algorithms. The article encompasses three algorithms, two for free-flocking and one for constrained flocking. Free-flocking refers to a flocking without obstacles while constrained flocking deals with obstacles. Among the free-flocking algorithms the first one results in multiple fragmentation³ while the second one does not. We will see in the following why this phenomena occurs and how it is solved. Even if this work appears to be the most accomplished of its kind (and its time), many related studies have been conducted and among them we can mention the work of Olfati-Saber and Murray [47], Fax and Murray [48] but also Tanner et al. [49, 26]. They take advantage of Olfati-Saber's framework and algorithms and extend them to suit their own problematic such as tracking a moving target [50].

In his paper, Olfati-Saber aims at preventing the two main pitfalls of flocking which are the fragmentation and the collapse while making simple hypothesis. It relies on a strong theoretical background including graph theory, stability analysis of dynamical systems and other mathematical tools. In the following paragraphs, we tried to extract the most relevant part of this work in function of our needs without losing consistency.

Consider N dynamic agents or particles which can be seen as nodes of a undirected graph \mathcal{G} . We have $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with node set $\mathcal{V} = \{1, 2, \dots, N\}$ representing the set of agents in the system and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ where edge $(i, j) \in \mathcal{E}$ represents a link between agent i and agent j . The adjacency matrix $A = [a_{ij}]$ of the graph is a matrix filled with elements satisfying the property $a_{ij} \neq 0 \iff (i, j) \in \mathcal{E}$ and we have $a_{ii} = 0$ for all i .

The N particles are ruled by the following equation of motion (double integrator):

$$\begin{cases} \dot{\mathbf{q}}_i = \mathbf{v}_i \\ \dot{\mathbf{v}}_i = \mathbf{u}_i \end{cases}, \quad (2.4)$$

where \mathbf{q}_i and \mathbf{v}_i represent respectively the position and the velocity of agent i and \mathbf{u}_i its control input.

³Disintegration of the flock into smaller groups.

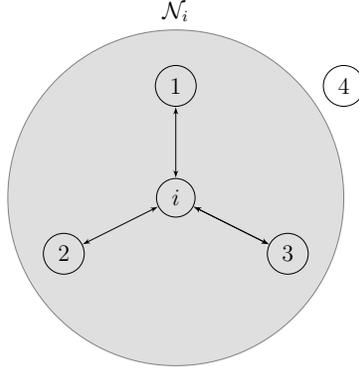


Figure 2.9: Agent i and its neighbors based on the definition of Eq. 3.9.

As the algorithm is not centralized, an interaction range r is considered to define the neighborhood of each agent. This neighborhood \mathcal{N}_i determines the set of agents an agent i is able to interact with:

$$\mathcal{N}_i = \{j \in \mathcal{V} : \|\mathbf{q}_j - \mathbf{q}_i\| \leq r\} \quad (2.5)$$

where $\|\cdot\|$ is the Euclidean norm.

Fig. 2.9 represents the neighborhood of agent i . We can see that $\|\mathbf{q}_4 - \mathbf{q}_i\| \geq r$ and hence agent 4 is not part of agent i neighborhood.

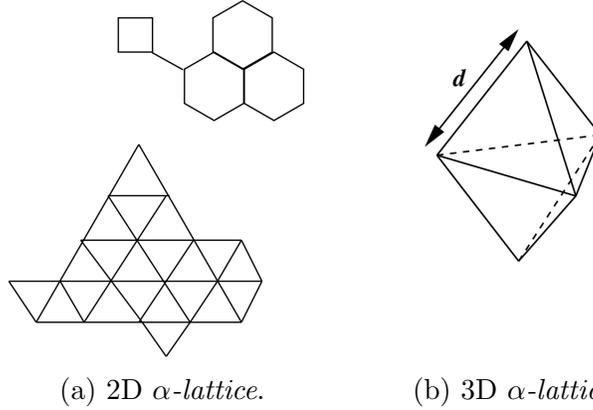


Figure 2.10: Representation of α -lattices. Credits [10]

To define the control input, the author uses *collective potential functions*. These potential functions are designed in a way that the configuration $\mathbf{q} = \text{col}(\mathbf{q}_1, \dots, \mathbf{q}_n)$ reached by the agents satisfies the following constraint:

$$\|\mathbf{q}_j - \mathbf{q}_i\| = d, \forall j \in \mathcal{N}_i \quad (2.6)$$

where d is the scale of the configuration. A configuration satisfying the constraints of Eq. 2.6 is known as an α -lattice. It is represented in 2D and 3D in Fig. 2.10

The idea behind the collective potential function is the relation between its local minimum and the solution of Eq. 2.6. Hence, by minimizing this function, we reach the desired configuration.

We will not detail all the formulation made by the author here but rather show through some graphs and explanation how it works. The collective potential function $V(q)$ is a non-negative function differentiable on its domain of definition:

$$V(\mathbf{q}) = \frac{1}{2} \sum_i \sum_{j \neq i} \psi_\alpha(\|\mathbf{q}_j - \mathbf{q}_i\|_\sigma) \quad (2.7)$$

with $\|\cdot\|_\sigma$ is the sigma norm⁴, differentiable everywhere and $\psi_\alpha(z)$ is a smooth attractive/repulsive pairwise potential with a finite cut-off at $r_\alpha = \|r\|_\sigma$ and a global minimum at $z = d_\alpha = \|d\|_\sigma$ (see [10]). In order to construct $\psi_\alpha(z)$, an action function $\phi_\alpha(z)$ that vanishes for all $z \geq r_\alpha$ is integrated:

$$\psi_\alpha(z) = \int_{d_\alpha}^z \phi_\alpha(s) ds, \quad (2.8)$$

$$\phi_\alpha(z) = \rho_h(z/r_\alpha) \phi(z - d_\alpha) \quad (2.9)$$

where $\rho_h(z)$ is a scalar function that smoothly varies between 0 and 1 and $\phi(z)$ is an uneven sigmoidal function with $\phi(0) = 0$.

The action and the potential functions are represented in Fig. 2.11.

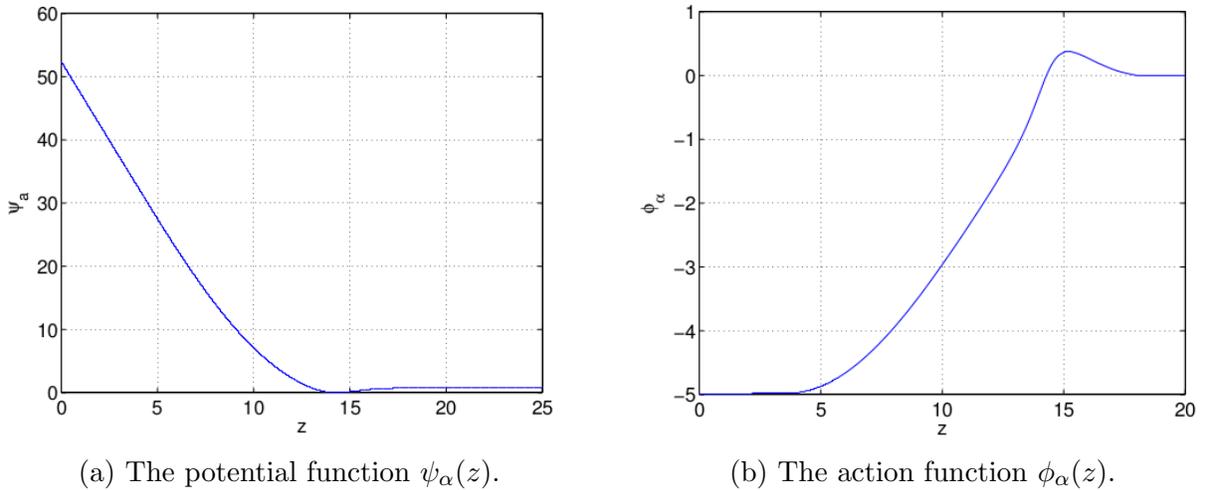


Figure 2.11: Plot of the potential and action functions. Credits [10]

⁴ $\|z\|_\sigma = \frac{1}{\epsilon}(\sqrt{1 + \epsilon\|z\|^2} - 1)$, $\epsilon > 0$

Now that most of the mathematical tools have been introduced, it is possible to present the different algorithms. In free-flocking the control input \mathbf{u}_i can be divided as follows:

$$\mathbf{u}_i = \mathbf{f}_i^g + \mathbf{f}_i^d + \mathbf{f}_i^\gamma \quad (2.10)$$

where $\mathbf{f}_i^g = -\nabla_{\mathbf{q}_i} V(\mathbf{q})$ is the gradient based term for attraction and repulsion, \mathbf{f}_i^d is the alignment term and \mathbf{f}_i^γ is the group objective.

Algorithm 1: The first algorithm does not include any navigational feedback (group objective) but still encompasses the three flocking interactions with the gradient based term for the attraction/repulsion and the weighted velocity difference for the alignment:

$$\mathbf{u}_i = \sum_{j \in \mathcal{N}_i} \phi_\alpha(\|\mathbf{q}_j - \mathbf{q}_i\|_\sigma) \mathbf{n}_{ij} + \sum_{j \in \mathcal{N}_i} a_{ij}(q) (\mathbf{v}_j - \mathbf{v}_i) \quad (2.11)$$

where \mathbf{n}_{ij} is a vector from agent i to agent j (with respect to the sigma-norm) and $a_{ij} = \rho_h(\|\mathbf{q}_j - \mathbf{q}_i\|_\sigma / r_\alpha)$ are the coefficient of the *spatial adjacency matrix* which represents the graph \mathcal{G} with distance-based coefficient.

Algorithm 2: The second algorithm is similar but includes the group objective \mathbf{f}_i^γ which combines a position regulation and a velocity consensus with regard to a *target* defined by its position \mathbf{q}_t and velocity \mathbf{v}_t :

$$\mathbf{f}_i^\gamma = -c_1(\mathbf{q}_i - \mathbf{q}_t) - c_2(\mathbf{v}_i - \mathbf{v}_t), c_1, c_2 > 0 \quad (2.12)$$

And the total control input is:

$$\mathbf{u}_i = \sum_{j \in \mathcal{N}_i} \phi_\alpha(\|\mathbf{q}_j - \mathbf{q}_i\|_\sigma) \mathbf{n}_{ij} + \sum_{j \in \mathcal{N}_i} a_{ij}(q) (\mathbf{v}_j - \mathbf{v}_i) + \mathbf{f}_i^\gamma \quad (2.13)$$

Even though these two algorithms combine all the rules defined by Reynolds [8] for flocking, the first algorithm leads to a flocking behavior only for a *very restricted set of initial states*. Without group objective, Algorithm 1 produces, most of the time, some fragmentation which is, as we mentioned above, one of the main pitfall of flocking (see Fig. 2.12). In comparison, Algorithm 2 produces a flocking behavior every time. However, the group objective is to use a centralized information that goes against the idea of a true flocking model (*i.e.*, with a decentralized control).

In his work, Olfati-Saber uses an agent-based terminology to identify the different actors at stake and their role. The first one is the α -agent which is basically applying Algorithm 1 in order to form an α -lattice with its neighbors. The second one is the γ -agent which appears in Algorithm 2 and can be dynamic (with a varying velocity) or static (with a null velocity). It has a role of target that α -agents aim for and is necessary for the emergence of a flocking behavior. The last one is the β -agent which is a virtual agent induced by the presence of a static obstacle close to an α -agent, and which must be avoided.

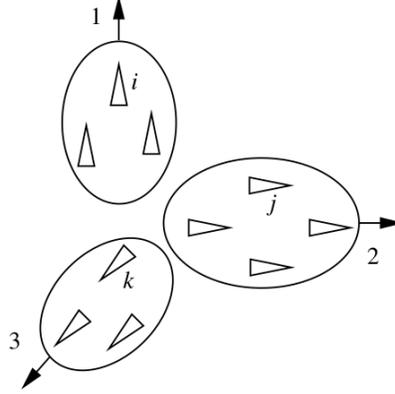


Figure 2.12: The fragmentation phenomenon. Credits: [10]

The β -agents are defined by their position $\hat{\mathbf{q}}_{i,k}$ which depends on the agent i projecting it on the closest boundary of a neighboring obstacle O_k and should satisfy the following constraint:

$$\hat{\mathbf{q}}_{i,k} = \operatorname{argmin}_{x \in O_k} \|\mathbf{x} - \mathbf{q}_i\| \quad (2.14)$$

and O_k is a convex object.

β -agents also have a velocity $\hat{\mathbf{v}}_{i,k}$, correlated to \mathbf{v}_i , which is tangent to the boundary it is placed on. This velocity is used to generate an alignment force with the α -agents but it doesn't mean that β -agents are moving on their own. An example of the projection of a β -agent on a spherical obstacle is represented in Fig. 2.13. In the same way that α -agents have a neighborhood to define agents they interact with, there is a neighborhood \mathcal{N}_i^β based on the set of obstacles $\mathcal{V}_\beta = \{1', \dots, l'\}$ which defines the set of β -agents an α -agent interact with:

$$\mathcal{N}_i^\beta = \{k \in \mathcal{V}_\beta : \|\hat{\mathbf{q}}_{i,k} - \mathbf{q}_i\| \leq r'\} \quad (2.15)$$

where r' is the interaction range with β -agents.

In order to interact with obstacles, a potential function $\phi_\beta(z)$ is used which aims at keeping α -agents away from β -agents from a certain distance d' . This constraint can be thought in the same way as Eq. 2.6:

$$\|\hat{\mathbf{q}}_{i,k} - \mathbf{q}_i\| = d', \forall k \in \mathcal{N}_i^\beta \quad (2.16)$$

Obviously, there is no need for attraction between agents and obstacles so potential $\phi_\beta(z)$ is designed in order to vanish directly after the repulsive range of obstacles:

$$\phi_\beta(z) = \rho_h(z/d_\beta)(\sigma_1(z - d_\beta) - 1) \quad (2.17)$$

with $\sigma_1 = z/\sqrt{1+z^2}$ and $d_\beta = \|d'\|_\sigma$.

Also there exists an associated *adjacency matrix* $B = [b_{i,k}]$ defining a relation between β -agents and α -agents:

$$b_{i,k}(q) = \rho_h(\|\hat{\mathbf{q}}_{i,k} - \mathbf{q}_i\|_\sigma/d_\beta) \quad (2.18)$$

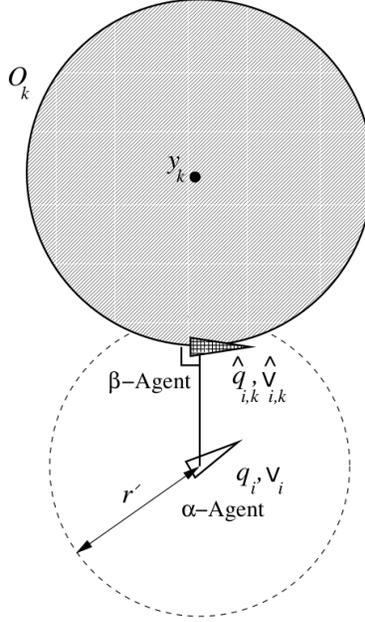


Figure 2.13: Projection of a β -agent on a spherical obstacle. Credits [10]

Algorithm 3: The global control input of Algorithm 3 is based on *multi-species interactions*. The first one is related to α -agents and includes attraction/repulsion and alignment. The second one deals with β -agents and includes repulsion and alignment for obstacle avoidance. Finally the last one is linked to γ -agents and includes attraction and alignment:

$$\mathbf{u}_i = \mathbf{u}_i^\alpha + \mathbf{u}_i^\beta + \mathbf{u}_i^\gamma, \quad (2.19)$$

with

$$\mathbf{u}_i^\alpha = c_1^\alpha \sum_{j \in \mathcal{N}_i} \phi_\alpha(\|\mathbf{q}_j - \mathbf{q}_i\|_\sigma) \mathbf{n}_{ij} + c_2^\alpha \sum_{j \in \mathcal{N}_i} a_{ij}(q) (\mathbf{v}_j - \mathbf{v}_i) \quad (2.20)$$

$$\mathbf{u}_i^\beta = c_1^\beta \sum_{k \in \mathcal{N}_i^\beta} \phi_\beta(\|\hat{\mathbf{q}}_{i,k} - \mathbf{q}_i\|_\sigma) \hat{\mathbf{n}}_{i,k} + c_2^\beta \sum_{k \in \mathcal{N}_i^\beta} b_{i,k}(q) (\hat{\mathbf{v}}_{i,k} - \mathbf{v}_i) \quad (2.21)$$

$$\mathbf{u}_i^\gamma = -c_1^\gamma \sigma_1(\mathbf{q}_i - \mathbf{q}_t) - c_2^\gamma (\mathbf{v}_i - \mathbf{v}_t) \quad (2.22)$$

where c_μ^ν are positive constants for all $\mu = 1, 2$ and $\nu = \alpha, \beta, \gamma$ and $\hat{\mathbf{n}}_{i,k}$ is a vector from i to $\hat{\mathbf{q}}_{i,k}$ with respect to the sigma-norm.

With this algorithm, the agents are swarming together while avoiding the potential obstacles on their path. As in Algorithm 2, the “objective” associated to γ -agents is necessary to avoid the fragmentation phenomenon. Besides its efficiency, this control algorithm gives rise to interesting maneuvers such as *squeezing* (going through narrow paths) or *split and join* (separation to avoid an obstacles before reaching back to a single cluster) as it can be seen in Fig. 2.14.

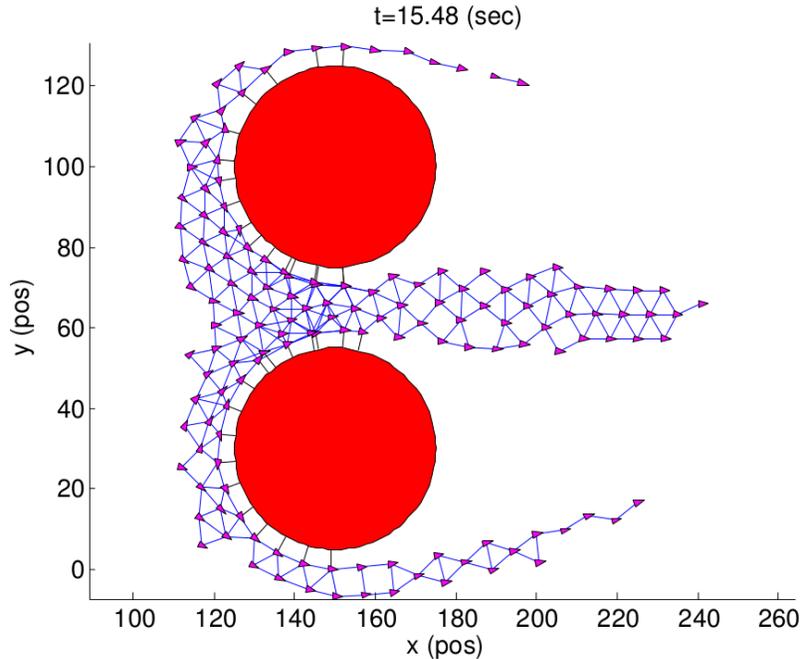


Figure 2.14: 150 agents squeezing themselves to go through highly constraining obstacles. Credits [10]

The only drawback of this model is the *conflict of tasks* which may occur in presence of non-convex obstacles. Indeed the group objective could force the agents to go toward a concave obstacle resulting in unwanted situations like for example the agents being stuck or even collisions with the obstacles.

The theoretical framework provided by Olfati-Saber is a fantastic tool for developing and analyzing distributed flocking algorithm. Each of the three algorithms presented have different performances and it is demonstrated that Algorithm 2 generically leads to a flocking behavior. For these reasons (but not only), this work has been repeatedly used in the literature such as in [50, 37, 51]. We also used it in one of the model developed in Chap. 4 (see also paper [52]) to extend its use to different contexts with other hypothesis.

2.2.4 Vászrhelyi et al.

A central issue of collective motion is addressed in the work of Vászrhelyi et al. [2], namely how to make large flocks of autonomous drones navigate in confined spaces. Through this work, the authors develop a flocking model for real UAVs which handles constrained environments with obstacles and walls as the swarm evolves in a closed space. It is inspired from the work of Vicsek [9], who is also author, but is also in the continuity of previous works of Vászrhelyi and Viragh [53, 1]. The model, based on self-propelling agents with position \mathbf{r}_i and velocity \mathbf{v}_i , defines a desired velocity $\tilde{\mathbf{v}}_i^d$ based on the flocking rules. It is a stochastic

model as it integrates several realistic parameters such as inaccuracy of onboard sensors, delays on the communication and reaction times.

At each time step, the desired velocity $\tilde{\mathbf{v}}_i^d$ of agent i is updated as follows:

$$\tilde{\mathbf{v}}_i^d = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} v^{flock} + \mathbf{v}_i^{rep} + \mathbf{v}_i^{frict} + \sum_s \mathbf{v}_{is}^{wall} + \sum_s \mathbf{v}_{is}^{obstacle} \quad (2.23)$$

where \mathbf{v}_i is the current velocity of agent i , v^{flock} is the preferred speed of the agents (related to the self propelling term), \mathbf{v}_i^{rep} is the repulsion term, \mathbf{v}_i^{frict} is the alignment term and \mathbf{v}_{is}^{wall} and $\mathbf{v}_{is}^{obstacle}$ are the obstacle and wall avoidance terms summed over all the sensed obstacles and walls of agent i . The final desired velocity \mathbf{v}_i^d is bounded to a maximum velocity v_{max} :

$$\mathbf{v}_i^d = \frac{\tilde{\mathbf{v}}_i^d}{\|\tilde{\mathbf{v}}_i^d\|} \min\{\|\tilde{\mathbf{v}}_i^d\|, v_{max}\} \quad (2.24)$$

The repulsion is a force derived from the half-spring model and occurs when the inter-agent distance, $r_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$, is smaller than the maximum interaction range r_0^{rep} :

$$\mathbf{v}_{ij}^{rep} = \begin{cases} p^{rep}(r_0^{rep} - r_{ij}) \frac{\mathbf{r}_i - \mathbf{r}_j}{r_{ij}}, & \text{if } r_{ij} < r_0^{rep} \\ 0, & \text{otherwise} \end{cases} \quad (2.25)$$

with p^{rep} a linear gain. The overall repulsion for agent i is $\mathbf{v}_i^{rep} = \sum_j \mathbf{v}_{ij}^{rep}$.

The alignment term is the centerpiece of this model. It acts as a damping medium, reducing noise-based oscillations and synchronizing motion among the agents. It scales to large velocity differences and takes into account the maximum allowed acceleration of the agents thanks to a smooth velocity decay function D :

$$D(r, a, p) = \begin{cases} 0 & \text{if } r < 0 \\ rp & \text{if } 0 < rp \leq a/p \\ \sqrt{2ar - a^2/p^2} & \text{otherwise} \end{cases} \quad (2.26)$$

where r is the distance between an agent and an expected stopping point, a is the preferred acceleration, and p is a linear gain also determining the crossover point between the two phases of deceleration. Thanks to this, the author defines the maximum allowed velocity difference between two agents $v_{ij}^{frictmax}$ and the resulting velocity alignment term \mathbf{v}_{ij}^{frict} :

$$v_{ij}^{frictmax} = \max\left(v^{frict}, D(r_{ij} - r_0^{frict}, a^{frict}, p^{frict})\right) \quad (2.27)$$

$$\mathbf{v}_{ij}^{frict} = \begin{cases} C^{frict}(v_{ij} - v_{ij}^{frictmax}) \frac{\mathbf{v}_j - \mathbf{v}_i}{v_{ij}}, & \text{if } v_{ij} > v_{ij}^{frictmax} \\ 0, & \text{otherwise} \end{cases} \quad (2.28)$$

where v^{frict} is a velocity slack, r_0^{frict} is the stopping distance, a^{frict} is the acceleration parameter, p^{frict} and C^{frict} are linear coefficients and $v_{ij} = \|\mathbf{v}_i - \mathbf{v}_j\|$. The overall alignment for agent i is $\mathbf{v}_i^{frict} = \sum_j \mathbf{v}_{ij}^{frict}$.

A visual explanation of the repulsion and the alignment terms is given in Fig. 2.15 where the blue line corresponds to the repulsion term and the green line defines the maximum allowed velocity difference between two agents. These terms are plotted in function of the inter-agent distance.

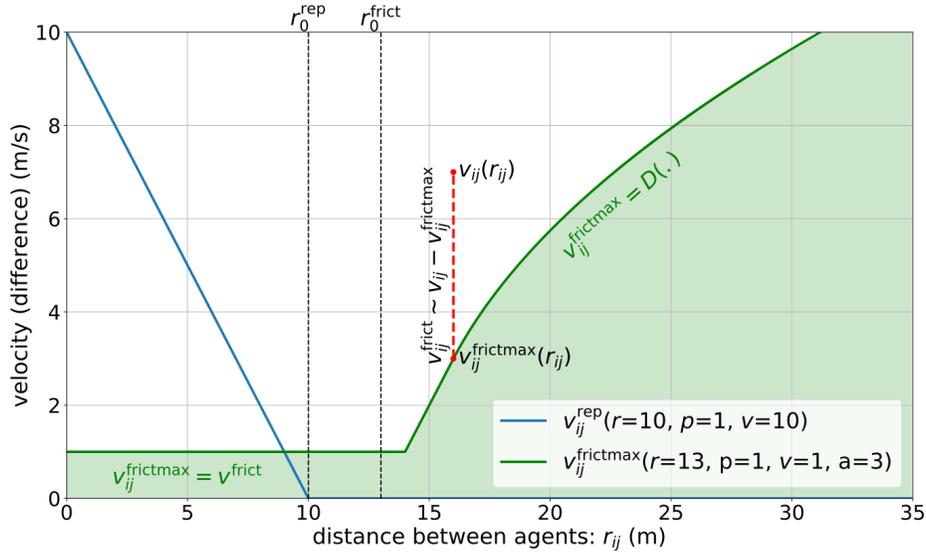


Figure 2.15: Visual interpretation of the repulsion term in blue and the alignment term given by the maximum allowed velocity difference between two agents $v_{ij}^{frictmax}$ in green. Credits[2]

The interactions with walls (boundaries of the environment the agents evolve in) and the obstacles are based on virtual agents called *shill agents*. In a similar fashion as it is defined by Olfati-Saber [10], each agent projects a *shill agent* on the closest point of nearby obstacles with a velocity \mathbf{v}_s pointing outward the obstacle (or inward for walls). Then the strategy used is to apply the alignment interaction term in order to relax its velocity to the one of the shill agent. Doing so, the agent is both repelled and slowed down. A maximum velocity difference $v_{is}^{shillmax}$ is also defined so as to reach the obstacle avoidance velocity term \mathbf{v}_{is}^{obst} :

$$v_{is}^{shillmax} = D(r_{is} - r_0^{shill}, a^{shill}, p^{shill}) \quad (2.29)$$

$$\mathbf{v}_{is}^{obst} = \begin{cases} (v_{is} - v_{is}^{shillmax}) \frac{\mathbf{v}_s - \mathbf{v}_i}{v_{is}}, & \text{if } v_{is} > v_{is}^{shillmax} \\ 0, & \text{otherwise} \end{cases} \quad (2.30)$$

where $r_{is} = \|\mathbf{r}_i - \mathbf{r}_s\|$, r_0^{shill} , a^{shill} and p^{shill} have the same signification that the one for alignment defined in Eq. 2.27 but with different values. $v_{is} = \|\mathbf{v}_i - \mathbf{v}_s\|$. The overall obstacle avoidance for agent i is $\mathbf{v}_i^{obst} = \sum_s \mathbf{v}_{ij}^{obst}$. A visual interpretation of an agent projecting its

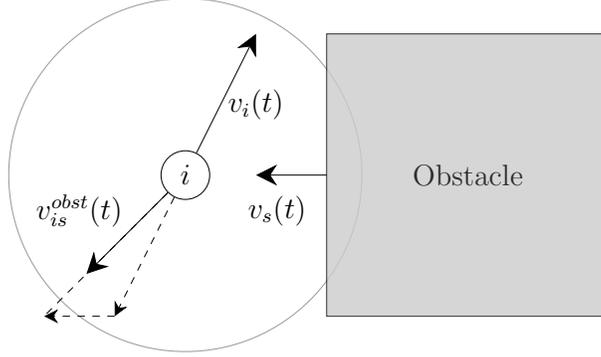


Figure 2.16: Agent i projecting a skill agent on an obstacle to perform the obstacle avoidance interaction.

skill agent on an obstacle is given in Fig. 2.16. The direction of \mathbf{v}_{is}^{obst} is also given with an arbitrary magnitude.

As it can be seen, this model concatenates many parameters that have to be tuned in order to reach an efficient flocking behavior. The authors use a genetic algorithm called CMA-ES [54] in order to take into account the stochastic characteristic of the model. This optimization relies on different order parameters (see Sec. 2.2.2) that measure the coherence, the obstacle avoidance and the collision avoidance during an experiment of time T . Regarding the coherence, it is associated to the correlation and is measured within connected clusters. Two agents i and j are said to be connected if their inter-distance is lower than $r^{cluster}$:

$$r^{cluster} = \max\left(r_0^{rep}, r_0^{frict} + \tilde{D}(v^{flock}, a^{frict}, p^{frict})\right) \quad (2.31)$$

where $\tilde{D}(v, a, p)$ is the breaking distance r for which $D(r, a, p) = v$.

Thus, consider N agents, the cluster \mathcal{J}_i of agent i of size N_i is defined as follows:

$$\mathcal{J}_i = \{j \in [1, \dots, N] : \|\mathbf{r}_j - \mathbf{r}_i\| \leq r^{cluster}\} \quad (2.32)$$

Hence, the cluster dependent correlation Ψ_{corr} , also known as alignment, can be defined:

$$\Psi_{corr} = \frac{1}{T} \frac{1}{N} \int_0^T \sum_{i=1}^N \frac{1}{N_i - 1} \sum_{j \in \mathcal{J}_i} \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|} dt \quad (2.33)$$

Regarding the collision, the collision risk Ψ_{col} is measured:

$$\Psi_{col} = \frac{1}{T} \frac{1}{N(N-1)} \int_0^T \sum_{i=1}^N \sum_{j \neq i} \Theta(r_{ij}(t) - r^{col}) dt, \quad (2.34)$$

with $\Theta(\cdot)$ the Heaviside step function and r^{col} the minimum allowed distance between two agents.

Potential collisions with walls and obstacles are measured by the order parameter Ψ_{wall} :

$$\Psi_{wall} = \frac{\int_0^T \sum_{i=1}^N \Theta(\tilde{r}_{is}(t)) \tilde{r}_{is}(t) dt}{\int_0^T \sum_{i=1}^N \Theta(\tilde{r}_{is}(t)) dt}, \quad (2.35)$$

where \tilde{r}_{is} takes positive values outside the arena and inside an obstacle, negative ones otherwise.

The average velocity of the flock Ψ_{vel} is also measured:

$$\Psi_{vel} = \frac{1}{T} \frac{1}{N} \int_0^T \sum_{i=1}^N \|\mathbf{v}_i(t)\| dt \quad (2.36)$$

The last metrics used regard the cluster size with Ψ_{disc} which counts the number of isolated agents (which cannot communicate with any other agent) and Ψ_{clust} which measures the minimum size of the clusters. More information about the optimization process will be given in Chap. 3.

The performance of this model is quite impressive and remains as is for a large velocity regime and a various number of agents. The outdoor experiments with real UAVs also confirm the effectiveness of the design of the control law with the integration of noises and delays. However these experiments also show the limit of the assumptions taken in terms of communication mostly. Indeed, in the model, the delay is constant and so is the communication range whereas in reality, they vary according to multiple factors such as the environment for the communication range or the distance between two drones for the delay. In the next section, we discuss the information flow in wireless sensor networks such as swarm of drones.

2.3 Information Flow in Swarm of UAVs

Collective motion requires an acute coordination among the involved entities. In nature, whether it is among flock of birds [55] or school of fish [56], this coordination is made possible through the perception of the animals which is based on vision mostly. When it comes to UAVs, different choices can be made. The improvements in image processing using neural networks for example have made cameras a very attractive sensor for swarm of drones [16] in order to derive neighbors positions. In addition, camera can also detect obstacles which is essential. However, cameras have drawbacks such as light condition sensitivity, computational cost/speed, directionality and environment in general which affects the line of sight. To overcome these limitations, wireless communication can be used as a complement [6] but also alone [1].

2.3.1 Exchanging information in flocking models

Let's first study how the information flow is realized in the flocking models discussed in Section 2.2. In Reynolds' boids [8], there is a strong analogy between the designed model

and flock of birds. Even though this latter relies on a limited perception of the environment through vision and hearing abilities, the available optical technology at the time this article was written was not efficient enough nor reliable to be considered in the model of the flocks. As a consequence, the information sharing relies on a communication-like process where all the information of the agents and the environment are centralized and distributed among all the agents (in the simulation). However, the information is not entirely global as it is shared only by agents in the same neighborhood (see Sec. 2.2.1).

The Vicsek Model [9] is purely simulation based and focused on the emergence of collective motion. It does not really consider the communication aspect and, like many others, only reckons that particles in a bounded circle interact together (see Sec 2.2.2). The authors make no assumption regarding the perception of the particles nor of the way of sharing information from a particle to another. They only assume that there is an underlying mechanism (based either on communications or perception) able to provide the necessary information to compute these interactions.

Olfati-Saber [10] is no exception. He considers a dynamic graph-based approach with a varying *communication complexity* which is related to the number of edges of the communication graph [57]. In this approach, the information flow through these edges is assumed local and perfect. The topology of a wireless sensor network with a bounded communication range is called a *net*. Because the communication range is the same for all the nodes, all *nets* are undirected graphs. No other aspects of the communication are studied in this paper. However, the author raises awareness about the computational costs if a centralized approach were foreseen for communications (for instance via a ground controller).

With Vásárhelyi's et al. [2], the communication aspect is much more studied and taken into account. Indeed the authors consider a local and delayed communication. The delay, even though it is constant, gives a better representation of realistic experiments which are mostly asynchronous. One main advantage of this work is that it has also conducted real outdoor experiment including two independent, parallel wireless modules for inter-agent communication in the 2.4-GHz range, both broadcasting the same status packets. This solution appeared to be unreliable as only 40% to 80% were received by close-by peers on average in a flight with 30 drones. In Fig. 2.17, we can see a detailed log of the distance dependence of the reception quality from a 5-min sample flight with 32 drones. We observe that the communication outages are more and more frequent as the inter-agent distance increases which means that communication is more reliable with close-by agents than with distant one. This distribution is actually not so inconvenient as critical information comes from the closest agents which do not suffer from extensive communication outages.

To conclude on the information flow in the flocking models, either there is no assumption on the way it is realized (like in the Vicsek model) or the authors assume perfect communications except in the work of Vásárhelyi's et al. in which a constant delay is added to the communications. Even if it is a first step towards a better modeling of the communication properties, it is still far from the reality. In this thesis, we go further on the communication aspect by introducing a realistic radio propagation model (see Chap. 3).

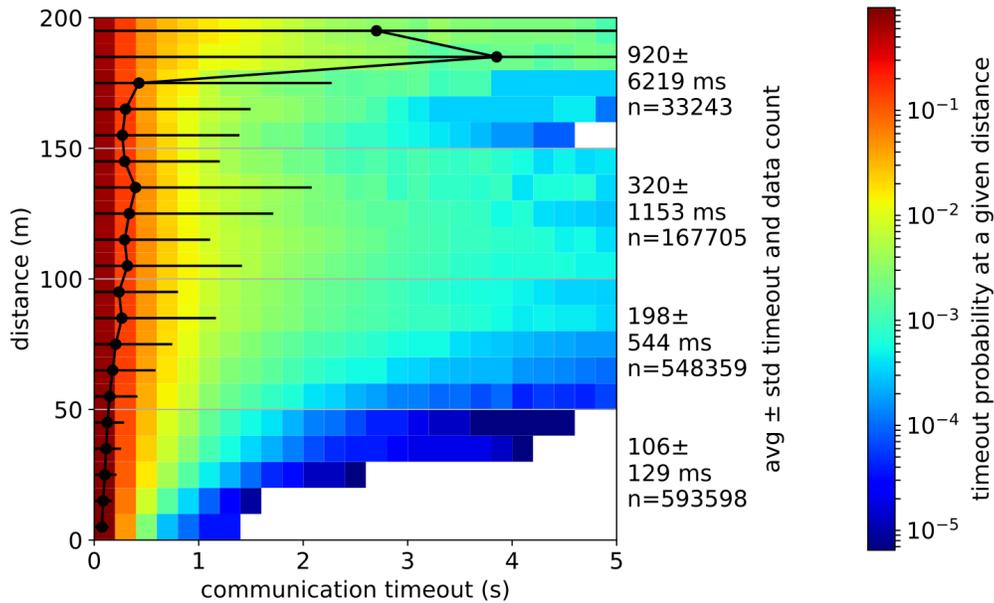


Figure 2.17: Probability distribution of the communication outages as a function of distance. The database was gathered from a 5-min section of a general flight with 32 drones in a remote open-air setting. Each drone logged a 5-Hz sampling of the elapsed time since the reception of the last status packet from all other drones. This value was matched later with the position of the drones recorded accurately by each drone onboard. Color indicates timeout probability in each bin for a given distance. Average timeouts with standard deviation and with the number of data points are indicated on the right for 50m distance gap, whereas the black line on the plot indicates average and std of timeout for each distance bin of 10m. [2]

2.3.2 The communication and mobility trade-off

Wireless communication is a very powerful tool as it enables the transmission of any kind of information that is either acquired through onboard sensors (*e.g.*, GNSS, camera, lidar) or also received from other nodes. Thanks to this, coordination among the agents can be more efficient than perception based sensors as the distribution of the information is possible. However, exchanging information between many nodes poses various challenges inherent to wireless communication. Different wireless communication technologies can be used and differ in terms of communication range, bandwidth and required infrastructure. Among them we can cite multiple papers that deal with swarm of drones using different protocols such as cellular network [18], Wi-Fi [2], Ultra-Wide-Band (UWB) [19] or even LoRa [20]. All these technologies have different performances and properties that may benefit a situation or another. For example, for a mission which requires long distance communication, UWB will most likely not be used in comparison to LoRa.

Independently of the protocol, communication relies on radio waves which are subject to various phenomena such as diffraction, absorption, ground effect, interference, noises and so on. Because of this, the performances of the communication can be degraded resulting in

losses, delays and even communication outages. The environment plays also a major role in the degradation of the communication. In the case of an urban environment, big obstacles such as buildings and houses generate unwanted phenomena like multipath which generates constructive or destructive interference [58]. On the other hand, dense environments like forests can give rise to multiple communication interruptions and noises.

Vicsek et al. show in [9] that the noise on measurements (for instance measurements of angle) prevents the emergence of a collective behavior. A degraded quality of the communications may also have harmful consequences on the swarm as shown in [59], in which the authors study, by simulation, how the swarm algorithms are vulnerable to degraded communication and confirmed by experimentation in [2].

In Chap. 4 and Chap. 5, we study the impact of environments with obstacles on different flocking models by using a radio propagation model that takes into account the obstacles.

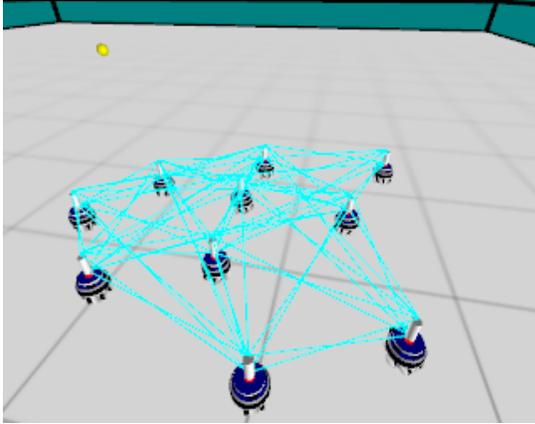
However, if degraded communication can prevent the emergence of a collective behavior, individual behaviors can themselves be adapted to provide a better communication quality between nodes as it is shown in [60] where the authors take advantage of the orientation of the UAVs to improve the communication throughput. Also in [61], the authors use the BER (Bit Error Rate) combined with the SNR (Signal to Noise Ratio) to characterize the communication quality and integrate it into a performance index expressing a trade-off between desired inter-agent distance and quality of the communication. In [62], the authors perform subterranean exploration based on signal measurements as the RSSI (Received Signal Strength Indicator) and adapt the path of the agents accordingly. From these different examples, we can see that the information flow in multi-agent systems should be carefully studied as it has a major role on the behavior of the swarm. However, modeling the information flow in multi-robot systems remains a challenge as they are dynamic, resulting in a high communication complexity.

2.3.3 Overview on communication in multi-robot simulators

When it comes to the study of swarm of UAVs communicating together, two types of simulator can be identified. The first one regards robots simulation, focusing on the accuracy of the motion including physical aspects and interactions with the environments. Among these simulators we can cite **GAZEBO** [11], **MORSE** [63] or **ARGoS** [64]. Snapshots of these simulators are displayed in Fig. 2.18 and 2.19.

Each of these simulators has its own characteristics, depending on the language it is written in but also the requirements of the team that made it. For example in **GAZEBO** and **MORSE**, the representation of the environment can be done “accurately” resulting in realistic simulations. However the computation cost increases accordingly which slows down the simulation and hence makes it difficult to simulate many robots. The **ARGoS** simulator is more focused on the simulation of large scale swarms of robots. This does not mean that the environment is not taken into account but that the rendering is simplistic.

Regarding the communication aspect, all these simulators integrate some functionalities that can be used to derive a “simple” communication model. In **ARGoS** and **MORSE**,



(a) ARGoS simulator [64].



(b) MORSE simulator [63].

Figure 2.18: Snapshots of different robots simulators.

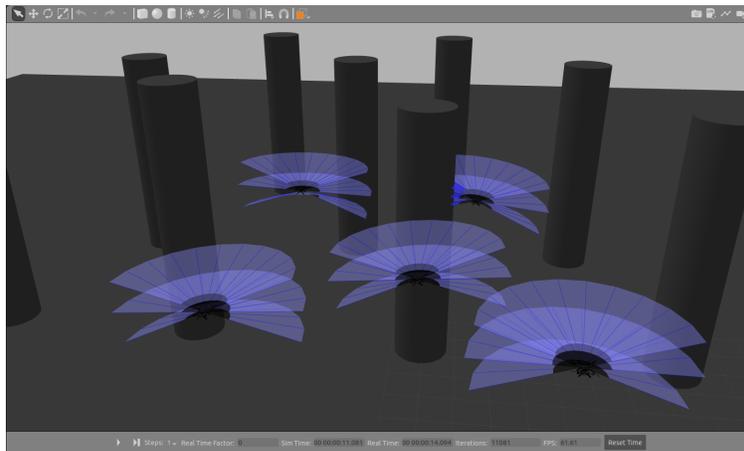


Figure 2.19: GAZEBO [11] (our flocking simulation)

the *range and bearing* model is implemented. It basically decides whether or not two robots can communicate depending on their inter-distance and their orientation. In **GAZEBO**, the robots are assumed to broadcast fixed-payload beacons transmitted at a fixed rate and frequency. These beacons can be received by any other robots in range itself computed using the *Okumura-Hata* model [65]. This model is interesting as it takes into account real communication characteristics but does not reflect the reality.

Accurately modeling wireless networks is a fully-fledged discipline that brings us to the second category of simulators namely network simulators. Among them we can cite **NS-3** [66] and **OMNeT++** [67]. These simulators are discrete event simulators that aim at accurately modeling networks from the physical layer to the application layer. In Fig. 2.20, we can see snapshots of these simulators. If the nodes in the network can be assimilated as robots, their movements, interactions and global dynamics will be very inaccurate as it is

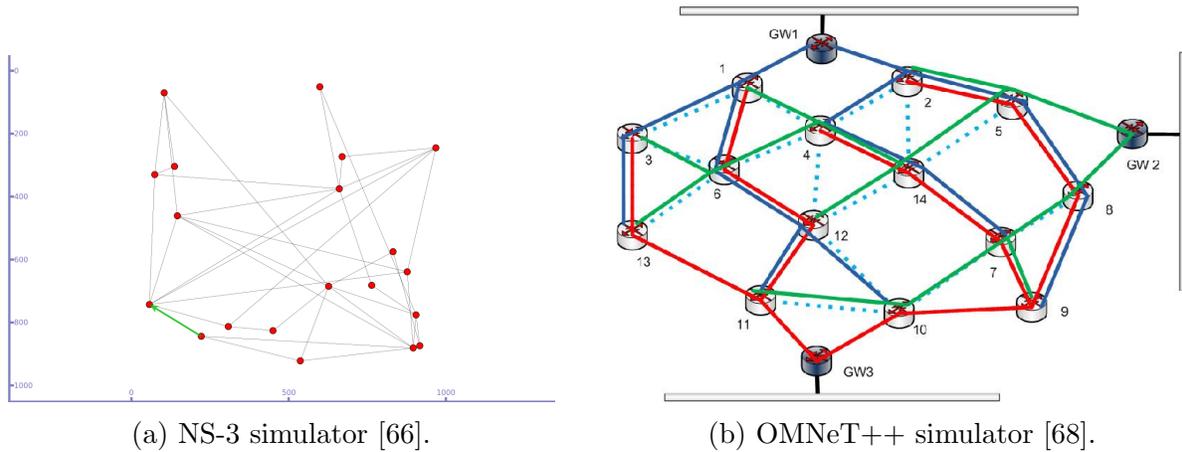


Figure 2.20: Snapshots of different network simulators.

not the purpose of these simulators and it would be too costly to integrate it.

Hybrid simulators have been developed such as **AVENS** [69] or **CUSCUS** [70] and act as a bridge between network and robotics by developing plugins in order to enable the communication between independent simulators such as **OMNeT++** and **X-Plane Flight Simulator** for the **AVENS** simulator. However, these hybrid simulators happened not to be maintained which make them not usable as the version of the used simulators are getting updated.

A recent simulator designed by Virágh et al. [1] has yet drawn our attention as it simulates the accurate motion of multiple coarse-grained (simplified) UAVs in complex environments. Moreover, it enables the modification of the control law and considers realistic communication aspects such as noises, delays and outages. We will see in the next chapter most properties of this simulator and also how we extended it with new functionalities.

As we have seen in Chap. 2 there is a lack of appropriate simulator merging both robot dynamics and realistic wireless communication. Researchers usually develop home made simulators that suit their needs in order to validate their control model before testing in the real world. In this chapter, we present the simulator of Viragh and how we extend it with new features. We focus on the robot model and introduce a radio propagation model in the simulator. We also present the settings used to define different scenarios. Finally, we introduce a framework in order to optimize flocking models based on performance metrics and using a genetic algorithm. The simulator and the optimization tools are closely related and will be regularly used throughout this thesis.

3.1 The Simulation Context

In this thesis, we need a simulator that satisfies multiple requirements in order to test new flocking models and optimize them in realistic conditions:

- Simulation of multiple UAVs using coarse-grained representation (to be scalable to large swarms)
- Modifiable UAVs control law
- Simulation of realistic communication
- Simulation of various physical environments with or without obstacles
- Simulation in 2D or 3D
- Low computational cost for scalability and fast simulation/optimization

Following these requirements, we decided to use and extend the simulator¹ developed by Virágh et al. [1]. It is a dynamic multi-agent simulator with “low level” rendering and is written in C. It is capable of solving stochastic differential equations (SDE) using *Euler-Maruyama* method. This property is fundamental as the dynamic of our system is not deterministic. Indeed, multiple noises defined by random variables are introduced in our model for the sake of realism. In addition to this property, we decided to use this simulator instead of off the shelf simulators such as ARGoS [64] or GAZEBO [11] for two main reasons. The first one is that the original model of Vásárhelyi et al. was already implemented in this simulator and hence would give us a fair comparison with our models. The second one, and probably the most important, is that we mostly use this simulator on servers to perform parameters optimization which requires hundreds of runs and this tool has a low computational cost speeding up the optimization process. The visualization is only used to check the behavior of the agents for a given set of parameters.

Two snapshots of the main window is displayed in Fig. 3.1. On Fig. 3.1a, we can see a 2D environment filled with a bounded arena in blue and obstacles in orange. The agents are represented by the yellow dots in the center of the window. In addition, some supplementary information is also given such as the number of clusters based on the communication graph (top left corner), a length scale (top left corner), the elapsed time since the beginning of the simulation (bottom left corner) and the speed of the simulation in comparison to the real clock of the computer (bottom left). It is also possible to have a 3D view of the environment as it can be seen on Fig. 3.1b. Other features are available in this visualization mode and will be presented throughout this thesis when they are used.

The simulator can be divided in three parts. The first one concerns the aerial robot model, focusing on its properties and motion. The second one regards the communication model, that is to say how we assume the information to be shared among the UAVs. The third one

¹Access to the original project here: <https://github.com/csviragh/robotsim>

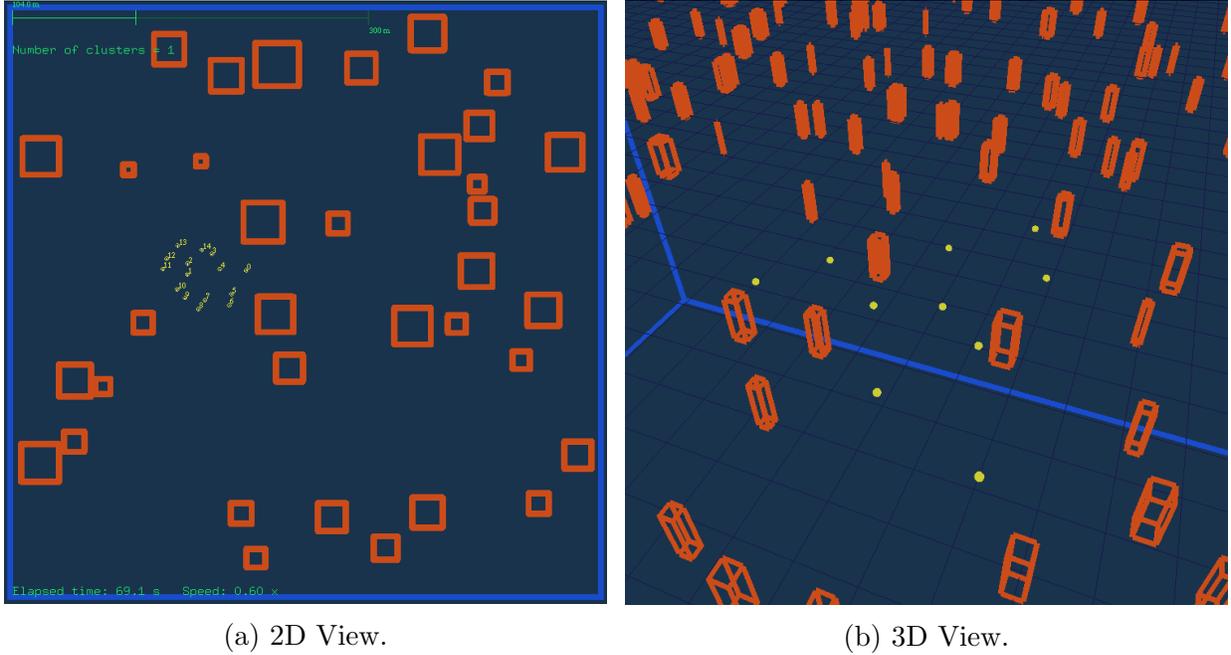


Figure 3.1: Main window of the simulator displaying the agents evolving in the arena.

is dedicated to the configuration of the simulator including the physical environment, the initial conditions and the different functionalities. Finally, the performance analysis based on custom metrics integrated to the simulator will be discussed and introduced along with the optimization process which is a crucial step of our work.

3.2 Aerial Robot Model

In this section we present the main features of the the aerial robot model proposed in Virágh’s simulator as we are going to use it for this thesis.

UAVs usually have a low level controller, like a PID controller [71] for instance, taking as input a desired velocity or acceleration or torque and so on. In this simulator, a velocity controller is used and the desired velocity of a unit i , \mathbf{v}_i^d depends on the position \mathbf{x}_j and velocity \mathbf{v}_j of its neighbors j :

$$\mathbf{v}_i^d(t) = \mathbf{f}_i(\mathbf{x}_j(t), \mathbf{v}_j(t)) \quad (3.1)$$

where \mathbf{f}_i is the “control law” which corresponds to our flocking algorithm. In Chap. 2 we have seen different examples for the design of \mathbf{f}_i . In Chap. 4 and Chap. 5 we will present new flocking algorithms.

For the sake of realism, various flaws are introduced by Virágh in the aerial robot model. If they cannot really express the deficiencies of real robots, they still raise awareness regarding the impact of each of them on the behavior of the robots independently and as a swarm:

- **Inertia:** The desired velocity \mathbf{v}_i^d is not instantaneously reached. It depends on a low-level controller, as we mentioned above, itself having a settling time τ_{ctrl} . This characteristic time defines the time it takes for the velocity input to be reached. However we cannot expect the robot to reach any kind of velocity in a fixed time, independently of its initial condition. This condition is directly linked to the inertia of the robot or acceleration, which magnitude is hence bounded to a given value a_{max} . The velocity is also bounded to v_{max} .
- **Inner noise:** Gaussian noises are introduced into the position and velocity measurements from the GNSS. It is represented by a stochastic function $\boldsymbol{\eta}_i^s(t)$ characterized by a standard deviation σ_s . For more information, please refer to [1].
- **Outer noise:** A Gaussian noise $\boldsymbol{\eta}_i(t)$ with standard deviation σ is also added to the acceleration of the robots accounting for the unpredictable effects of the environment like the wind for example.
- **Refresh rate of sensors:** Each robot updates its sensors at a given frequency f_s . The higher this frequency, the more reactive the robot will be.
- **Processing delay:** We assume that there is a fixed time t_{del} representing the delay between the acquisition and the treatment of neighboring data. This makes perfect sense when multiple robots exchange information simultaneously at high frequency.

All these parameters can be modified on-line which is a big advantage of this simulator as one can see the impact of each of them. They should be carefully studied and their value should be eventually set to a realistic one. For our experiments, we used the values gathered in Tab. 3.1. These values come from [2] and can be fairly associated to a realistic hardware. For example, a typical refresh rate for a GNSS sensor varies between 1Hz and 10Hz.

Table 3.1: Parameters of the robot model

Parameter	Description	Value
τ_{ctrl} (s)	Settling time of the low-level controller	$\tau_{ctrl} = 1s$
a_{max} (m/s^2)	Maximum acceleration in magnitude	$a_{max} = 6m/s^2$
v_{max} (m/s)	Maximum velocity in magnitude	$a_{max} = 7m/s$
σ_s (m^2/s^2)	Inner noise standard deviation	$\sigma_s = 0.005m^2/s^2$
σ (m^2/s^3)	Outer noise standard deviation	$\sigma = 0.2m^2/s^3$
f_s (Hz)	Refresh rate of sensors	$f_s = 5Hz$
t_{del} (s)	Processing time of the data	$t_{del} = 0.2s$

The control law \mathbf{f}_i should lead to equivalent performances whenever some parameters of the aerial robot model are changed in a reasonable range, especially regarding the various noises introduced as well as the refresh rate of the sensors and the processing delay. Indeed,

depending on the hardware used, these parameters might be different but we expect the performances of our model to remain constant.

Finally, the calculation of the desired acceleration of the UAV is done by computing the error between the actual velocity of the UAV and the desired one. Some noise is also introduced for the sake of realism. This variation of velocity per period of time (proper to the low level controller) is equivalent to the acceleration $\mathbf{a}_i(t)$ that the UAV should have at each time step. As we also set a maximum acceleration a_{max} and an outer noise $\boldsymbol{\eta}_i(t)$, this should also be considered in the equation (as done in [1]):

$$\mathbf{a}_i(t) = \boldsymbol{\eta}_i(t) + \frac{\mathbf{v}_i^d(t) - \mathbf{v}_i(t) - \mathbf{v}_i^s(t)}{\|\mathbf{v}_i^d(t) - \mathbf{v}_i(t) - \mathbf{v}_i^s(t)\|} \cdot \min \left\{ \frac{\mathbf{v}_i^d(t) - \mathbf{v}_i(t) - \mathbf{v}_i^s(t)}{\tau_{ctrl}}, a_{max} \right\}, \quad (3.2)$$

where $\mathbf{v}_i^d(t)$ is the desired velocity from 3.1, $\mathbf{v}_i(t)$ is the velocity of the UAV and $\mathbf{v}_i^s(t)$ is the velocity noise derived from the the inner noise by integrating it, $\dot{\mathbf{v}}_i^s(t) = \boldsymbol{\eta}_i(t)$. Thanks to the *Euler* method, the equation above can be solved and the resulting velocity and position of the robot can be calculated.

The aerial robot model is a fundamental layer of the simulator, the closer it reflects reality the more reliable it is towards real experiments. However, it often means increasing the complexity and the computational cost.

3.3 Simulating Degraded Communication by Obstacles

Another important aspect of the simulator is the communication layer. Most of the flocking simulations consider ideal communications and do not take into account the impact of the environment on the communication quality which usually results in issues when performing real experiments [2]. A majority of the simulations uses a distance threshold under which the communication is possible and above which it is not and the communication is either perfect [10, 50], or more realistic (with delays and packet losses) [2, 1]. However, even this latter has limitations as we explained in Sec. 2.3. In [59], the authors explain how an emergent behavior, such as flocking, is impacted when we use a relatively accurate propagation model and tune the inherent losses due to shadowing effects, SNR, jamming and so on. However, these studies consider only simplistic flocking models, far from realistic fleet of UAVs.

Whenever outdoor experiments with swarm of UAVs are being conducted, different technologies can be used to perform air-to-air communication. Among them we usually find **Wi-Fi** [2, 58], **Ultra Wide Band** [6], **XBee** [72] and cellular network communication technologies [73]. These different technologies have their own properties that make them more suitable depending on the application and its requirements. One feature of the IEEE 802.11 standard, also known as Wi-Fi, is the ability to integrate portable and moving stations without the use of any pre-existing infrastructure, which is exactly what we seek in this work. It is also a reliable technology for short communication distance ($< 100m$), making it very suitable for a flocking application where the UAVs are expected to move in a tight formation. Note also that Wi-Fi interfaces are small and light, impacting, at the margin,

the drones' payload. For these reasons, we have decided to consider the Wi-Fi technology in our simulator.

As in the robot model, we aim at having a realistic model for the communication layer. Even though we cannot accurately simulate all the radio waves from an emitter to a receiver because of the computational cost, we still want to introduce some features that are fundamental in the communication:

- **Communication protocol:** Each robot is equipped with an isotropic antenna and broadcasts its information using the Wi-Fi protocol on the 2.4GHz band. We consider that each robot has an emitting power of $P_{Tx_{dBm}} = 20dBm$ and that the communication threshold is $P_{min} = -70dBm$. Under this power, the reception is no longer possible.
- **Communication range:** The communication between two robots is local and bounded to a given area. This area is not constant and depends on the surrounding environment of the robot. Indeed, obstacles usually hinder the propagation of the radio waves and hence prevent the information flow. This dynamic range $cr(t)$ is computed using the *log distance path loss* model. The detailed presentation of this model will be done in Sec. 3.3.1.
- **Communication loss:** All communication systems are subject to packet losses. This phenomena is more likely to happen when the communicating robots are far from each other or when the received power $P_{Rx_{dBm}}$ (in terms of radio wave) is low. To model this loss we define a packet loss ratio $pl^r \in [0, 1]$ and a packet loss power pl^p representing the minimum received power below which losses are most likely to occur. At each time step we compute the quadratic loss coefficient inspired from [1], $K = \frac{P_{Rx_{dBm}} * P_{Tx_{dBm}}}{pl^p * pl^p} * pl^r$ and randomly pick a number $\zeta \in [0, 1]$. If $\zeta < K$, then we consider that the transmitted packets have too many losses and hence the communication link is broken. Otherwise the communication takes place. In [1], the coefficient is computed using distances but in our case, we use the received power to fit our communication model.

These parameters both define the characteristics of the communication protocol we use but also the different flaws we introduce in this protocol. The values used in the simulation are gathered in Tab. 3.2.

Table 3.2: Parameters of the communication model

Parameter	Description	Value
f_{prot} (GHz)	Frequency used for the communication protocol	$f_{prot} = 2.4GHz$
$P_{Tx_{dBm}}$ (dBm)	Power level of the emitter	$P_{Tx_{dBm}} = 20dBm$
P_{min} (dBm)	Minimum level of power before communication outage	$P_{min} = -70dBm$
pl^r (-)	Packet loss ratio	$pl^r = 0.3$
pl^p (dBm)	Packet loss power	$pl^p = -65dBm$

The use of powers instead of distances is directly linked to the fact that we use an actual radio wave propagation model for the communication. This model is integrated to the simulator and depending on its output, we know whether or not the communication between two UAVs is possible or not.

In the following, we introduce a prediction step of the radio propagation using the *log distance path loss model* (denoted LDPL) for both free space propagation but also inside obstacles and we see how obstructed environments affect the quality of the communications and consequently the flocking quality.

3.3.1 Log distance path loss model

In this section, we aim at modeling how radio waves dynamically evolve with the environment. The goal is to define the resulting *communication range* which sets the boundaries of one agent neighborhood. To represent this communication range we will use the path loss PL which can be defined as follows:

$$PL = P_{Tx_{dBm}} - P_{Rx_{dBm}}, \quad (3.3)$$

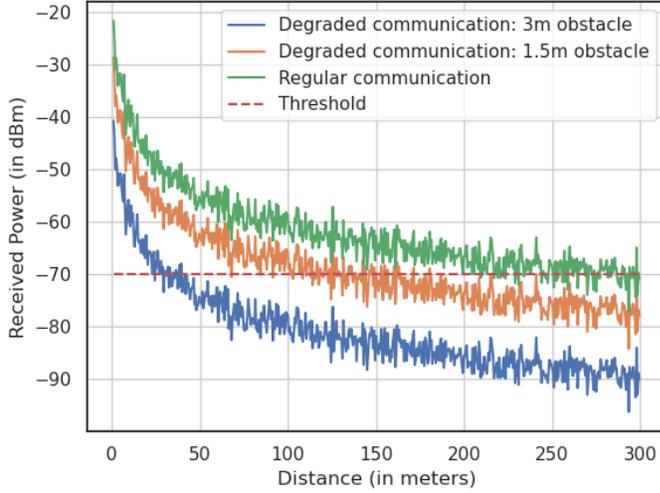
where PL is the path loss measured in decibels (dB), $P_{Tx_{dBm}}$ is the transmitted power in dBm at the emitter, $P_{Rx_{dBm}}$ is the received power in dBm at the receiver. As we need to estimate the power received by the UAVs knowing the transmitted power, we need a method to compute the path loss and thus retrieve $P_{Rx_{dBm}}$. To do so, we consider the LDPL model that predicts the path loss a signal encounters within different types of environments [74]. The LDPL model has been repeatedly used in the literature to measure the path loss between the different nodes of a “flying ad-hoc networks” (FANETs) [75] like in [58, 20] but estimating the real path loss exponent of the environment may be difficult and out of the scope of our research [76]. Other models can also be used for the radio wave propagation [77], but most of them are very specific to a given environment or set-up. Even if the LDPL model is “simple” it is generic and that is why we decided to use it. Last but not least, modeling radio waves accurately is extremely costly especially when the transceivers are moving (in our case, the UAVs).

The LDPL model is formulated as follows:

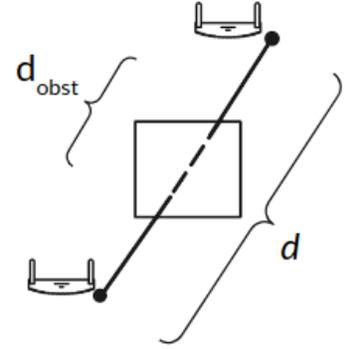
$$PL = \begin{cases} PL_0 + 10\gamma \log_{10} \frac{d}{d_0} + X_g, & \text{if } d \geq d_0 \\ PL_0, & \text{otherwise} \end{cases} \quad (3.4)$$

where, PL_0 is the path loss at the reference distance d_0 calculated using the Friis free-space path loss model, d is the distance between the emitter and the receiver, γ is the path loss exponent that depends on the environment (we use $\gamma = 2$ in free space) and X_g reflects the variations of the path loss caused by shadowing effects and/or multiple paths. This latter is a zero mean Gaussian random variable with a standard deviation of $\sigma_c = 2dB$. For the sake of clarity, we will refer to this communication model as **regular**.

The green curve in Figure 3.2a represents the evolution of $P_{Rx_{dBm}}$ on a path where there is no obstacle and $P_{Tx_{dBm}} = 20dBm$. This corresponds to the situation represented in Fig. 3.2b with $d_{obst} = 0m$. We can see that it takes approximately 300 meters to reach the power threshold $P_{min} = -70dBm$ which is often considered as the minimum power required to enable a “reliable” communication between agents, in Wi-Fi for instance.



(a) Evolution of the received power as a function of inter-agent distance. The green line represents a path without obstacle, the orange line has a 1.5 meters long obstacle on its path and the blue one has a 3m long obstacle on its path.



(b) Schema representing two agents separated by a distance d including a portion of length d_{obst} within an obstacle.

Figure 3.2: Received power for a given scenario with varying size of obstacle.

In the previous formula, the environment is modeled via, among others, the parameter γ and X_g . In order to take into account the impact of the different obstacles on the communication quality, we add a dissipation model inside the obstacles leading to high loss of the signal strength when the signal goes through an obstacle. The resulting path loss is expressed as follows:

$$PL = \begin{cases} PL_0 + 10\gamma \log_{10}\left(\frac{d - d_{obst}}{d_0}\right) & \text{if } d - d_{obst} \geq d_0 \\ + X_g + PL_{obst}(d_{obst}), & \\ PL_0 + PL_{obst}(d_{obst}), & \text{otherwise} \end{cases} \quad (3.5)$$

where, d_{obst} is the length of the path inside the obstacle and PL_{obst} is the path loss in dB due to obstacle attenuation. By using a simplified model of the path loss, this latter is defined

as:

$$PL_{obst}(d_{obst}) = 10\gamma_{obst}\log_{10}(d_{obst}) + K \quad (3.6)$$

where γ_{obst} is the path loss exponent for the obstacle and K is a constant. We assume that $d_{obst} \geq 1m$. This communication model will be called **degraded**. All the parameters of the communication model are gathered in Tab. 3.3.

Table 3.3: Parameters of the LDPL model

Parameter	Description	Value
γ (-)	Path loss exponent in free space	$\gamma = 2$
γ_{obst} (-)	Path loss exponent in the obstacles	$\gamma_{obst} = 4$
σ_c (dB)	Standard variation of the random variable X_g	$\sigma_c = 2dB$
d_0 (m)	Reference distance	$d_0 = 1m$

The orange and the blue curves in Figure 3.2a represent the evolution of $P_{Rx_{dBm}}$ on a path where respectively $d_{obst} = 1.5m$ and $d_{obst} = 3m$ (see Fig. 3.2b). In this configuration we have $\gamma_{obst} = 4$, $P_{Tx_{dBm}} = 20dBm$ and $K = 0$. We can see that this model has a strong impact on the communication's quality as it takes less than 30 meters to first lose the communication among agents with a 3 meters obstacle between them.

We can now define a new function $cr(t)$ representing the communication range by finding the inverse function of the path loss and setting the received power $P_{Rx_{dBm}}$ to the minimum level P_{min} . We consider the case here where $d - d_{obst} \geq d_0$ as we are looking for the maximum distance before the communication is no longer reliable.

So we can write:

$$P_{Tx_{dBm}} - PL > -70$$

$$P_{Tx_{dBm}} - (PL_0 + 10\gamma\log_{10}(\frac{d - d_{obst}}{d_0}) + X_g + PL_{obst}(d_{obst})) > -70$$

$$\log_{10}(\frac{d - d_{obst}}{d_0}) < \frac{70 + P_{Tx_{dBm}} - PL_0 - PL_{obst}(d_{obst}) - X_g}{10 * \gamma} = K(d_{obst}, X_g)$$

and finally we get

$$d < d_0 * 10^{K(d_{obst}, X_g)} + d_{obst}$$

Hence we have:

$$cr(t) = d_0 * 10^{K(d_{obst}(t), X_g(t))} + d_{obst}(t) \quad (3.7)$$

Using this formula to compute the communication range for $d_{obst} = 0$ and $X_g = 0$, we get $cr = 316m$ which correlates with our results plotted on Fig. 3.2a.

With this method, we are now capable of predicting the agents that are able to receive information that has been broadcasted by another agent. This is fundamental in flocking models where the interactions are local.

3.3.2 Neighborhood definition

The notion of **neighborhood** is central in flocking. Indeed, it commonly represents the set of agents whose states (position and velocity mostly) can be accessed by another agent, through communication or sensors such as camera, in order to compute its next move. If we go back to Eq. 3.1, we said that the desired velocity of one agent i depends on the position and velocity of its neighbors j and was calculated using a control f_i . Thanks to the LDPL model that we presented in the previous section (Sec. 3.3.1), we are now capable of defining the actual neighbors implicated in the calculation. Hence, the control law can be seen as the sum of all the local interactions with each neighbor:

$$\mathbf{f}_i = \sum_{j \in \mathcal{N}_i} \mathbf{f}_{ij}, \quad (3.8)$$

where \mathbf{f}_{ij} is the interaction term between agents i and j and \mathcal{N}_i is the **neighborhood** of agent i . In the following, we present two neighborhoods that will be used in this thesis.

To define the neighborhood properly, we use some graph notations. We model a multi-agent system by a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with node set $\mathcal{V} = [1, 2, \dots, N]$ representing the set of agents in the system and the directed edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ where edge $(i, j) \in \mathcal{E}$ represents a directed link of agent i sending information to agent j . The opposite is not always true, $(i, j) \in \mathcal{E} \not\Rightarrow (j, i) \in \mathcal{E}$. We consider that the edge (i, j) exists (*i.e.* agent j received the information from agent i) if and only if $j \in \mathcal{N}_i$ with \mathcal{N}_i being the **communication neighborhood (CN)** of agent i :

$$\mathcal{N}_i = \{j \in \mathcal{V} : \|\mathbf{r}_j - \mathbf{r}_i\| \leq cr(t)\} \quad (3.9)$$

where $cr(t)$ is the communication range which varies according to the LDPL model (see Eq. 3.7), \mathbf{r}_j and \mathbf{r}_i are the positions of agents j and i . From this definition, we understand why our graph is directed. Indeed the communication range is not deterministic and its value for agent i might be smaller for agent j .

In Fig. 3.3 we can see the representation of the graph \mathcal{G} formed by 4 agents with their respective communication neighborhood (gray circles) and the edge set \mathcal{E} represented by the black arrows. We use dashed arrows to represent an uncertain communication link between agents 2 and 4 as they are separated by an obstacle.

As in [25, 24], we also consider a topological neighborhood. We call it the **active neighborhood (AN)**, defined by the set of agents with whom an agent i interacts, *i.e.* those whose information will be received and exploited by agent i . We express this active neighborhood \mathcal{N}_i^+ as follows:

$$\mathcal{N}_i^+ = \{j \in \mathcal{V} : i \in \mathcal{N}_j\} \quad (3.10)$$

\mathcal{N}_i^+ is also bounded by a maximum size N_{max} in order to limit the number of agents that interact with agent i . \mathcal{N}_i^+ becomes:

$$\mathcal{N}_i^+ = \begin{cases} \mathcal{N}_i^+, & \text{if } |\mathcal{N}_i^+| \leq N_{max} + 1 \\ \mathcal{N}_i^{filtered}, & \text{otherwise} \end{cases} \quad (3.11)$$

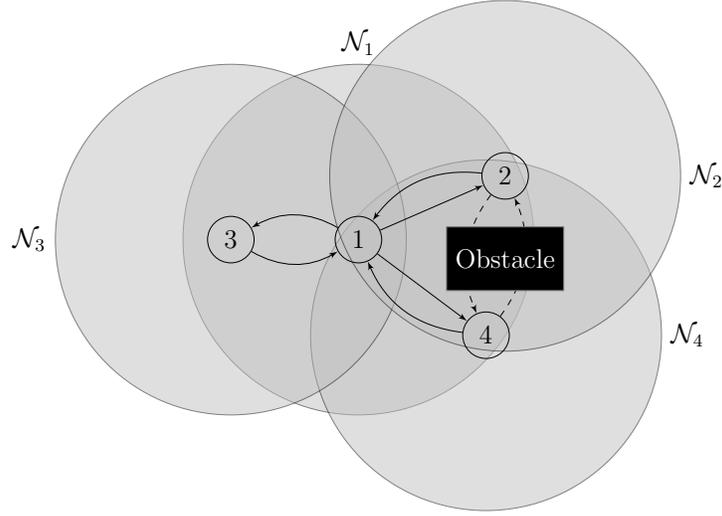


Figure 3.3: Representation of the directed graph with 4 nodes and their respective communication neighborhood (CN) represented by the gray circles. The full arrows represent an effective communication link while the dashed arrows represent an uncertain communication as there is an obstacle in the middle.

where $\mathcal{N}_i^{filtered}$ corresponds to a subset of \mathcal{N}_i^+ where the neighbors with the highest path loss are removed until $|\mathcal{N}_i^+| = N_{max} + 1$. Note that the set is defined such that agent i is always part of its own neighborhood, so **when $N_{max} = 1$ it means that agent i has an AN of size 2 including its closest neighbor and itself.** If two agents i and j have the same AN, we write $\mathcal{N}_{i,j}^+$ to express the equality between \mathcal{N}_i^+ and \mathcal{N}_j^+ . We also note \mathcal{E}^+ the edge set containing the different directed edges induced by all the ANs associated to the different agents.

As an illustration, Fig. 3.4 shows the resulting graph representation of ANs (represented by the gray ellipses) derived from the original graph representation of Fig. 3.3. For this example, we consider $N_{max} = 1$, which means that each agent will exploit the information of its closest neighbor. The resulting edge set \mathcal{E}^+ is represented by the black arrows. We can see that agents 1 and 3 have the same AN and will hence exploit each others information only ($\mathcal{N}_1^+ = \{1, 3\}$, $\mathcal{N}_3^+ = \{3, 1\}$). Agents 2 and 4 both exploit agent 1 information.

Thanks to this definition of the neighborhood, we will be able to master the scalability of the models proposed in this thesis. Indeed, no matter the number of agents in the flock, the number of interactions per agent will never be higher than N_{max} . In addition, we expect to interact with the agents that have the lowest path loss and hence that have a more reliable communication. Hence, we can rewrite Eq. 3.8 using the **active neighborhood**:

$$\mathbf{f}_i = \sum_{j \in \mathcal{N}_i^+} \mathbf{f}_{ij}, \quad (3.12)$$

where \mathbf{f}_{ij} is the interaction term between agents i and j .

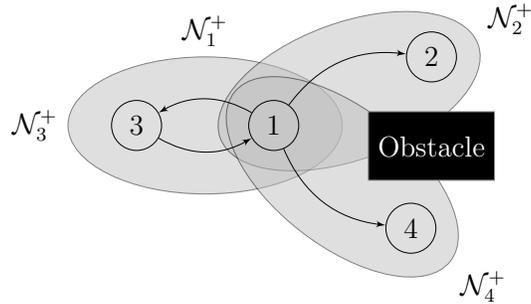


Figure 3.4: Representation of the active neighborhood with 4 agents. Here $N_{max} = 1$, *i.e.* each agent exploits only information from its closest neighbor (black directed edges correspond to these data exchanges). Gray ellipses illustrate the topological neighborhood, that we call the active neighborhood (AN).

3.4 Environments and Settings

In this section, we talk about the configuration layer of the simulator. It mostly relates to all the constant parameters and files that are fed to the simulator before the beginning of an experiment such as the environment or the initial conditions but also the desired outputs.

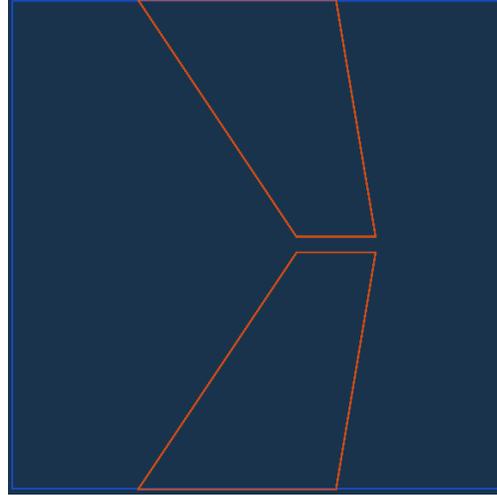
3.4.1 The physical environment

In this simulator, the environment is always composed of an arena and some obstacles. The arena prevents the agents from moving too far away from a designated area. It has the shape of a square or a circle (in 2D) and is defined by its *radius* R_{arena} , equivalent to half the size of the square side (if the arena is a square). The obstacles are represented by polygons and usually quadrilaterals. The flocking algorithm and especially the obstacle avoidance feature is designed for convex obstacles. Even if it will not fail to avoid concave obstacles, undesired behaviors might arise such as collisions or fragmentation with agents being stuck inside concave obstacles. Each obstacle is defined by its vertices that are themselves defined by their coordinates. In addition, obstacles can be rotated by a given angle which will be fixed during all the simulation. This information is gathered in a file which is then fed to the simulator, taking care of properly drawing “walls” between the vertices. We can see an example of the obstacle file and the result in the simulator in Fig. 3.5. The simulator uses a Cartesian coordinate system with an origin on the center of the arena. Because all this work is done in 2D, we usually do not consider the third component of all the simulated objects, it is automatically set to zero. Working in 2D comes from the fact that it is the most constraining configuration as we remove one degree of freedom. Consequently, if our model works properly in 2D, we expect it to work in 3D as well. Also, in most application, the aerial robots usually remain at the same altitude. And finally, a 3D configuration can be seen as multiple 2D layers stacked together.

```

1 [init]
2
3 angle=0
4
5 [obstacles]
6
7
8 obst0.point=-30000 -62000
9 obst0.point=10000 -2000
10 obst0.point=30000 -2000
11 obst0.point=20000 -62000
12
13
14
15 obst1.point=-30000 62000
16 obst1.point=10000 2000
17 obst1.point=30000 2000
18 obst1.point=20000 62000
--

```



(a) Example of an obstacle file. (b) Corresponding environment in the simulator.

Figure 3.5: Obstacle file and its equivalent in the simulator.

Building the obstacle file from scratch might seem easy at first but it clearly depends on the environment we need. In Vásárhelyi’s et al. work [2], the obstacle distribution is not a priority and it is not even clear what file is being used for the simulations. In this thesis, obstacles have a major role as they have a big impact on the communication, as we saw in the previous section, and hence on the behavior of the swarm. Because of this, we have to define several rules and metrics in order to automatize the obstacle file generation. As a result, the rules are the following:

- The obstacles should be convex.
- Obstacles must not intersect.
- Obstacles must not cross the boundaries of the arena.

In order to simplify the first rule, we actually force all the obstacles to be squares. Regarding the second rule, it is obvious that it can rapidly become costly and sometimes impossible to compute intersections when the number of obstacles N_{obst} is high or when the desired size of the obstacle side R_{obst} is big. The third rule can actually be concatenated to the second rule if we see the walls of the arena as obstacles.

Thanks to these rules, we can build an algorithm that will automatically generate the obstacle file. The last elements needed for this are the parameters of the environments. Given the desired number of obstacles N_{obst} , our algorithm proposes two ways of generating the obstacles. The first one relies a length metric where the length of an obstacle side $\mathcal{L}(obst)$ is calculated according to a continuous random variable \mathcal{X}_{obst} with mean R_{obst} and standard deviation σ_{obst} , $\mathcal{L}(obst) \rightarrow \mathcal{X}_{obst}(R_{obst}, \sigma_{obst})$. Depending on the value of σ_{obst} the obstacles are quite similar but there is still some variability. The second option is based on the *density*

\mathcal{D} which gives the proportion of the obstacles area over the arena one:

$$\mathcal{D} = 100 \times \frac{\sum_{s=1}^{N_{obst}} A_s}{A_{arena}}, \quad (3.13)$$

where A_s is the area of obstacle s and A_{arena} is the arena area. With this method, the user can set the desired density and the algorithm will automatically generate N_{obst} obstacles with various dimensions in order to fit the density. It appears this solution was not the best as the algorithm would usually generate big obstacles at first almost matching the desired density and then small ones to adjust it. As a consequence, we prefer the first solution but still compute the resulting density.

The position of the obstacles in the arena is also done according a random variable \mathcal{P} . It is either a uniform distribution, or a normal distribution. The uniform distribution is more intuitive and will place obstacles all over the arena whereas the normal distribution will generate a very scattered area with a low density on its surroundings. Even if this can be used for some case scenarios we will prefer the uniform distribution.

All the parameters of an environment are gathered in Tab. 3.4. These parameters will be used to define the environments we design in the following chapters.

Table 3.4: Parameters of the environment

Parameter	Description
R_{arena} (m)	Radius of the arena
N_{obst} (-)	Number of obstacles in the environment
R_{obst} (m)	Average length of an obstacle
σ_{obst} (m)	Standard deviation of an obstacle length
\mathcal{D} (%)	Density of obstacles in the arena
\mathcal{P} (-)	Distribution of the obstacles in the arena

3.4.2 Initial conditions and configuration of an experiment

Initial conditions are very important as they represent the starting point of an experiment. Combined with an identical setting of the simulator, they ensure the repeatability of an experiment and hence enable us to properly compare different models.

An experiment is usually defined by its duration T and the number of UAVs N . The initial positions and velocities of the simulated agents depend on the scenario but in our case we always set the initial velocity to zero. Regarding the positions, we usually initialize the positions inside a circle in the middle of the arena. The positions are yet not deterministic as we introduce some randomness in them. This is done to avoid a potential “overfitting” when repeating an experiment several times for optimization for example. In addition, the distance between two UAVs should not be less than R_{col} otherwise we consider that there is a

collision. This is true during all the experiment. Another crucial parameter is the accuracy of the *Euler* method Δ_t . As a reminder, this method is used to solve the differential equations and Δ_t can be seen as the time step between two computations. For example, given the acceleration $\mathbf{a}_i(t)$ and the velocity $\mathbf{v}_i(t)$ of agent i at time t , the velocity at time $t + \Delta_t$ is computed as follows:

$$\mathbf{v}_i(t + \Delta_t) = \mathbf{v}_i(t) + \mathbf{a}_i(t) * \Delta_t \quad (3.14)$$

As a consequence, the smaller Δ_t is, the more accurate the simulation is. However, this value has a big impact on the speed of simulation.

All the parameters mentioned above are gathered in Tab. 3.5. We have filled the value column whenever the parameter remains unchanged for all the experiment we have conducted in this thesis.

Table 3.5: Parameters of an experiment

Parameter	Description	Value
T (s)	Duration of an experiment	$T = 600s$
N (-)	Number of simulated UAVs	To be defined
R_{col} (m)	Distance below which collision occurs	$R_{col} = 3m$
Δ_t (s)	Accuracy of the Euler method	$\Delta_t = 0.01s$

An important characteristic of this simulator is that it has two modes. In the first one, the duration of the experiment is infinite. It is used to visually validate a model and observe the behavior of the swarm. The second mode has no visual and runs only during the duration T . Removing the visual ensures a lower computational cost. It is used when the user wants to get some output files related to the performances of the model for a given experiment. There are multiple outputs available in the simulator and thanks to its versatility, it is easy to add a new one. The desired outputs must be chosen in a configuration file as well as their format. There are three different formats:

- Timeline: at each time step Δ_t the output is saved into the file;
- Stat: the file contains the average of the output over the whole duration T of the experiment;
- Steadystate: the file contains the average of the output during the time interval $[t_0, T]$ where t_0 is the beginning of the steady state. Indeed, one may want to get rid of the first measures as they could skew the average measure depending on the initialization and the duration of the experiment.

In the next section, we will talk about these outputs and their related metrics. They are fundamental in the optimization process and enable us to numerically validate a model.

3.5 Metrics and Optimization Process

The flocking models we are going to present in the next chapters concatenate multiple parameters that locally act on a given interaction term but also globally on the sum of the interactions. For these reasons, it is quite difficult and most certainly not optimal to tune the parameters by hand. This is why an optimization strategy is necessary. It involves two major steps which are the evaluation of the performances of a model for a given experiment based on custom metrics and the optimization process taking the performances as inputs. These two steps are now detailed in following sections.

3.5.1 Measuring the performances of the flocking

The performances of a flocking model are closely related to the efficiency of the collective navigation, which can be measured by the desired interactions between the UAVs. Hence, we can measure the average distance between the agents, their alignment, the number of collisions and so on. In Sec. 2.2.4 we have presented several metrics from [2] that we will also use in this thesis. In Tab. 3.6 we remind these metrics and their description. For the formulas of these metrics, please refer to Sec. 2.2.4. Some new metrics will also be defined in the following chapters as they depend on new elements and their usage will be specified accordingly.

Table 3.6: Metrics for the performances of a flocking model

Metric	Description
Ψ_{corr} (-)	Alignment of the agents within a cluster
Ψ_{col} (-)	Collision ratio among agents
Ψ_{wall} (-)	Collision ratio with obstacles
Ψ_{vel} (m/s)	Average velocity of the swarm
Ψ_{disc} (-)	Agents that cannot communicate with any other agent
Ψ_{clust} (-)	Minimum size of a cluster

In order to “normalize” these metrics, we use different functions inspired from [2]. The goal of these functions is to map the metrics between 0 and 1, 0 being the worst value and 1 the best. The mapping is non-linear so as to weight some values accordingly with their impact on the model. For example, as we do not want any collision, the function should attribute 1 to the input 0 and then decrease rapidly to 0 for the remaining values.

The first function corresponds to this previous example where we need a strong constraint around 0:

$$\mathcal{F}_1(x, \alpha) = \frac{\alpha^2}{(x + \alpha)^2} \quad (3.15)$$

where α is a parameter that modifies the sharpness of the peak around 0.

The second function is less constraining than the previous one and is inspired from the probability density function of a Gaussian distribution. It is maximum at 0 and then smoothly decays:

$$\mathcal{F}_2(x, \beta) = \exp\left(-\frac{x^2}{\beta^2}\right) \quad (3.16)$$

where β is a parameter related to the inflection point of the function.

Finally the last function can be seen as a sigmoid where the inflection points can be tuned in order to adapt the penalization with regards to the metric being measured:

$$\mathcal{F}_3(x, l, \kappa) = \begin{cases} 0 & \text{if } x < l - \kappa, \\ \frac{1}{2} \left(\cos\left(\frac{\pi}{\kappa}(x - l)\right) + 1 \right) & \text{if } l - \kappa < x < l, \\ 1 & \text{otherwise} \end{cases} \quad (3.17)$$

We also underline the Heaviside step function as it is used as a mapping function:

$$\Theta(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{otherwise} \end{cases} \quad (3.18)$$

The visualization of these functions for a given set of parameters is displayed in Fig. 3.6. The Heaviside step function is not represented as it is equal to 1 on the domain we set.

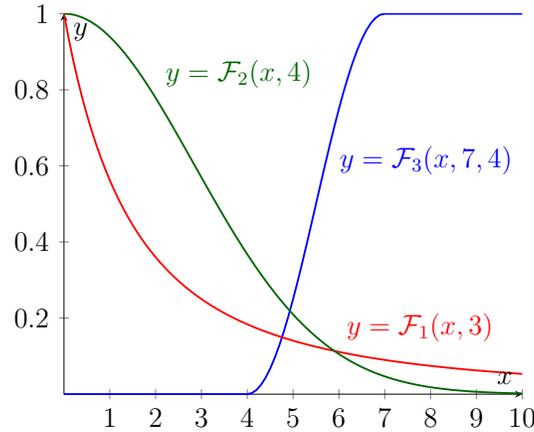


Figure 3.6: Representation of the mapping functions with $\alpha = 3$, $\beta = 4$ and $l = 7$, $\kappa = 3$.

Thanks to these functions, we can eventually define the fitness functions related to each metric. We took inspiration from the work of Vásárhelyi et al. but the mapping function associated to one metric can change depending on the constraints we need (the parameters as well).

In Tab. 3.7 we give the formulas associated to each fitness function. δ_{col} and δ_{wall} are the

Table 3.7: Fitness functions for the performances of a flocking model

Fitness function	Formula
F_{corr} (-)	$\Theta(\Psi_{corr}) \cdot \Psi_{corr}$
F_{col} (-)	$\mathcal{F}_1(\Psi_{col}, \delta_{col})$
F_{wall} (-)	$\mathcal{F}_2(\Psi_{wall}, \delta_{wall})$
F_{vel} (-)	$\mathcal{F}_3(\Psi_{vel}, v^{flock}, \delta_{vel})$
F_{disc} (-)	$\mathcal{F}_1(\Psi_{disc}, N/5)$
F_{clust} (-)	$\mathcal{F}_3(\Psi_{clust}, N/5, N/5)$

tolerance values for the collision among agents and with the walls. The tolerance for the collisions should be as low as possible in order to penalize the least occurrence but cannot be null because of the definition of \mathcal{F}_1 and \mathcal{F}_2 . v^{flock} is the self propelling speed, δ_{vel} is the tolerance for the speed and N is the number of agents. These fitness functions are likely to change throughout this thesis.

3.5.2 Optimization based on a genetic algorithm

Thanks to the fitness functions defined in the previous section, we now have a way to measure the performances of a flocking model. This is fundamental in an optimization process as it mostly relies on comparing different solutions. In addition, we need our optimization algorithm to satisfy multiple requirements:

- Adapted to multi-objective problems corresponding to the different fitness functions;
- Adapted to multi-dimensional problems (the dimension of a solution is always > 10 in our case);
- Handle non-deterministic experiments due to the multiple noises introduced in the model (see Eq. 3.2).

These requirements are actually not mandatory in order to find a good solution of the problem. Indeed, in a related work [2], the authors used an evolutionary algorithm called **CMA-ES** [54]. Even though it works with stochastic models with many parameters, it is a single objective algorithm. To cope with this, the authors design a single objective optimization scenario by multiplying all the fitness functions together. The resulting “cost function” is very harsh and whenever a fitness function has a bad result, it greatly affects the overall performance. The authors of [2] still manage to find appropriate solution to the problem but the computational cost as well as the time complexity are not negligible. For these reasons we decided not to use the same algorithm. We want to have full control over the fitness functions independently and for example prioritize one fitness over another one.

The optimization process we present is based on a recent genetic algorithm called the **NSGA-III** [78]. This is a fast non-dominated sorting genetic algorithm designed for multi-objective problems. The selection process consists of sorting an initial population, filled with N_{pop} solutions, using a non-dominated sorting algorithm. The resulting solutions are then compared with regards to their orthogonal distance to some “reference directions” defined during the initialization. Those elite solutions are then used to generate an offspring population of size N_{pop} using various operations such as *crossover* and *mutation*. These operations are repeated for a given number of *generations* N_{gen} or until the performances match a desired value or during a given time. An important point about this algorithm is that it outputs a Pareto front. Hence, there is not always one optimal solution at the end of the process but rather a front with multiple solutions on the space of the fitness functions. Choosing one solution is then an empirical work but also a question of experience. For example, whenever multiple optimal solutions arise, we apply a prioritization over the different metrics in the following order:

1. F_{col} : Collisions among UAVs are the number one priority. A solution should always generate 0 collision otherwise it is a bad solution.
2. F_{wall} : Collisions with obstacles are also a major concern. However, as we are not working on the obstacle collision avoidance model, we may accept solutions generating few collisions with obstacles and then tune manually the parameters to get rid of the collisions. This is why experience is also important here.
3. F_{disc} : In this thesis, we seek no fragmentation phenomenon. Hence, avoiding isolated agents is very important.
4. F_{clust} : Preventing fragmentation also means maximizing the size of the clusters. But we would rather have two separated clusters than one big cluster with one isolated agent.
5. F_{vel} : Reaching the desired speed should not be neglected as motion is the key to the emergence of a collective behavior.
6. F_{corr} : Alignment is a major interaction in flocking and it should have a significant role in our prioritization. However it does not impact the safety of the swarm as F_{col} and F_{wall} and it appears that whenever F_{clust} is high, so is the alignment. This is due to the fact that agents within a cluster will always tend to align themselves and thus increase F_{corr} .

NSGA-III is actually an extension of a previous algorithm, the **NSGA-II** [79]. If the two algorithms seem to have the same performances for problems with not more than two objective functions [80], the **NSGA-III** is yet much more efficient with many objective functions. Hence we decided to use it. It is yet not suited for stochastic models which is one of our requirement. This point is actually very important as the experiments are always different for a given set of parameters because of the initial conditions or the randomness

in the sensors measurements. In order to take into account this stochastic characteristic, we decided to introduce a new fitness function F_{rob} corresponding to the *robustness* of the model. It computes the norm of the vector filled with the standard deviation of the previously defined fitness functions derived from a finite number of experiments N_{rob} with the same set of parameters.

$$F_{rob} = \sqrt{\sigma_{F_{corr}}^2 + \sigma_{F_{col}}^2 + \sigma_{F_{wall}}^2 + \sigma_{F_{vel}}^2 + \sigma_{F_{disc}}^2 + \sigma_{F_{clust}}^2} \quad (3.19)$$

where σ_{F_x} is the standard deviation of the F_x fitness function after running an experiment N_{rob} times with the same solution. Thanks to this new fitness function, we are now capable of “measuring” the stochastic aspect of the model for a given solution. The lower this function is, the more robust the solution is. In Fig. 3.7, we can see a graph that will be encountered

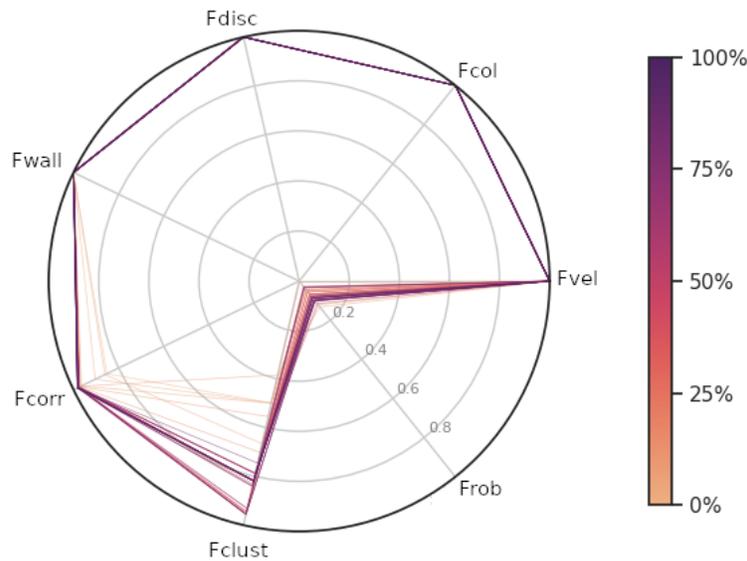
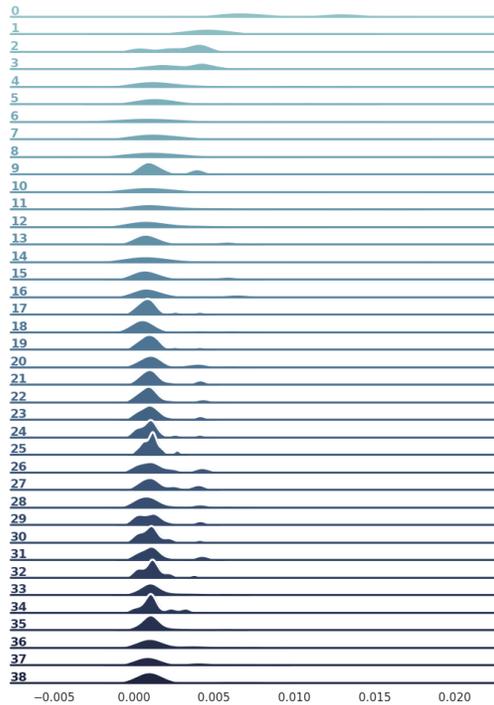
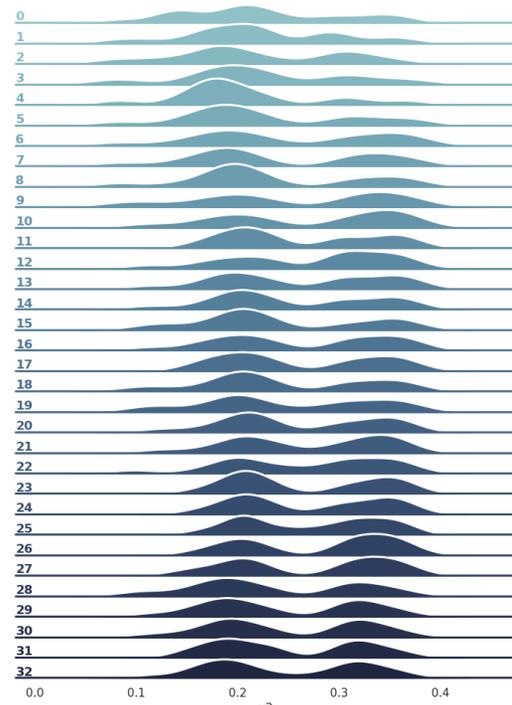


Figure 3.7: Evolution of the fitness functions during the optimization.

several times in this thesis. It shows the evolution of the value of the fitness functions at each *generation* represented by a line. Because we may have multiple solutions in our optimal Pareto front, the value on the line is actually the average of all these solutions. Another visualization of the optimization is represented in Fig. 3.8 where we can see two images describing the evolution of a parameter of the model. At each *generation*, the distribution is computed by the occurrences of the parameter found in the elite solutions. On the left, we can see the evolution of the attraction gain. It seems that the optimization does not modify the parameter that much which means that it is a good candidate for the solution. On the other hand, on the right, it appears that two “clusters” of solution predominate. This means that there is no optimal value for this parameter but does not mean that the performances will be different. The whole solution must be seen as a whole. This visual representation has a great interest in the design of the initial population of solution and also the search space for



(a) Evolution of the attraction gain during the optimization.



(b) Evolution of the repulsion gain during the optimization.

Figure 3.8: Evolution of two flocking parameters during the optimization.

each parameter. Indeed we can see the values that are being tried by the algorithm, through *mutation* mostly, and see if they are pertinent or not. For example, if a cluster of value is generated at the border of the search space, it means that we should probably extend the boundary on this side. Tuning the initial population and the search space is a fundamental work that has a major impact on the convergence speed of the algorithm.

All the details of the **NSGA-III** algorithm can be found in [78] and [79]. Thanks to its versatility and the framework developed by Blank and Deb [81], we managed to combine both our simulator and the genetic algorithm.

In this chapter, we focus on two reference models which are the one of Vásárhelyi et al. [2] and the one of Olfati-Saber [10]. We show the limits of these models and how we modify and combine them in order to have better performances. So far, the models we presented in chapter 2 (Sec. 2.2) are, in their respective framework, efficient and lead to good performances or at least lead to valuable conclusions regarding some properties of the flocking. However, it is sometimes hard to find all the information such as the working hypothesis, testing scenarios and performance analysis. These elements are necessary for the repeatability of an experiment and thus the ability of comparing properly some models. In this chapter, and during the rest of this thesis, we make sure to give all the necessary information for anyone to be able to repeat the experiments with existing models but also new ones.

4.1 Impact of Obstacles on Collective Motion

Since the formulation of flocking by Reynolds [8], obstacles are being introduced and dealt with different extensions of the model [82, 10, 83, 2]. In chapter 2 (Sec. 2.2), we explained some of the strategies used in order to interact with obstacles. But before diving in new strategies, it is important to understand how the obstacles impact the flocking behavior and what constraints they add in the environment so that we can better adapt the collective strategy later on. These constraints are usually the only reason why flocks are being perturbed and give rise to two phenomena. The first one is the *fragmentation* and the second one is the *collapse*. *Fragmentation* and *collapse* are both consequences of two opposite behaviors. If we go back to the initial interactions as they are defined by Reynolds [8], the *fragmentation* can be seen as an excessive *separation* while the *collapse* can be seen as an extreme *cohesion*. However, these behaviors are not natural and arise because of some constraints in the environment.

The fragmentation phenomena, as we mentioned above, can be described as the formation of distinct clusters that were originally one. This notion has already been studied by Olfati-Saber [10] and has been identified rather as a consequence of a flocking model without global objective. In this thesis, we show that we can maintain a good cohesion (*i.e.* a low fragmentation) among self-propelled agents without introducing any external objective, by extending the range of the attraction. But in complex environments with a lot of obstacles we will see that we still have some fragmentation.

Convex obstacles are usually the main cause of fragmentation because they “smoothly” divide the group in distinct parts as it can be seen on Fig. 4.1. Depending on the size of the obstacles in the environment as well as their distribution, agents may never merge back to a single cluster. This problem is due to the interactions as designed in state of the art flocking

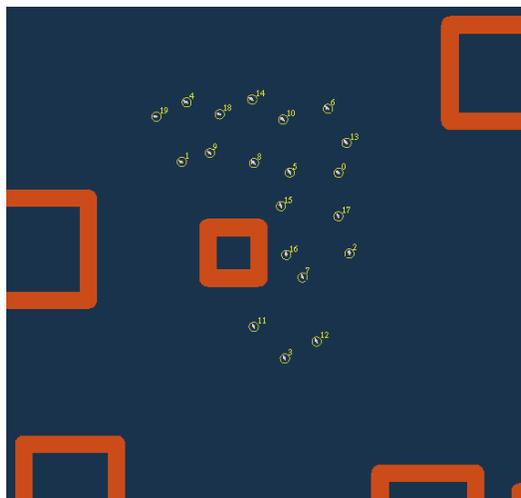


Figure 4.1: 20 agents being separated by a convex obstacle. The agents are moving from the right to the left of the environment.

models. Researchers usually bound the effect of the attraction closely after the equilibrium distance [10, 50, 61] in order to limit the number of agents being attracted. Even if this helps preventing collisions (collapse), it may lead to more fragmentation as we can see in Fig. 4.2. In this figure, we observe 3 agents interacting together at two different times with a focus on the interaction ranges of agent 1 at $t = 0s$ on the left image and the interaction ranges of agent 2 at $t = 5s$ on the right image.

In these classical models, usual notations are:

- The repulsion area \mathcal{R}_{rep} is defined by a disk of radius r_{rep} .
- The attraction area \mathcal{R}_{att} is defined by an annulus starting from radius r_{rep} to r_{att} .
- \mathcal{R}_{rep} and \mathcal{R}_{att} intersect in a circle of radius d_{eq} ($= r_{rep}$), in red here, which corresponds to the “equilibrium distance”.

Such a model (as the one defined by Olfati-Saber [10]) works pretty well in environments without too many obstacles. It is easy to see in fig. 4.2 that a simple obstacle can lead agent 3 to be unable to follow agent 2, leading it to get out of agent 2 attraction area. After overcoming the obstacle, even if agent 3 might still be able to communicate with agent 2, the range of the attraction is not large enough to drive agent 3 closer to agent 2.

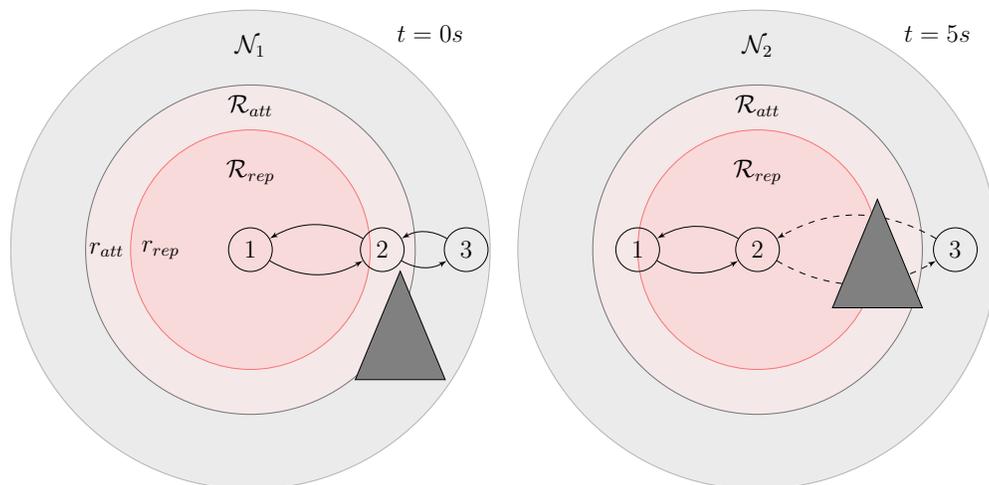


Figure 4.2: 3 agents interacting together at two different time. On the left, at $t = 0s$, the interaction ranges represented by the colored disks, are focused on agent 1. On the right, at $t = 5s$, the focus is on agent 2. The plain black arrows represent an interaction while the dashed arrows represent an uncertain interaction due to the obstacle.

On the other hand, concave obstacles, such as the corners of an arena, usually lead to collisions (collapse) because of the inertia of the system and the geometry of the obstacle. Indeed when reaching a corner, the agents on the front line are trapped between two walls and the rest of the swarm, heading in their direction, which can be seen as a virtual wall (see

Fig. 4.3). Although the repulsion is designed to avoid collisions, its effect might be effective too close to the agents depending on the value of the equilibrium distance (see Eq. 2.25 for example) and its strength cannot be infinite in order to keep it realistic. Regardless of the



Figure 4.3: 20 agents heading in the corner of a concave obstacle.

strategy used to avoid the obstacles, it appears that both fragmentation and collapse are inherent to self propelled flocking models evolving in highly constrained environments.

In this chapter, we aim at extending some flocking models we presented in Chap. 2 in order to take into account the impact of obstacles. Our strategy is not to modify the obstacle avoidance interaction but rather adapt and extend the rest of the interactions such as the attraction and the repulsion. The framework used as well as the hypothesis considered are the ones described in Chap. 3. The environments used and the hyper-parameters will be detailed below.

4.2 Extending Flocking Models

In this section, we extend the control law developed by Vásárhelyi et al. (see Sec. 2.2.4) and Olfati-Saber (see Sec. 2.2.3) in order to increase their performance regarding fragmentation and collision. We also adapt them to the simulation framework we use and the working hypothesis we consider (see Chap. 3).

- The agents are self-propelled.
- An agent only exploits the information of its neighbors in its active neighborhood.
- The control law should be in the form of a *desired velocity*.
- The velocity and the acceleration are bounded.

These assumptions give us a starting point in the definition of the control laws presented below.

4.2.1 Vásárhelyi and Attraction: VAT

Backed by a robust behavior, the velocity controller defined by Vásárhelyi (see Eq. 2.23) was originally designed **without any attraction term**. However within our simulation environment, we observed that, the repetitive obstacles avoidance maneuvers combined with degraded communications eventually led to fragmentation. We hence decided to introduce an attraction term between neighbors based on the half-spring model in order to enhance the cohesion whenever it is possible:

$$\mathbf{v}_{ij}^{att} = \begin{cases} p^{att}(r_0^{att} - r_{ij}) \frac{\mathbf{r}_i - \mathbf{r}_j}{r_{ij}}, & \text{if } r_{ij} > r_0^{att} \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

where p^{att} is a linear gain, $r_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$ is the distance between agents i and j and r_0^{att} is the attraction range under which agents stop attracting each other so as to prevent additional collisions. Also we have $\mathbf{v}_i^{att} = \sum_j \mathbf{v}_{ij}^{att}$ for all neighbors j .

Adding this term to the input control must be done carefully regarding the repulsion term in order to avoid oscillations. Hence, we introduce a buffer distance between attraction and repulsion by setting a radius r_0^{att} over which the attraction is applied. During the rest of this thesis, we set

$$r_0^{att} = \rho * r_0^{rep}. \quad (4.2)$$

where r_0^{rep} is the radius under which the repulsion is applied (see Vásárhelyi et al. model 2.25) and ρ is a real positive coefficient. In the experiments, we will use $\rho = 1.2$. It comes from extensive simulation tests and appeared to enhance the cohesion without generating oscillations. Also we call the annulus between r_0^{rep} and r_0^{att} the **equilibrium zone** referred to as \mathcal{R}_{eq} .

In Fig. 4.4, we show the different interactions based on the distances between 4 agents. The distance between agents 1 and 2, r_{12} is smaller than r_0^{rep} resulting in a repulsion (represented in green). On the contrary, we have $r_{13} > r_0^{att}$ leading to an attraction (represented in red). Regarding agents 4, we have $r_0^{rep} < r_{14} < r_0^{att}$ which means that it is within the buffer area (represented in white) where no attraction nor any repulsion takes place. This particular area called \mathcal{R}_{eq} is a major difference with what we have seen so far. Also, the attraction is unbounded unlike the model of Olfati-Saber for example (see Fig. 4.2).

Finally, we can set our first model proposition, which is a velocity controller, called VAT for Vásárhelyi and Attraction:

$$\tilde{\mathbf{v}}_i^d = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} v^{flock} + \mathbf{v}_i^{rep} + \mathbf{v}_i^{att} + \mathbf{v}_i^{frict} + \sum_s \mathbf{v}_{is}^{wall,obst} \quad (4.3)$$

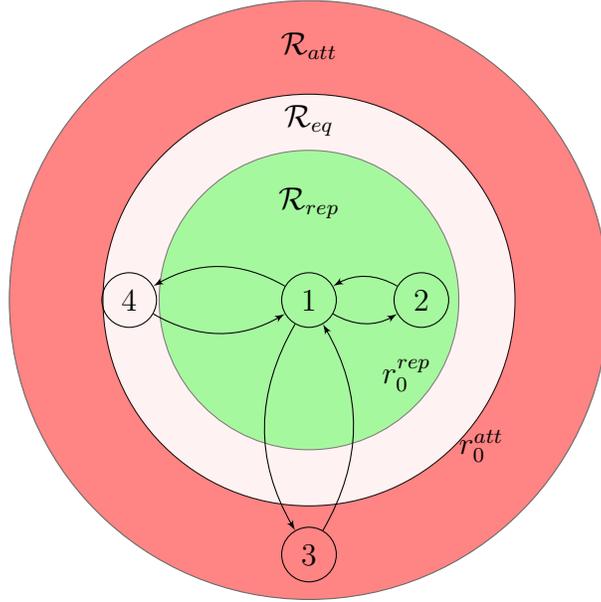


Figure 4.4: Representation of the different parameters depending on the distance between two agents.

where v^{flock} , \mathbf{v}_i^{rep} and \mathbf{v}_i^{frict} are the same terms as defined in Eq. 2.23. The obstacle and wall interaction terms are concatenated in $\mathbf{v}_{is}^{wall,obst}$ for convenience but remain the same as in Eq. 2.23. As we want the velocity to be bounded, the final form of the desired velocity \mathbf{v}_i^d is:

$$\mathbf{v}_i^d = \frac{\tilde{\mathbf{v}}_i^d}{\|\tilde{\mathbf{v}}_i^d\|} \min\{\|\tilde{\mathbf{v}}_i^d\|, v_{max}\}, \quad (4.4)$$

where v_{max} is the maximum allowed velocity magnitude (see Tab. 3.1).

This resulting desired velocity encompasses 12 parameters, acting on different terms of the law and producing different behaviors. We will see in Sec. 4.3 how those parameters are optimized to reach an optimal behavior and what are the resulting performances.

4.2.2 Vásárhelyi and Olfati-Saber: VOS

The second model we propose, denoted VOS (Vásárhelyi + Olfati-Saber), is based on a combination of Olfati-Saber model, exploiting the stability of the behavior derived from using its control input (see Eq. 2.11), and two features of Vásárhelyi's model which are the self-propelling term and the obstacle avoidance term. The objective here is to have a swarm fully decentralized and with a limited-range communication. As a consequence, we get rid of the centralized *navigational feedback* term of Olfati-Saber's model (\mathbf{f}_i^f in Eq. 2.13). Instead, we use the self propelling term which does not encapsulate any shared information but maintains a perpetual motion which is a necessary condition for the emergence of collective

motion [9]. In order to prevent the fragmentation, we exploit the *alignment term* of Olfati-Saber (\mathbf{f}_i^d) and the attraction/repulsion term (\mathbf{f}_i^g). Regarding the obstacle avoidance term, we use the one of Vásárhelyi as it is designed as a velocity control law.

The VOS model is defined as a nested control structure¹ merging both controllers:

$$\begin{cases} \mathbf{u}_i = \alpha \mathbf{f}_i^g + \beta \mathbf{f}_i^d & (4.5a) \\ \tilde{\mathbf{v}}_i^d = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} v^{flock} + \sum_s \mathbf{v}_{is}^{wall,obst} + \int \mathbf{u}_i dt & (4.5b) \end{cases}$$

where \mathbf{f}_i^g accounts for the attraction and the repulsion, \mathbf{f}_i^d is related to the alignment and α and β are some linear coefficients which weight the attraction/repulsion and alignment terms (see Sec. 2.2 for more details). v^{flock} is the self propelling speed and $\mathbf{v}_{is}^{wall,obst}$ is the interaction with walls and obstacles (see Eq. 2.23). The last term of Eq. 4.5b is the integration of the output of Eq. 4.5a which is an acceleration. This is done in order to get a velocity given that we design a velocity controller.

For the same reason as before, we also have:

$$\mathbf{v}_i^d = \frac{\tilde{\mathbf{v}}_i^d}{\|\tilde{\mathbf{v}}_i^d\|} \min\{\|\tilde{\mathbf{v}}_i^d\|, v_{max}\}, \quad (4.6)$$

Similarly to the VAT model, the 10 parameters of the VOS model will be optimized before evaluation in Sec. 4.3.

4.3 Optimization and Comparison

4.3.1 Optimization process

The models we propose and study have several parameters that must be optimized to reach their best possible flocking behavior. Most of performance metrics have been defined in Sec. 2.2.4 and the optimization process has been described in Sec. 3.5. In this chapter we are going to use Ψ_{corr} , Ψ_{col} , Ψ_{wall} and Ψ_{vel} as they have been defined in Sec. 2.2.4. In addition, we modify the cluster metric Ψ_{clust} in order to count the number of clusters during the experiment. This metric concatenates several information that was before divided in two metrics Ψ_{clust} and Ψ_{disc} . By counting the number of clusters with a targeted value of 1, we both minimize isolated agents (previously Ψ_{disc}) and maximize the size of a cluster (previously Ψ_{clust}). We also introduce a new metric Ψ_{pow} directly related to the quality of the communication within a cluster. The goal is to measure the averaged received power of all the agents:

$$\Psi_{pow} = \frac{1}{T} \frac{1}{N} \int_0^T \sum_{i=1}^N \frac{1}{|\mathcal{N}_i^+| - 1} \sum_{j \in \mathcal{N}_i^+} P_{Rx_{ij}}(t) dt, \quad (4.7)$$

¹The structure is composed of a velocity controller and an acceleration controller which output is directly transformed into a velocity.

where $P_{Rx_{ij}}(t)$ is the power received by agent i from agent j (see Sec. 3.3.1). For this metric, we only take into account agents within the same *active neighborhood* as the power will obviously be lower for agents which are not interacting together but still in the same *communication neighborhood*.

Based on these new metrics and the previous ones, the following conditions are required to obtain an optimal flocking behavior: $\Psi_{vel} \rightarrow v^{flock}$, $\Psi_{col} \rightarrow 0$, $\Psi_{clust} \rightarrow 1$, $\Psi_{wall} \rightarrow 0$, $\Psi_{corr} \rightarrow 1$, $\Psi_{pow} \rightarrow P_{opt}$ with $P_{opt} = -50dBm$ which corresponds to good quality communications. The associated fitness functions are given in Tab. 4.1. The mapping functions used are the same as in Sec. 3.5. The parameters used are $\delta_{col} = 0.00003$, $\delta_{wall} = 0.0005$, $\delta_{vel} = 1.5$

Table 4.1: Fitness functions for the performances of VAT and VOS

Fitness function	Formula
F_{corr} (-)	$\Theta(\Psi_{corr}) \cdot \Psi_{corr}$
F_{col} (-)	$\mathcal{F}_1(\Psi_{col}, \delta_{col})$
F_{wall} (-)	$\mathcal{F}_1(\Psi_{wall}, \delta_{wall})$
F_{vel} (-)	$\mathcal{F}_3(\Psi_{vel}, v^{flock}, \delta_{vel})$
F_{clust} (-)	$\mathcal{F}_2(\Psi_{clust} - 1, \delta_{clust})$
F_{pow} (-)	$\mathcal{F}_3(\Psi_{pow}, P_{opt}, P_{min})$

and $\delta_{clust} = 1$. We remind that $P_{opt} = -50$ and $P_{min} = -70$. Most of these values come from [2] but also from an empirical study and a comparison to realistic hardware (for the communication metric especially).

The environment used for the optimization is the same for VOS and VAT. It is called *generic* and its characteristics are gathered in Tab. 4.2.

Table 4.2: Parameters of the *generic* environment

Parameter	Description	Value
R_{arena} (m)	Radius of the arena	$R_{arena} = 625m$
N_{obst} (-)	Number of obstacles in the environment	$N_{obst} = 90$
R_{obst} (m)	Average length of an obstacle	$R_{obst} = 24m$
σ_{obst} (m)	Standard deviation of an obstacle length	$\sigma_{obst} = 12m$
\mathcal{D} (%)	Density of obstacles in the arena	$\mathcal{D} \approx 3.3\%$
\mathcal{P} (-)	Distribution of the obstacles in the arena	$\mathcal{P} = \text{Uniform}$

We also use the *degraded communication* model (see Sec 3.3.1) so as to be in the most challenging case. The number of agents N is set to 10. The optimization process is represented in Fig. 4.5 which tracks the evolution of the average of the metrics of the Pareto front at each generation for both models (VOS in blue and VAT in red). The color bar gives an

Table 4.3: VAT and VOS optimized parameters in GENERIC env. The columns correspond to the fitness functions F_{vel} , F_{col} , F_{clust} , F_{wall} , F_{corr} and F_{pow} .

	Vel	Col	Clust	Wall	Corr	Pow
VAT	0.999	0.99	0.94	1	0.86	0.202
VOS	0.612	0.88	0.92	1	0.784	0.26

index on how advanced we are in the optimization, based on a maximum number of iterations. Although VOS seems to be outperformed by VAT, it is only because its Pareto front is larger with more widespread values than VAT (refer to Sec. 3.5 for more information).

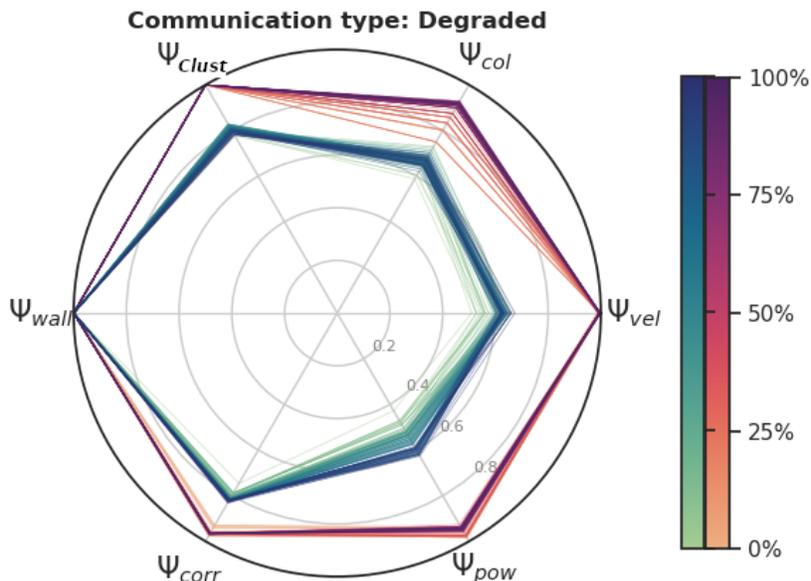


Figure 4.5: Evolution of the metrics during optimization using degraded communication in the genetic environment (VAT in red, VOS in blue).

4.3.2 Results and comparison

Once the best solutions of the VAT and VOS models are retrieved, we first evaluate them in the “training” environment, the *generic* one, with *degraded communication* so as to assess the performances and compare them. In order to cope with the stochastic property of the simulation, the evaluation consists of averaging the results of a hundred runs. The average values of the mapped metrics (1 being the best, 0 the worst) are gathered in Tab. 4.3. One can observe that the VAT model has better performances than the VOS model especially for the *velocity*. Regarding the received power metric (Pow), the low values for both models (0.202 and 0.26) does not mean that the communication is always bad but that there are

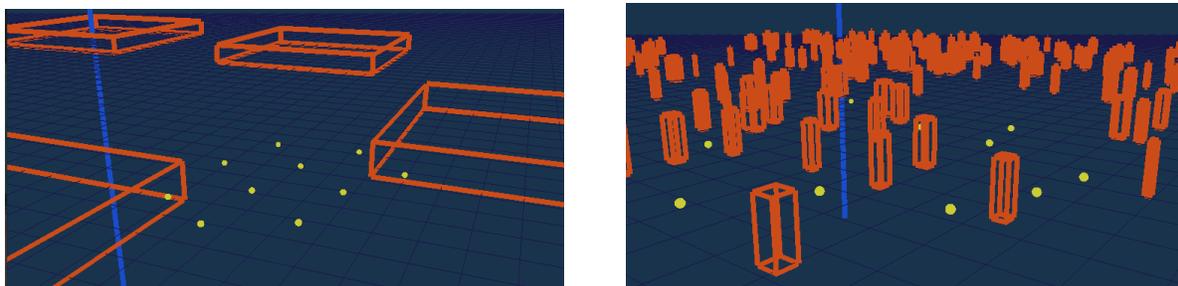
multiple communication outages that penalize the average value. Even if the difference is small, the VOS model appears to have a better performance regarding the communication.

The models are then evaluated in two new environments different from the optimization one. We also compare them with *degraded* and *regular* communications in order to see the impact of the communication’s quality. The testing environments are defined by the characteristics gathered in Tab. 4.4. Their design are very different in order to impact differently the flocking behavior. In Fig. 4.6, we can see how different the two testing

Table 4.4: Parameters of the *forest* and *city* environments

Parameter	Description	<i>Forest</i>	<i>City</i>
R_{arena} (m)	Radius of the arena	625m	625m
N_{obst} (-)	Number of obstacles in the environment	500	15
R_{obst} (m)	Average length of an obstacle	4m	200m
σ_{obst} (m)	Standard deviation of an obstacle length	2m	20m
\mathcal{D} (%)	Density of obstacles in the arena	0.7%	36.5%
\mathcal{P} (-)	Distribution of the obstacles in the arena	Uniform	Uniform

environments are in terms of obstacle density and sizes. On the left, the *city* environment has few obstacles but big ones creating some natural paths but also some bottlenecks making it hard for the agents to explore the arena and move freely. On the right, the *forest* environment has many small obstacles in a way of an actual forest. The fragmentation phenomenon is most likely to happen here and this is exactly what we wanted to test. We will also make a



(a) 3D view of the environment *city*

(b) 3D view of the environment *forest*

Figure 4.6: Snapshots of the testing environments.

comparison to the Vászrhelyi et al. model, denoted VA, that we simulate using its optimal set of parameters given in [2]. The original performances of VA are reported in Tab. 4.5.

Tab. 4.6 and 4.7 gather the results for each model. The average values of the metrics as well their relative variance over the 100 runs are displayed. We also report on the last line of these tables, the average inter-agent distance within an *active neighborhood*. Keep in mind

Table 4.5: VA performances extracted from [2].

	Vel	Col	Clust	Wall	Corr	Pow
VA	0.938	0.945	1	0.997	0.916	N/A

that in these models, the equilibrium distance is optimized and hence, the models will not have the same desired equilibrium distance.

The first important observation we can draw here is that the VA model has much lower performances than the original results of [2] reported in Tab. 4.5. This can be explained by the fact that the authors consider small arenas in their simulations, leading to an indirect effect of cohesion, the agents having no choice but to head back toward the center of the arena. In our study, and to avoid this advantageous behavior, we set a large enough value ($1250m$) for our arena length. Also the distribution of the obstacles is not explicitly mentioned in [2] which obviously impacts most of the metrics. Finally, taking into account a radio propagation model also adds a new constraint to the model that was not considered in the initial work. The drawbacks of not adding an attraction term are even more visible on the *forest* environment where the cluster parameter is relatively low, meaning there has been some fragmentation (Clust=0.637 in Tab. 4.6). One may also wonder why adding an attraction term in the VAT model does not create more collisions. This can be explained by the fact that this attraction term is only effective above a fixed inter-agent distance with a bounded resulting inertia preventing oscillations and hence collisions. Also, thanks to the *active neighborhood*, the interactions are limited to an optimized number of agents preventing dangerous behaviors.

Maintaining the Connection

One of our goal in this thesis was to implement models capable of preventing the undesired effect of fragmentation. We can see in both tables that this objective is mostly satisfied. Indeed, for both VAT and VOS, the average *cluster* metric is high, $0.868 \leq \text{Clust} \leq 0.985$. This means that even if fragmentation occurred, the resulting clusters managed to repair broken connections and got back to a single cluster. This situation is quite frequent in the *forest* environment as the obstacles are really small leading recurrent communication outages.

Received Power and Collision

The inter-agent distance is a resulting parameter of all interactions and is mostly optimized by the *received power* metric but also the *collision* one. Agents must lie within a restricted range of distances between each other in order to be far enough to avoid collisions but close enough to remain connected. Please note that in these tables, a low value of the *received power* doesn't mean that the connection is lost. For instance in Tab. 4.7, we have $\text{Pow} = 0.573$ for the VOS model with *degraded* communications, but this is equivalent to

Table 4.6: FOREST

	Regular			Degraded		
	VA	VAT	VOS	VA	VAT	VOS
Vel	0.975	0.946	0.032	0.975	0.955	0.055
σ	0.010	0.03	0.052	0.008	0.023	0.063
Col	0.301	1	0.793	0.286	0.961	0.799
σ	0.429	0.099	0.382	0.429	0.172	0.380
Clust	0.637	0.98	0.923	0.508	0.938	0.868
σ	0.148	0.085	0.13	0.118	0.121	0.163
Wall	1	1	1	1	1	1
σ	0	0	0	0	0	0
Corr	0.133	0.712	0.54	0.128	0.698	0.531
σ	0.068	0.106	0.12	0.06	0.12	0.112
Pow	0.033	0.53	0.517	0	0.045	0.031
σ	0.049	0.11	0.074	0	0.079	0.066
Dist	132.3	37.9	49.9	132.5	42.11	54.37

Table 4.7: CITY

	Regular			Degraded		
	VA	VAT	VOS	VA	VAT	VOS
Vel	0.999	1	0.484	0.999	1	0.501
σ	1.1e-5	0	0.095	4.8e-5	0	0.101
Col	0.71	0.999	0.918	0.654	1	0.86
σ	0.442	0.09	0.264	0.47	0	0.33
Clust	0.671	0.985	0.939	0.54	0.915	0.925
σ	0.18	0.065	0.142	0.197	0.17	0.165
Wall	1	1	1	1	1	1
σ	0	0	0	0	0	0
Corr	0.605	0.912	0.886	0.73	0.945	0.894
σ	0.165	0.067	0.082	0.112	0.04	0.073
Pow	0.512	0.797	0.684	0.335	0.777	0.573
σ	0.205	0.092	0.073	0.286	0.16	0.204
Dist	82.19	29.4	44.3	86.76	33.8	45.55

a value of $P_R = -59.4dBm$, meaning that the communication was possible on average. As a comparison, in Tab. 4.6, for VAT model with *degraded* communications, $Pow = 0.045 \simeq -71.7dBm$ ($-70dBm$ being the lowest value for a reliable communication). Even if this means that on average the communication was not possible, it was still sufficient to maintain the cohesion of the swarm as $Clust = 0.938$. This shows the resilience of the model and its robustness to communication outages.

Environment Overfitting

As mentioned earlier, the models we simulate are not optimized on the environments we use for the comparison. Therefore, there is a risk of overfitting a particular obstacle distribution, the *generic* environment in our case. We can observe this for the VOS model in Tab. 4.6 where the average *velocity* metric is low, $0.032 \leq Vel \leq 0.055$, equivalent to 60% of the desired velocity v^{flock} whereas in Tab. 4.3, in the *generic* environment, $Vel = 0.612$ which is much better but the number of obstacle is around 5 times smaller. Within the *forest* environment, there are many obstacles close to each other leading to a “cage effect” and thus slowing down the flock movements. It also creates many oscillations $0.531 \leq Corr \leq 0.54$ and potential collisions $0.793 \leq Col \leq 0.799$. The VOS model is hence not robust to the variation of obstacle distribution.

Regular and Degraded Communication

The resulting effects of the *degraded* communication model can be observed in both tables. In particular, the overall tendency to increase the measured inter-agent distance is due to the frequent communication outages preventing the swarm to efficiently remain cohesive. In Tab. 4.7, the differences between the two communication models are not significant and the *degraded* model leads to better results, for some metrics, than the *regular* one. However in Tab. 4.6, because of all the obstacles, there is a huge difference regarding the average received power metric. The rest of the metrics are quite equivalent.

From these results, we can clearly state that VAT outperforms VOS and VA, whether it is in the *forest* or *city* environments, with *regular* or *degraded* communication. Nevertheless the model is not perfect as some collisions between agents occurred which is highly undesired. In addition, we have also observed the limits of the received power metric in terms of relevance. Indeed, it is quite low in all the experiments and its correlation to the desired equilibrium distance is not straightforward. We will see in the next section how we use it differently to reach an even better behavior.

4.4 Exploiting the Equilibrium Distance

4.4.1 Context

Since the beginning of this chapter, we consider the desired equilibrium distance as a parameter of the flocking. Indeed, during the optimization, various values of d_{eq} (or r_0^{rep}) can be generated in order to suit the required objectives (velocity, correlation, collisions and so on). Fig. 4.7 shows the measured inter-agent distance between an agent and its closest neighbor (as well as the standard deviation and the maximum value) using an optimized VAT model in a *forest* environment. In this model, the equilibrium distance is set to $d_{eq} = 20m$ (resulting from the optimization). However in the figure, we can see that the measured average value is around $d = 12m$. This phenomenon is explained by the superposition of

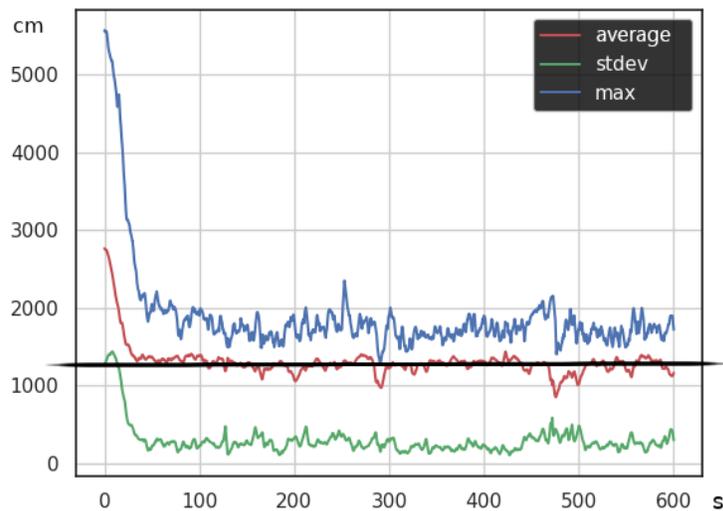


Figure 4.7: Inter-agent distance measured between the agents and their closest neighbor. The red, blue and green lines correspond respectively to the average, the maximum and the standard deviation of the measure. The black line corresponds to the average value over the whole experiment and we have $d = 1200cm$.

the attraction forces that are, in some situations, shrinking the desired equilibrium distance as it can be seen in Fig. 4.8. In this figure divided in 3 phases, we observe 3 agents with the same *active neighborhood* ($\mathcal{N}_{1,2,3}^+$) interacting together. In Phase 1, agents 1 and 3 are at the equilibrium distance with agent 2 and hence only share information for the alignment force (represented by the black arrows). Otherwise, agents 1 and 3 being further than d_{eq} will attract each other (red arrows). This leads eventually to Phase 2 where, because of the attraction, agents 1 and 3 end up being closer than d_{eq} with agent 2 resulting in a repulsive action represented in blue. Finally, all the agents are further than d_{eq} with each others as

shown in Phase 3. This leads to attraction forces among all of them. This cycle will repeat itself until it reaches an equilibrium close to Phase 2 and hence the observed equilibrium will be shrunk in comparison to the desired one. This phenomenon also explains the appearance

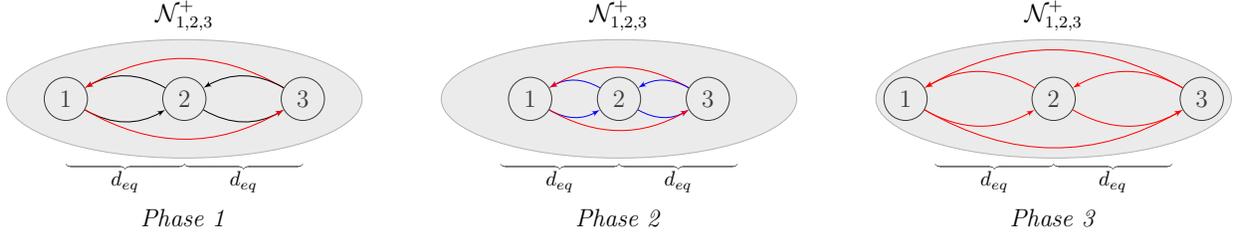


Figure 4.8: Superposition process leading to the modification of the observed equilibrium.

of oscillations in the measured inter-agent distance as it can be seen in Fig. 4.7. Note that this is true for VAT but not for VOS as the attraction force vanishes close after d_{eq} .

The superposition explains, in part, why the optimization process will most of the time generate a desired equilibrium distance relatively high. It takes into account this phenomenon to prevent the collisions leading sometimes to excessively high values but increasing in the same time the risk of fragmentation. However, depending on the application, the user might want to set an equilibrium distance which will be observed in the simulation. This motivates us to include a new optimization criteria based on the desired equilibrium distance.

4.4.2 Introduction of a new distance metric

In this section, we optimize our “best” model VAT with a new objective function based on the minimum distance between an agent and its neighbors in its *active neighborhood*. Moreover, we also removed the *received power* metric as an objective of the optimization because of its strong correlation with the inter-agent distance. We still take the *received power* into account but rather as a constraint removing all solutions with an average received power less or equal to P_{min} . The new metric we use is defined as follows:

$$\Psi_{dist} = \frac{1}{T} \frac{1}{N} \int_0^T \sum_{i=1}^N r_{ij}^{min}(t) dt, \forall j \neq i \quad (4.8)$$

where $r_{ij}^{min}(t) = \operatorname{argmin}_{j \neq i} r_{ij}(t)$. The objective here is to have this metric close to d_{eq} , $\Psi_{dist} \rightarrow d_{eq}$. The advantage is that because we only look at the closest distance between an agent and its neighbors, it will not affect the rest of the distances which are not meant to be equal to d_{eq} (not for all at least). In addition, because of the environment, agents may have to be separated further than d_{eq} for a short moment and this should not penalize the model. To take this into account, Ψ_{dist} is passed into a mapping function with a penalization only under d_{eq} . The mapping function used is \mathcal{F}_3 and the resulting fitness is $F_{dist} = \mathcal{F}_3(\Psi_{dist}, d_{eq}, d_{tol})$ with $d_{tol} = 2m$ the distance tolerance for this metric. Fig. 4.9 shows the evolution of the optimization with the new distance component. We will see in the next section the performances of the resulting model.

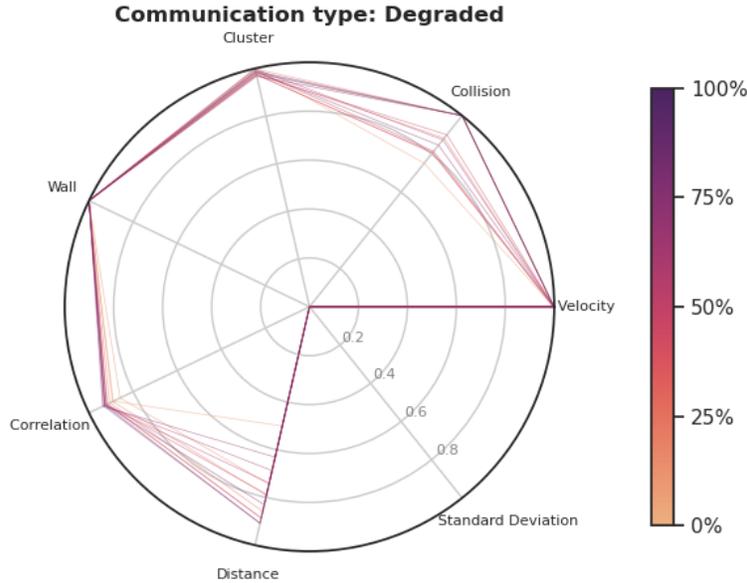


Figure 4.9: Optimization process with the distance and standard deviation metrics.

4.4.3 Performances Analysis

In order to compare the performances of this new variant of the model to the ones of the previous version we have run the simulation in the same environment as previously with the VAT model. Fig. 4.10 shows the average, the standard deviation and the maximum value of the distance between an agent and its closest neighbor. We can see that after a certain time, considered as a transition time, the system seems to reach an equilibrium around 21m (represented by the black line) which is close to the desired value of 20m. This proves the efficiency of the new metric defined in Eq. 4.8. We can also see that we have less oscillations in time but larger ones. This is also due to the definition of the metric and especially the fact that it is still “optimum” to have a distance above d_{eq} . Even though this is not so much observed on the average value (in red) it is quite true for the maximum value (in blue).

We can see here the advantages of this new optimization model which appears to be more suitable for various applications. However it still implies running the optimization for the different equilibrium distance desired.

In the next chapter we will see how we derive a new measure from neighbors distances that we exploit in order to get rid of the limitations of the previous models.

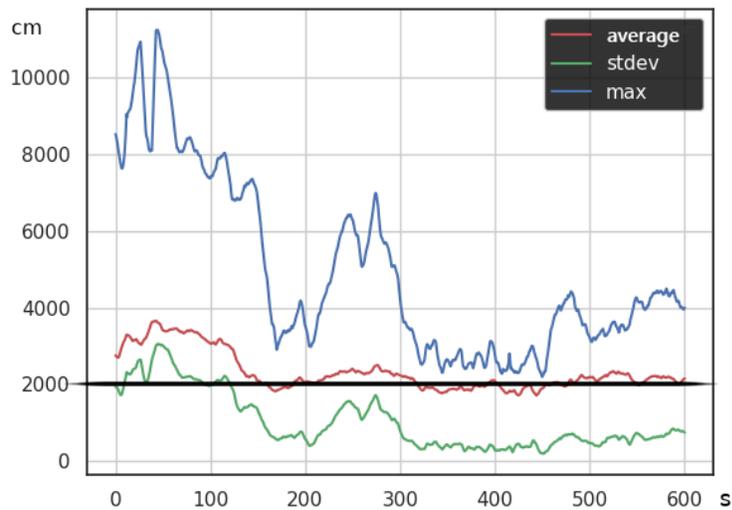


Figure 4.10: Inter-agent distance measured between the agents and their closest neighbor. The red, blue and green lines correspond respectively to the average, the maximum and the standard deviation of the measure. The black line corresponds to the average value over the whole experiment and we have $d = 2100cm$.

After discussing the limits of the VAT and VOS models, we understand the need of detecting the risk of collisions and fragmentation more accurately. For this purpose, we introduce a new individual “pressure” measurement correlated to the risk of collision of an agent with its neighbors. This pressure is communicated to neighbors, then allowing to define an asymmetric repulsion force. This force aims at regulating the pressure and thus anticipate the risk of collision. In addition, in order to reduce the risk of fragmentation, we design an asymmetric attraction force based on a neighbors filter. This filter gives more weight to the agents that are more likely to be isolated. Finally, we come up with a new flocking model, called Asymmetric Pressure Regulation (APR) integrating the interactions we mentioned above into a self propelling model including an alignment force as well as an obstacle avoidance force. The performance of this model is evaluated through simulations in various types of environment and is compared to the VAT model presented in the previous chapter.

5.1 Motivations for a New Asymmetric Model

Even though the VAT and VOS models, presented in the previous chapter, have good performances in highly constrained environments, they are not perfect and still lead to unwanted behaviors such as collisions and fragmentation. We understood the duality of these two phenomena and their causes. The collisions are mainly due to the *superposition* of the attraction forces (see Sec. 4.4.1) which leads to instabilities and shrinks down the equilibrium space as we saw with VAT. However, bounding the attraction, as it is done with VOS, may hinder the cohesion and lead to fragmentation. This apparent dead end is mostly due to the fact that models include pairwise and symmetric interactions. The symmetry is yet commonly used as it “guarantees” the stability of the model.

Besides the stability, symmetric models are also used because the interactions rely on “pairwise” information such as the inter-distance or the difference of velocity between two agents. In this chapter, we aim at introducing asymmetric interactions based on asymmetric information which is proper to each agent. The goal is to create independent motion which does not always impact the neighborhood of an agent and hence participate to its own purpose of avoiding collisions and maximizing the cohesion. To build this information we want to use the knowledge available to each agent and concatenate it into a single measurement that would then be shared.

Two different measures will be introduced in order to create a new asymmetric model. The first one called **pressure** is a local estimation of the risk of collision of an agent based on the distance to the equilibrium of all its neighbors. Thanks to the communication of this pressure, an immediate action can be taken by all the other agents in range and not only the one below the equilibrium. The second measure is correlated to the risk of fragmentation. We define an **adaptive filter** that identifies the agents that are more likely to be isolated in order to initiate an attractive motion towards the cluster.

The chapter is organized as follows. In Sec. 5.2, we define the new pressure measurement before introducing the resulting repulsion in Sec. 5.3. In Sec. 5.4, we develop the idea of the adaptive attraction based on the risk of fragmentation. The entire *asymmetric pressure regulation* model is presented in Sec. 5.5 and the performances of the model as well as a study on the pressure dissipation are given in Sec. 5.6.

5.2 An Individual Pressure Measurement

In most flocking models, the inter-agent distance is exploited as a local and reciprocal input of the interactions such as the repulsion and the attraction. Hereafter, we want to exploit the inter-agent distance more globally by defining an information that scales to any neighborhood. More precisely, we aim at **assessing the risk of collision of an agent with its neighbors**.

For this purpose, we **identify the neighbors that are evolving under the equilibrium distance**. We derive a measure, called **pressure**, which calculates the gap to the equilibrium distance:

$$p_i = \frac{1}{\sum_{j \in \mathcal{N}_i^+} \Theta(d_{eq} - r_{ij})} \sum_{j \in \mathcal{N}_i^+} \Theta(d_{eq} - r_{ij}) * (d_{eq} - r_{ij}), \quad (5.1)$$

where p_i is the pressure of agent i and \mathcal{N}_i^+ its active neighborhood. r_{ij} is the distance between agents i and j , d_{eq} is the equilibrium distance and Θ is the Heaviside step function¹. The Heaviside function allows to consider only neighbors which are closer than the d_{eq} distance. Note that this formula is only valid when at least one agent is below the equilibrium distance with agent i . Otherwise we consider that the pressure is 0.

We illustrate the pressure in Fig. 5.1 with three agents, where agents 1 and 2 are under the equilibrium distance and agents 2 and 3 are above. As a consequence, agents 1 and 2 have a positive pressure value (represented by the orange color on the agents).

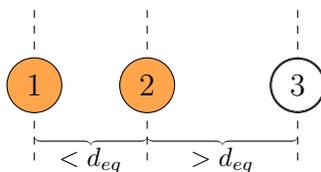


Figure 5.1: Representation of the pressure with 3 agents depending of the inter-agent distance.

As for position and velocity data, **the pressure is also shared between neighboring agents**. Thanks to this sharing, agents can now be informed of a risk of collision with their neighbors. We will see in the next section how we exploit this information to define a new repulsion law.

The measure of pressure can also be helpful to identify sudden changes in the environment such as obstacles. Whenever the flock is being constrained by obstacles, it usually forces the agents to brutally change their headings or to get closer to each others. Most of the time, this situation will yield an high pressure region. In Fig. 5.2, we can see that the narrow path between the two obstacles constrains the agents to move critically close to each others leading to a lot of pressure represented by the yellow circles around the agents. If this pressure measurement tells us that some agents in the system are below their equilibrium state, it also gives us an essential information which is a sudden change in the environment.

¹ $\Theta(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$

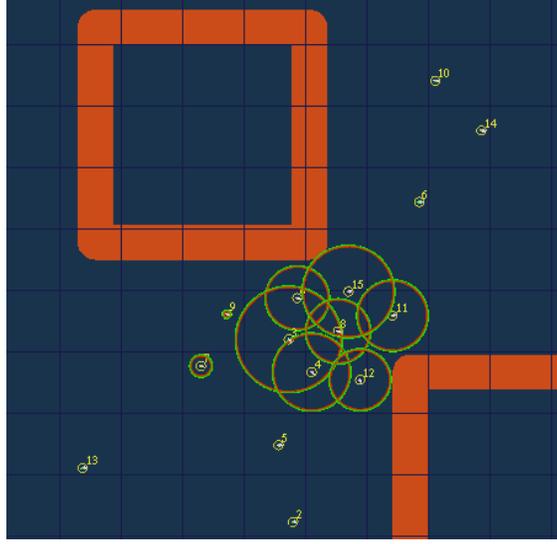


Figure 5.2: Measure of the pressure in a narrow path. The pressure is represented by the yellow circles around the agents. The larger the circle, the greater the pressure measured.

5.3 Pressure Based Repulsion

As we are now able to communicate the risk of collision between neighboring agents, we define a repulsion law based on this information. Unlike classical models, we want to initiate a repulsive motion above the equilibrium distance so as to anticipate critical situations.

Repulsion definition

As we want to reduce the pressure in the swarm, *i.e.* reduce the risk of collision, we design the repulsion so as to move agents away from pressured areas (areas around agents with pressure). Thus, the repulsive force \mathbf{v}_{ij}^{press} of an agent i receiving the pressure information of an agent j is:

$$\mathbf{v}_{ij}^{press} = -k_{press} * p_j * \|\mathbf{v}_i\| * \mathbf{n}_{ij} \quad (5.2)$$

where k_{press} is a linear gain, p_j is the pressure of agent j , \mathbf{v}_i is the velocity vector of agent i and \mathbf{n}_{ij} is the unit vector between i and j . And the overall repulsive force of agent i is $\mathbf{v}_i^{press} = \sum_{j \in \mathcal{N}_i^+} \mathbf{v}_{ij}^{press}$. One can see that this new force is proportional to the pressure of the agents sending their pressure information to agent i but also to the velocity of the agent receiving it. \mathbf{v}_i acts as a gain in order to adapt the strength of the repulsion. Indeed, if agent i is moving fast, there is a urge to have a strong braking motion.

Preventive repulsion

This new repulsion defined in Eq. 5.2 differs from state of the art models as it is not bounded to an area below the equilibrium distance. It means that **the repulsion can**

happen simultaneously with the attraction. The idea here is to soften the strength of the attraction when it drives an agent towards a pressured area. This mechanism acts as a brake but should not hinder the attraction motion. Hence, we define a condition on the strength of the pressure repulsion when it is combined with the attraction:

$$\mathbf{v}_{ij}^{press} = \frac{\mathbf{v}_{ij}^{press}}{\|\mathbf{v}_{ij}^{press}\|} \min\{\|\mathbf{v}_{ij}^{press}\|, \|\mathbf{v}_{ij}^{att}\|\} \quad (5.3)$$

With Eq. 5.3, we have a guarantee that the pressure repulsion will never be stronger than the attraction.

In Fig. 5.3 we compare a classical repulsion (see Eq. 2.25) on the left and a pressure repulsion (see Eq. 5.2) on the right. In this figure, four agents are moving towards a concave

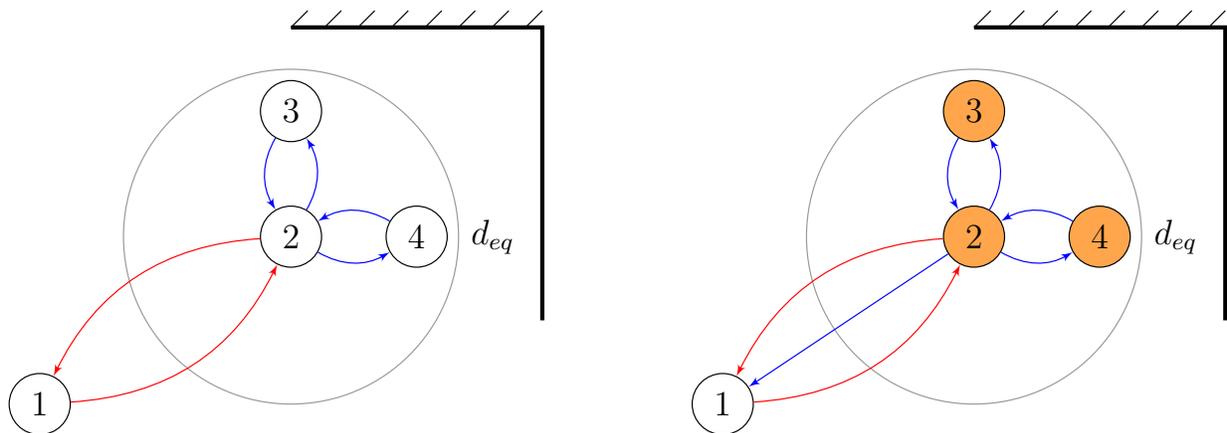


Figure 5.3: Comparison of two models with 4 agents. On the left we use a classical repulsion and a pressure repulsion on the right.

obstacle. We only plot the interactions (arrows) related to agent 2. In both images, one can see that repulsive forces are generated between agents 3 and 2 and between agents 4 and 2, the inter-agent distances being smaller than d_{eq} (blue arrows). On the contrary, the distance between agents 1 and 2 is greater than the equilibrium distance leading to a reciprocal attraction between them (red arrows). The main difference between the two models is that the pressure repulsion, on the right, is generated further than d_{eq} . It slows down the attraction towards agent 2 which has already a lot of pressure (indicated by the orange color of the nodes). On the left image, the repulsion will happen only when agents 1 and 2 will be closer than d_{eq} , increasing agent 2 pressure along with the risk of collision.

To summarise, the pressure repulsion can be seen as a corrective action under the equilibrium distance and a preventive action above it. In Sec. 5.6, we will see the performances of this repulsion and compare it to the VAT model.

5.4 Adaptive Neighborhood for Attraction

Most state of the art models consider that the attraction must be applied only to agents in a limited neighborhood (in terms of distance). If this offers some stability to the system, it yet leads easily to some fragmentation and isolated agents in presence of obstacles [52]. In contrast, some models such as VAT apply the attraction force to the entire communication neighborhood. In that case, the cohesion is improved but we can also observe a gap between the desired equilibrium distance and the actual inter-agent distance likely to generate collisions as we explained in the previous chapter (see Fig. 4.8). As an illustration of the VAT model, Fig. 5.4 shows 3 agents at the equilibrium distance two by two. As they share the same active neighborhood, one can see, in Phase 1, that agents 1 and 3 attract each other which will get them closer to agent 2. Even though the repulsion will happen once this state is reached in Phase 2, it will stabilize the movement under the equilibrium distance. This mechanism can become dangerous when it is applied among many agents.

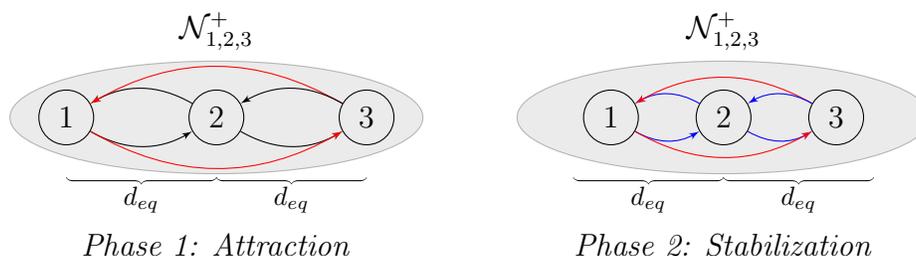


Figure 5.4: Representation of three agents at the equilibrium two by two in Phase 1 and once they are stabilized in Phase 2 (VAT model).

In the following, we propose a new approach to overcome this issue while applying the attraction to a large neighborhood.

Neighbors Filter

We design a neighborhood filter in order to dynamically modify the impact of each agent regarding the attraction. Indeed, in some environments, the attraction might not be necessary to maintain the cohesion, as in the work of Vásárhelyi et al. and the one of Vicsek [2, 9] where there is no attraction at all. Thanks to the alignment force, agents within the same active neighborhood (AN) usually remain together without any other forces. However, whenever obstacles generate fragmentation, the alignment force is no longer sufficient to drive the agents back to a single cluster.

In a general manner, neighboring agents tend to align with each other as they have the same AN. On the contrary, the less similar the active neighborhoods of two agents are, the less aligned they will be. In this case, the attraction is necessary to avoid a risk of fragmentation. We illustrate this risk in Fig. 5.5 where there are five agents interacting together with an AN bounded by $N_{max} = 2$. We can distinct two main clusters formed respectively by agents

5 and 3 on the left and by agents 1, 2 and 4 on the right. Independently, these clusters will remain cohesive and aligned as they are fully connected. But the only link between them is the directed edge from agent 1 to agent 3, they are thus partially connected. This means that agent 3 is the only one capable of ensuring the cohesion of the whole swarm.

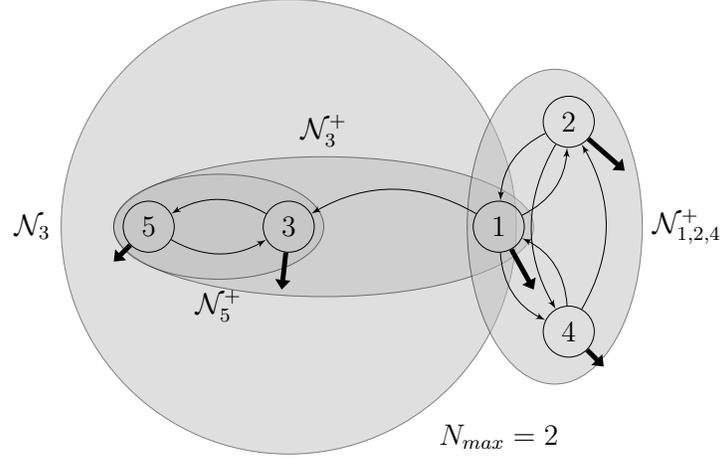


Figure 5.5: Representation of 5 agents with $N_{max} = 2$. The thick black arrows represent the current velocities of the agents.

Based on this observation, we can deduce that the impact of the attraction should be increased only between partially connected agents with few similarities between their AN. Thus, we define a filter J_{ij} , from the point of view of an agent i , comparing its similarities with another agent j :

$$J_{ij} = \begin{cases} 1 - \frac{|\mathcal{N}_i^+ \cap \mathcal{N}_j^+|}{|\mathcal{N}_i^+ \cup \mathcal{N}_j^+|}, & \text{if } i \notin \mathcal{N}_j^+ \\ 0, & \text{otherwise} \end{cases} \quad (5.4)$$

This filter is directly inspired from the Jaccard index [84]. It measures the ratio of similarities between neighborhoods and “map” it between 0 and 1. In the following subsection, we use this filter to weight the attraction between the agents.

Attraction Definition

In order to define the attraction force, we have to take into account some properties. For obvious reason, we do not want the attraction to be effective below the equilibrium distance. We also need its strength to be stronger when the inter-agent distance is large so as for isolated agents to catch up with the main cluster. Finally, we want to use the neighbors filter J_{ij} in order to have a stronger attraction between agents that do not have a similar active neighborhood. Doing so, we expect isolated clusters or lone agents to reach the main cluster. According to these properties, the attraction force \mathbf{v}_{ij}^{att} of agent i receiving

information of agent j is:

$$\mathbf{v}_{ij}^{att} = \begin{cases} p^{att}(\mathcal{D}(r_{ij}) - d_{eq}) * J_{ij} * \mathbf{n}_{ij}, & \text{if } r_{ij} > d_{eq} \\ 0, & \text{otherwise} \end{cases} \quad (5.5)$$

where p^{att} is a gain parameter, r_{ij} is the inter-distance between agent i and agent j , \mathcal{D} is a positive increasing function of r_{ij} , J_{ij} is the filter defined in Eq. 5.4 and \mathbf{n}_{ij} is the unit vector between i and j . Note that in our experiments, we use $\mathcal{D}(r_{ij}) = r_{ij} * \log(1 + q_{ij})$ as the quadratic expression was too harsh and the linear one was too soft. The overall attraction for agent i is $\mathbf{v}_i^{att} = \sum_{j \in \mathcal{N}_i^+} \mathbf{v}_{ij}^{att}$. From this definition, and especially because of the filter, the attraction is now nonreciprocal, which means that two agents interacting together can no longer attract each other. As

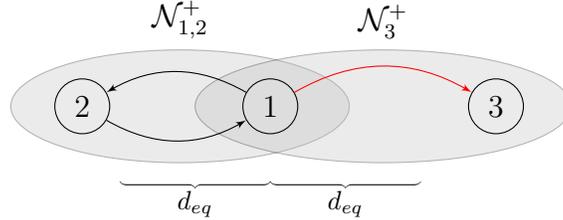


Figure 5.6: Representation of the attraction in red based on the neighbors filter.

illustrated in Fig. 5.6 one can see that agents 1 and 2 have the same AN and an inter-distance larger than d_{eq} but are no longer attracting each other as $\mathcal{N}_1^+ = \mathcal{N}_2^+ = \mathcal{N}_{1,2}^+$. On the other hand, agent 3 is attracted by agent 1 (represented by the red arrow) because their inter-distance is larger than d_{eq} and they do not have the same AN ($\mathcal{N}_3^+ \neq \mathcal{N}_1^+$). Consequently, agent 3 will reach back to the main cluster formed by agents 1 and 2.

In Figure 5.7 we can see a snapshot of a simulation with a swarm of 15 agents (with $N_{max} = 7$). The red arrows represent the attraction forces derived from Eq. 5.5. We can distinguish two clusters in this image, one on the top and one on the bottom. The direction of the attractions appears to drive the clusters to gather which is what we aim with our new attraction force. The solid lines represent a bidirectional interaction (*i.e.* $i \in \mathcal{N}_j^+$ and $j \in \mathcal{N}_i^+$) while the dashed lines represent a unidirectional interaction (*i.e.* $i \in \mathcal{N}_j^+$ and $j \notin \mathcal{N}_i^+$).

In the next subsection, we integrate this new force as well as the pressure repulsion in a whole model called APR.

5.5 The Asymmetric Pressure Regulation Model (APR)

In this section, we present the general form of the model we propose, called *asymmetric pressure regulation model* (APR). It is inspired from Vásárhelyi's article defined by Eq. 2.23

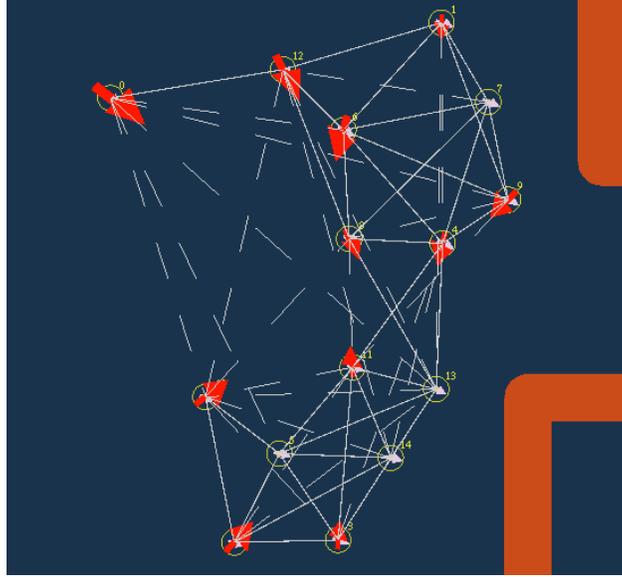


Figure 5.7: Attraction forces among a 15 agents swarm represented by the red arrows. The actual velocity of each agent is not represented here.

and, as the VAT model (see Eq. 4.3), it includes an attraction force. In the proposed APR model, we introduce the pressure repulsion (see Sec. 5.3) as well as the filter based attraction (see Sec. 5.4). The goal of this combination is to increase the cohesion (*i.e.* decrease the fragmentation) while reducing the risk of collision. The final form of the model is described by the following desired velocity $\tilde{\mathbf{v}}_i^d$:

$$\tilde{\mathbf{v}}_i^d(t+1) = \frac{\mathbf{v}_i(t)}{\|\mathbf{v}_i(t)\|} v^{flock} + \mathbf{v}_i^{press}(t) + \mathbf{v}_i^{att}(t) + \mathbf{v}_i^{frict}(t) + \sum_s \mathbf{v}_{is}^{wall,obst}(t), \quad (5.6)$$

which is bounded to v^{max} according to the following expression

$$\mathbf{v}_i^d = \frac{\tilde{\mathbf{v}}_i^d}{\|\tilde{\mathbf{v}}_i^d\|} \min\{\|\tilde{\mathbf{v}}_i^d\|, v^{max}\}, \quad (5.7)$$

where \mathbf{v}_i^d is the desired velocity of an agent i and v^{max} is the maximum allowed velocity. Most of the terms have been described in the previous chapters. We decided not to modify the terms \mathbf{v}_i^{frict} and $\mathbf{v}_{is}^{obstacle}$ as they are already performing well independently, while \mathbf{v}_i^{press} is defined in Eq. 5.2 and \mathbf{v}_i^{att} is defined in Eq. 5.5.

The acceleration \mathbf{a}_i of each agent is computed in the same way we explained in Sec. 3.2 and all the parameters are given in Tab. 3.1:

$$\mathbf{a}_i = \boldsymbol{\eta}_i + \frac{\mathbf{v}_i^d - \mathbf{v}_i - \mathbf{v}_i^s}{\|\mathbf{v}_i^d - \mathbf{v}_i - \mathbf{v}_i^s\|} \cdot \min\left\{\frac{\mathbf{v}_i^d - \mathbf{v}_i - \mathbf{v}_i^s}{\tau_{CTRL}}, a_{max}\right\}, \quad (5.8)$$

where $\boldsymbol{\eta}_i$ models the outer noise (*e.g.* the wind), \mathbf{v}_i^s represents the inner noise on the velocity measurements (*e.g.* due to GNSS approximations), τ_{CTRL} is the settling time of the velocity controller (*e.g.* PID controller) and a_{max} is the acceleration boundary.

In the following section, we evaluate the APR model via simulation through different types of environments. We optimize the parameters of the model in order to compare it to other self propelled models.

5.6 Experimentation & Results

5.6.1 Simulation settings

For the simulations, we test flocks of 15 agents evolving in various environments with obstacles. We use the same framework that we presented in Chap. 3. We added a visual interpretation of the pressure as we showed in Fig. 5.2.

As for the previous models, we define three types of environments. The *generic*, the *forest* and the *city*. While the first one is used for optimization and testing, the other ones are used for performance evaluation and comparisons. We decided to use smaller arenas in this series of evaluations as it makes the simulation faster, the number of obstacles being smaller. The arena is still large enough for the agents not to reach a wall during an experiment. All the parameters of the environments are gathered in Tab. 5.1.

Table 5.1: Parameters of the *generic*, *forest* and *city* environments

Parameter	Description	<i>Generic</i>	<i>Forest</i>	<i>City</i>
R_{arena} (m)	Radius of the arena	250m	250m	250m
N_{obst} (-)	Number of obstacles in the environment	35	200	15
R_{obst} (m)	Average length of an obstacle	24m	2m	60m
σ_{obst} (m)	Standard deviation of an obstacle length	5m	1m	10m
\mathcal{D} (%)	Density of obstacles in the arena	10%	0.4%	25%
\mathcal{P} (-)	Distribution of the obstacles in the arena	Uniform	Uniform	Uniform

All the different experimental constants have been defined in Chap. 3 and remain the same here. In addition, because we no longer optimize the equilibrium distance, we set it to $d_{eq} = 10m$.

5.6.2 Metrics and parameters optimization

The APR model integrates several parameters that have to be optimized such as the gains introduced in Eq. 5.2 and Eq. 5.5, but also an hyper-parameter N_{max} (size of the neighborhood). For a given environment, these parameters can be optimized in order to obtain the best flocking behavior according to some metrics. The optimization strategy defined in Sec. 3.5 is applied here with a new metric regarding the measured pressure (which must be

minimal):

$$\Psi_{press} = \frac{1}{T} \frac{1}{N} \int_0^T \sum_{i=1}^N p_i(t) dt, \quad (5.9)$$

where p_i is the pressure of agent i (see Eq. 5.1), T is the time of an experiment and N is the number of agents. In Tab. 5.2, we gather the different metrics used for the APR model and the objective value. All the formulas have been defined previously (see Sec. 3.5.1 and 4.3.1).

Table 5.2: Metrics used for the APR model and the objective value

Metrics	Description	Objective
Velocity (m/s)	Measure the average velocity of the flock	$\Psi_{vel} = 4$
Correlation (-)	Measure the average alignment per cluster	$\Psi_{corr} = 1$
Cluster (-)	Count the number of clusters	$\Psi_{clust} = 1$
Collisions (-)	Count the number of collisions between the agents	$\Psi_{col} = 0$
Obstacle collisions (-)	Count the number of collisions between an agent and an obstacle	$\Psi_{obst} = 0$
Pressure (m)	Measure the average pressure of the flock	$\Psi_{press} = 0$

The associated fitness functions are presented in Tab. 5.3 and the parameters used are $\delta_{col} = 0.00003$, $\delta_{wall} = 0.0005$, $\delta_{vel} = 1.5$, $\delta_{clust} = 1$ and $\delta_{press} = 1$.

Table 5.3: Fitness functions for the performances of the APR model

Fitness function	Formula
F_{corr} (-)	$\Theta(\Psi_{corr}) \cdot \Psi_{corr}$
F_{col} (-)	$\mathcal{F}_1(\Psi_{col}, \delta_{col})$
F_{wall} (-)	$\mathcal{F}_1(\Psi_{wall}, \delta_{wall})$
F_{vel} (-)	$\mathcal{F}_3(\Psi_{vel}, v^{flock}, \delta_{vel})$
F_{clust} (-)	$\mathcal{F}_2(\Psi_{clust} - 1, \delta_{clust})$
F_{press} (-)	$\mathcal{F}_2(\Psi_{press}, \delta_{press})$

We use the *NSGA-III* algorithm here again as it gave us quick and reliable solutions in our previous experiments. We can see in Fig. 5.8 the evolution of the optimization process. Note that the standard deviation metric has been adapted in order to reach one whenever its value is minimal. The best solution is retrieved at the end of the optimization process by selecting it on the Pareto front according to the “rules” we defined in Sec. 3.5.2.

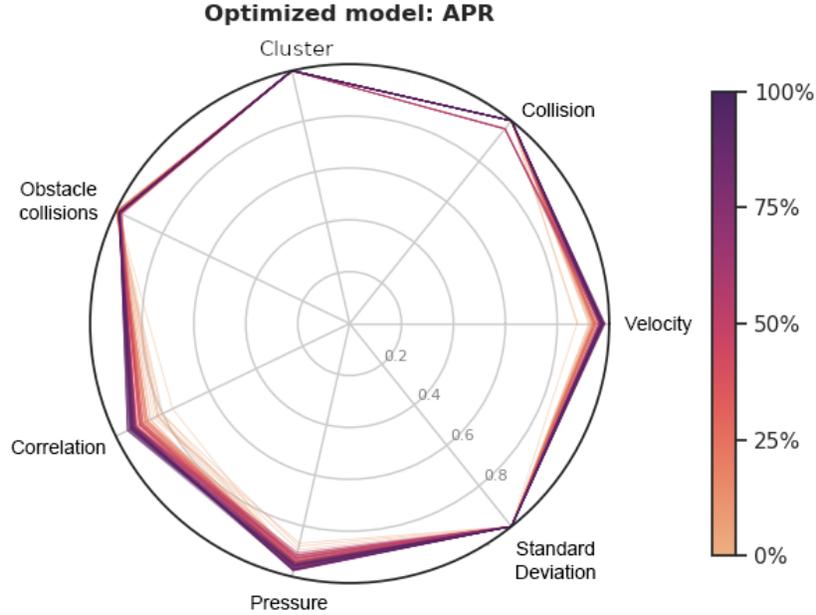


Figure 5.8: Optimization process of the APR model with the NSGA-III algorithm.

5.6.3 Results

In this section, we gather all the results of the tests we ran regarding the APR model but also the VAT model in order to compare them. The models are tested on the City and Forest environments but also on the Generic one so as to discuss the performances on testing and training environments. For each model, we run one hundred simulations on the same environment and compute the average and standard deviation of the metrics. One simulation consists of a $T = 600s$ experiment where the agents are initialized inside a ring of radius $R_{init} = 50m$ with a null velocity.

Generic Environment

We begin the tests with the APR model on the Generic environment as it is the one where we expect the best results. The average and the standard deviation of the metrics are presented in Tab. 5.4. We can see that the results are almost optimal. We do not have any collision between the agents or with the obstacles and none of the agents have left the cluster. In other words, there is no collapse nor fragmentation. It is also important to note that the pressure is very low (0.2, considering that the maximum pressure is $d_{eq} = 10m$). It means that the pressure repulsion is efficient without leading to extreme repulsion as we can see that, on average, the inter-agent distance is very close to the desired equilibrium distance and the standard deviation of the pressure is small as well (0.29). The only drawback here is the velocity of the flock which is at $3.57m/s$ for an objective of $4m/s$. This can be explained by the fact that the pressure repulsion acts as a brake and hence reduces a bit the velocity of the swarm. It is yet still in the range of tolerance.

Table 5.4: Performances of the APR model in the Generic environment.

Metrics	Generic		-	
	Average	Stdev	Objective	Tolerance ¹
Velocity (m/s)	3.57	0.5	4.00	1.00
Correlation (-)	0.94	0.07	1	0.1
Cluster (-)	0	0	0	1
Collisions (-)	0	0	0	0
Obstacle collisions (-)	0	0	0	0
Pressure (m)	0.2	0.29	0	1
Inter-agent distance (m) ²	10.8	1.64	≈ 10	-

^aThe tolerance corresponds to the maximum gap with the objective above which the model is considered as deficient.

^bThe inter-agent distance is not optimized. The objective corresponds to $d_{eq} = 10m$.

The good performances here are expected, as we are testing the model in the optimization environment, and are very encouraging.

Forest and City Environments

It is also important to test the APR model in different environments from the optimization one in order to check if it is robust. Otherwise, we could be confronted with an overfitting phenomenon. The results in the Forest and City environments are displayed in Tab. 5.5.

Table 5.5: Performances of the APR model in the Forest and City environments.

Metrics	Forest		City		-	
	Average	Stdev	Average	Stdev	Objective	Tolerance
Velocity (m/s)	3.01	0.54	3.71	0.46	4.00	1.00
Correlation (-)	0.90	0.11	0.95	0.05	1	0.1
Cluster (-)	10^{-6}	≈ 0	10^{-5}	≈ 0	0	1
Collisions (-)	0	0	0	0	0	0
Obstacle collisions (-)	0.1	0.34	0.04	0.19	0	0
Pressure (m)	0.26	0.36	0.18	0.27	0	1
Inter-agent distance (m)	10.86	1.81	10.72	1.21	≈ 10	-

For the Forest environment we can see that the performances are not as good as in the generic environment. First, for the velocity, we have a 25% gap regarding the objective.

This is clearly explained by the high number of obstacles slowing down the agents repeatedly combined with the effect of the pressure repulsion. Regarding the collisions, we can see that we do not have any collision among the agents but a none zero value for the obstacle collisions. This 0.1 value can be seen as 10% of the experiments generating 1 collision between an agent and an obstacle over the 600 seconds experiment. The Cluster metric, that we can call here “agent not in cluster”, is also very low, which is a really good performance in such an environment where the obstacles are constantly breaking the communication links between the agents. For the pressure, the value is also low, which is a good indicator for the respect of the equilibrium distance. Indeed, the inter-agent distance is on average $10.86m$, close to the desired one ($10m$). Concerning the correlation, the value is really good as we cannot expect the whole swarm to head in the same direction while avoiding obstacles.

For the City environment, the performances are slightly better and it is due to the low number of obstacles leading to a better cohesion of the swarm and hence a better velocity, stronger correlation and less collisions with the obstacles. Indeed, because there are less obstacles (even though they are bigger), the agents are less frequently constrained by the environment and there exist wider paths where they can evolve “freely” which is the main reason why the speed is almost equal to the desired value ($3.71m/s$). For the same reason, the agents evolve all together in the same direction leading to a good correlation metric (0.95) as well as a very good value for the metric “agent not in cluster”. The rest of the metrics are quite similar to the ones obtained in the Forest environment. However, we cannot ignore the obstacle collisions value. Even though collisions with the obstacles are rare, there is room for improvement.

The results of the APR model in the City and Forest environments are very good and allow us to say that there is no global overfitting in the optimization process. One can see that no collision among agents happened in any of the environments which is a major advantage of this model. Nevertheless, we count some collisions with the obstacles. There might be a local overfitting phenomenon regarding the obstacle avoidance feature as there are no collision in the Generic environment. Despite this drawback, the gains brought by APR in terms of fragmentation and collision are significant.

Comparison with the VAT model

After evaluating the APR model in various environments, we can now compare it to other self propelling models. In particular, we use the VAT model (see Eq. 4.3) as it is robust in constrained environments and efficient in avoiding fragmentation and collision.

Both APR and VAT have been optimized using the same process and algorithm. The only difference is that VAT does not use the pressure but the inter-agent distance as a metric of the optimization. The results of the tests for the VAT model in the Generic, Forest and City environments are gathered in Tab. 5.6. We have highlighted the metrics in red when they are worse than the values obtained with the APR model. From this table, we can see that APR performs better in terms of collisions among agents, fragmentation (agent not in cluster) and correlation. We also observe that with VAT the inter-agent distance is

lower than the desired value ($10m$) in the Generic environment ($9.96m$) and in the Forest environment ($9.54m$). This is caused by the superposition of the attraction leading to a shift between the desired and actual equilibrium distance as we explained in the introduction of Sec. 5.4.

Table 5.6: Performances of the VAT model in the Generic, Forest and City environments.

Metrics	Generic		City		Forest	
	Average	Stdev	Average	Stdev	Average	Stdev
Velocity (m/s)	3.53	0.60	3.61	0.59	3.28	0.58
Correlation (-)	0.83	0.19	0.85	0.17	0.88	0.14
Cluster (-)	0.03	≈ 0	0.06	≈ 0	10^{-4}	≈ 0
Collisions (-)	0.51	0.92	0.57	0.98	0.47	0.81
Obstacle collisions (-)	0	0	0	0	0.007	0.02
Inter-agent distance (m)	9.96	3.72	10.43	5.53	9.54	2.07
Pressure (m)	0.86	0.72	0.85	0.70	0.88	0.80

The interesting property of the VAT model is that there is no collision with the obstacles in the Generic and City environments and almost none in the Forest environment. Even though APR and VAT share the same obstacles avoidance term, it seems that the dynamics brought by the new interactions in APR affect directly the performances of the obstacle avoidance. To sum up, we can say that the APR model appears to outperform the VAT model in many aspects and hence answers our problematic. However, it would be interesting to study further the obstacle avoidance function, which was not our primary focus in this work.

5.6.4 Analysis of pressure dissipation

One of the most interesting phenomena regarding the APR model is how it dissipates the pressure and thus reduces the risk of collision. In this section, we analyze this mechanism and compare it, in a similar situation, to the VAT model.

For this purpose, we designed a simple experiment, without obstacles, where 15 agents are initialized without speed, in a deterministic way, in a very small area (a circle of radius $r = 20m$) that will for sure generate some pressure as most of them will be under the equilibrium distance with their neighbors ($d_{eq} = 10m$). The initialization can be seen in Fig. 5.9 and we can see the pressure generated as the agents are below the equilibrium distance with their neighbors. The idea is to see how the pressure is then dissipated and how it evolves towards a steady state. The results for the VAT and the APR models are plotted in Fig. 5.10 and are discussed hereafter through different periods.

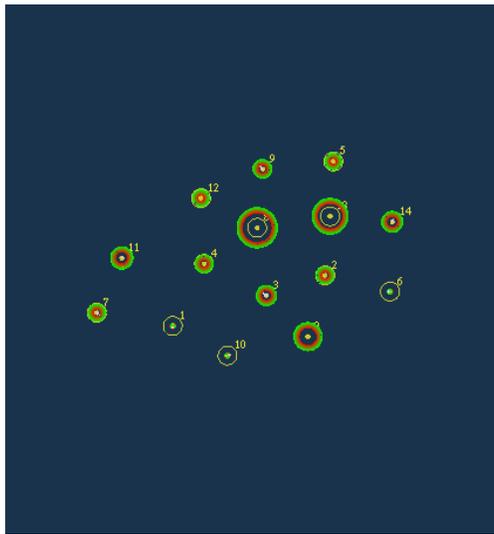


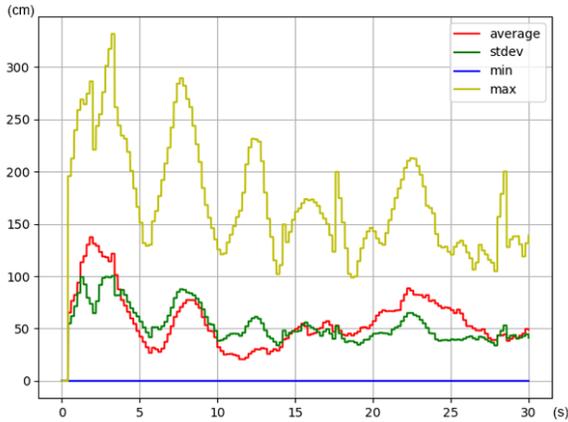
Figure 5.9: Initialization of the experiment with 15 agents.

The Transient State: This state corresponds to the initial response of the models after the constrained initialization. The first difference is the trend of the average pressure (in red). While it keeps rising for the VAT model, it almost instantaneously decreases for the APR model. This property is representative of the preventive action of the pressure regulation. Regarding the value, both models have almost the same maximum pressure (in yellow) but it is on average twice as superior for the VAT model (in red).

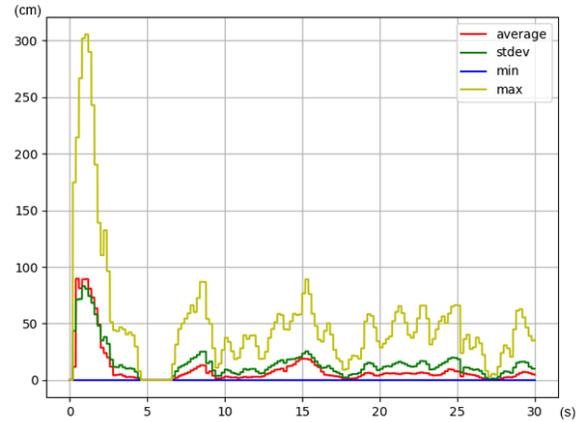
The second difference concerns the time of this transient state. One can see that while the VAT model struggles to reduce the average pressure (in red) to a “steady-state” in 6s, the APR model manages to vanish the average pressure in only 3s which is twice less for the same amount of pressure initially. This fast response is representative of the corrective action of the pressure regulation.

The Steady State: This state corresponds to the rest of the experiment where we expect the agents to reach a steady state after the transient state. One more time, the average pressure (in red) of the VAT model is much higher than the one of the APR model which is almost null. Also this average pressure of the APR model remains in a bounded and small range of values which are negligible in comparison with the VAT model. These small values of the APR pressure can be caused by the noises and delays (*e.g.* localization measures and communication delays) introduced in the simulated environment. As a perspective, it will be interesting to study an integral controller to see if we could get rid of it.

From this analysis we can see the advantages of measuring and exploiting the pressure. However, even though there are no obstacles in this experiment, there is still some pressure and it can be explained by the noises and delays as we mentioned above but also by the attraction force. But what is the real impact of these elements ? In Fig. 5.11, we have conducted the same experiment as before but without the attraction and only for the APR

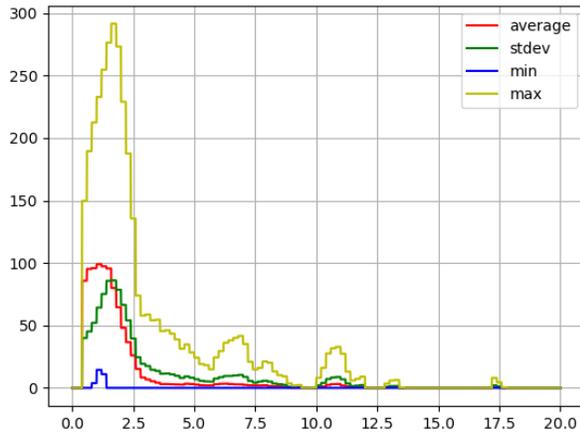


(a) Measure of the pressure with the VAT model

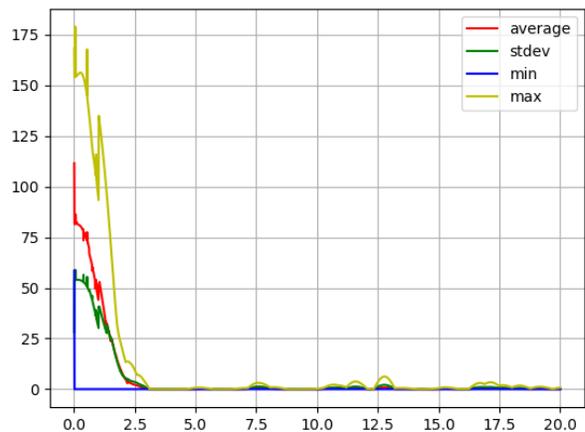


(b) Measure of the pressure with the APR model

Figure 5.10: Measure of the pressure (y axis) over time (x axis). The red, green, blue and yellow lines respectively represent the average, the standard deviation, the minimum and the maximum of the pressure among all the agents.



(a) Measure of the pressure with the APR model with no attraction in **realistic** conditions (with noises and delays)



(b) Measure of the pressure with the APR model with no attraction in **idealistic** conditions

Figure 5.11: Measure of the pressure (in cm) over time (in sec). The red, green, blue and yellow lines respectively represent the average, the standard deviation, the minimum and the maximum of the pressure among all the agents.

model. The goal is to observe the impact of the simulated conditions regarding the pressure generation when the attraction is null. The figure on the left corresponds to the measure of the pressure without attraction in **realistic** conditions with a delay on the communication of $t^{comm} = 0.2s$ and a refresh rate of the localization sensor (GNSS) equals to $t^s = 0.2s$. Also

the sensor noise is $\sigma^s = 50\text{cm}^2/\text{s}^2$. The figure on the right, on the other hand, corresponds to the same experiment with all the previous values set to zero also called **idealistic** conditions. In both figures, we can see that the average pressure (in red) vanishes quickly to a negligible level and remains as such during the steady state. But in realistic conditions, we observe some sudden variations in the maximum measured pressure (in yellow). These variations do not represent an actual risk of collision as they are small but can explain the one we see in Fig. 5.10.

From this experiment, we can conclude that the attraction is mostly responsible for a steady pressure but is yet necessary to avoid the fragmentation. The observed pressure is however bounded in an acceptable range of values and hence does not harm the behavior of the model.

Reaction towards obstacles

In Sec. 5.6.3, we saw that the APR model was subject to some collisions with the obstacles whereas the VAT model was more likely to generate some collisions among the agents. Even though both models share the same obstacle avoidance term, they do not have the same parameters setting but we know that it is efficient at least in the optimization environment. To better understand why collisions happen, we have to analyze the global behavior of the flock when it approaches an obstacle. As we saw in Chap. 4, the VAT model suffers from the superposition phenomenon (see Sec. 4.4.1) leading the agents to frequently evolve under the equilibrium distance resulting in some pressure. Hence, whenever the flock is constrained by an obstacle, the pressure keeps rising leading sometimes to collisions among the agents. We can see this effect in Fig. 5.12 on the left. On the contrary, agents with the APR model have more space between them and as we can see on the right picture of Fig. 5.12, the major part of the pressure is located on the agents close to the obstacles. This acts as a warning sign for the other agents that are therefore repelled from this area. The obstacle collision risk is however still high for these pressured agents close to the wall. Indeed, because of the measurement noises and communication delays, some pressure might be generated within the flock, leading to a repulsive action potentially pushing the agents close to the wall even closer to it. This mechanism appears to be the main reason why there are some collisions with the obstacles. However, the APR model remains interesting as it acts in a preventive way for agents that have not yet detected the obstacles.

We have shown through simulations that the APR model could not only drastically reduce the fragmentation phenomenon but also prevent any collision among the agents. Furthermore, it outperforms on most of the metrics the VAT model which was specifically designed for constrained environments. Despite its good performances, the APR model appears to give rise to some collisions with obstacles. We believe it could be interesting to integrate the pressure measure for the obstacle avoidance as it has given excellent results for the collision avoidance among the agents.

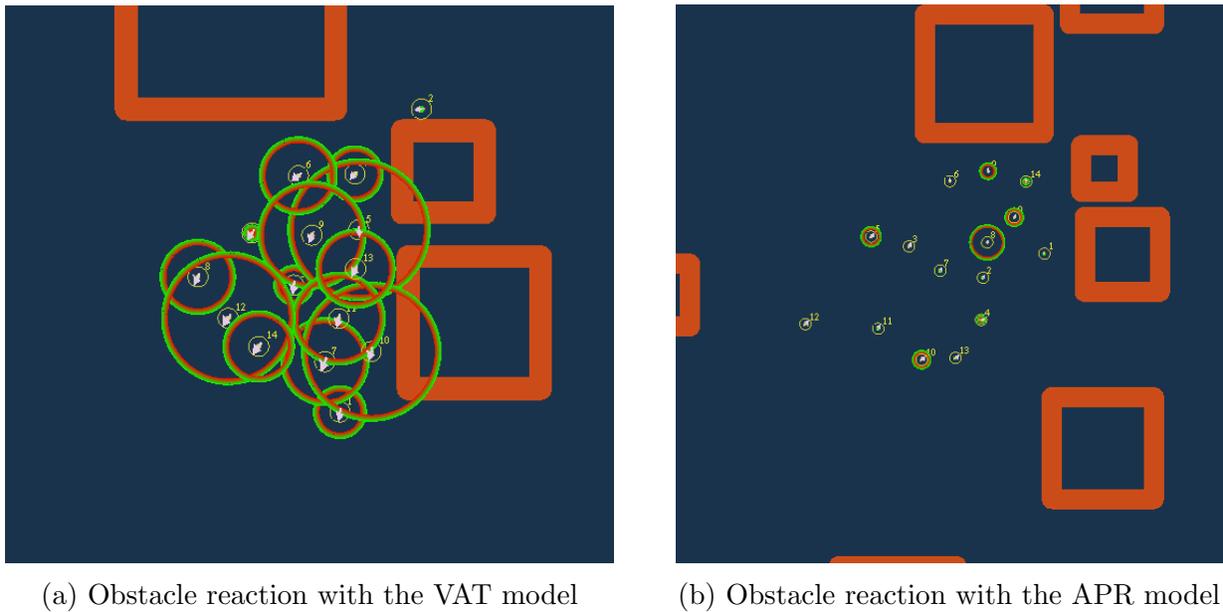


Figure 5.12: Behavior of the models towards obstacles

Given the good performances of this model, it could be interesting to add some functionalities like guiding the swarm toward a defined goal. In the next chapter, we develop a distributed leader follower strategy in order to satisfy this purpose. We will see that the APR model is well suited for this application and has better performances than other models such as VAT.

Throughout the previous chapters, we aimed at developing new flocking models that make it possible for self-propelled agents to evolve safely in constrained environments. One major hypothesis of this work was the absence of preferred direction or common goal that would be shared by all the agents. However, this usually means that our system is not suited for applications like target tracking or leader follower. In this chapter, we aim at coping with this limitation by introducing a distributed leader follower algorithm. We present an incremental leader-follower model, inspired from hierarchical leadership and exploiting inter-agent communications. This model is evaluated in particular in "tunnel" crossing missions.

6.1 Context and Objectives

Given the performances of the APR model, we decided to integrate it with a leader follower approach so as to go through highly constrained environments with a defined goal. The main objective is to use the APR model to ensure a safe and cohesive motion and a leader follower model in order to lead the swarm toward a defined position.

Leader follower models have been widely studied in the literature. Among them we can cite the well known work of Olfati-Saber [10] that we have been using in this thesis. In his work, the author uses a distributed navigational feedback term in order to enhance the cohesion among the swarm and avoid fragmentation. He shows that it is a necessary condition for the emergence of a flocking motion independently of the initial conditions. However, as we mentioned earlier, this requires a global information where the position and velocity of the virtual leader is known by all the agents. The same concept is used by La and Sheng in [50]. In [85], Balázs et al. introduce a time-dependent leadership hierarchy that adapts locally to agents' intention of changing direction. In other words, the leader follower relationship adapts itself over time. Doing so, they manage to enhance the responsiveness of standard collective motion models. Other articles such as [27, 86] consider a distributed leader follower flocking control where the states of the virtual leader are partially observable but also not always known by all the agents in the swarm. Last but not least, in [12], the author studies the emergent behavior of “Cucker-Smale” flocking under *hierarchical leadership* and shows the convergence toward cohesive patterns. The *hierarchical leadership* (HL) introduces the notion of sub-leaders sharing the role of both a leader and a follower. In Fig. 6.1 we can see the difference between a classical leader follower approach on the left where the two followers (in green) can only follow the leader (in red) whereas the HL on the right allows a follower to become a leader or sub-leader (in green) and a follower to have multiple leaders (in blue). An explicit definition of the HL will be given in the next section.

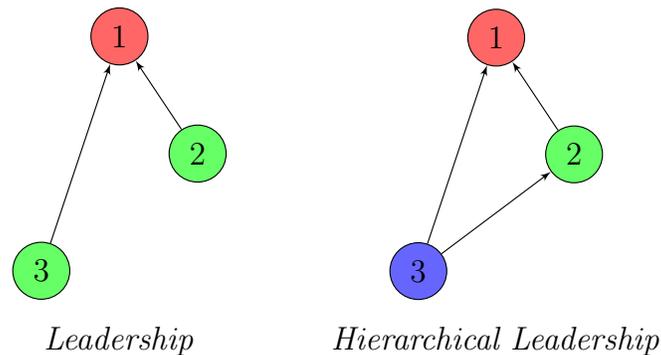


Figure 6.1: Graphical representation of two leadership approaches: on the left the classical one and on the right the hierarchical one.

In this chapter, we use the *hierarchical leadership* concept on top of our APR model with only one leader or sub-leader per follower. The underlying idea is to guide one agent toward

the most informed agent in its neighborhood. This distributes the information in a local way and thus we do not violate the hypothesis regarding no use of global information. Doing so we aim at ensuring the scalability of the model and improve the overall “guidance” of the swarm. The system remains fully decentralized assuming that the leader is itself autonomous. In case if the leader is remotely controlled, then the rest of the swarm is decentralized but not the leader. In the coming sections, we show how we define our tracking force and our incremental leader follower algorithm in order to fit in the APR model. Simulation results and comparison with the VAT model and a classical leader follower model will also be presented.

6.2 Incremental Leader Follower Strategy

In this section, we present a new leader follower strategy. The idea is to prevent the fragmentation phenomenon that usually occurs when the communication links between the leader and the followers are broken. We aim at allowing followers to become leaders in order to preserve cohesion and movement. The interaction terms as well as the leader follower algorithm are presented hereafter.

6.2.1 Navigational feedback

The navigational feedback term is used to define the dynamics between a follower and a leader. It can be seen as a tracking force attracting the follower toward the leader. The leader being the objective to reach, the navigational feedback is associated to the tracking force. In order to fit our framework, we define the **tracking force** by a velocity \mathbf{v}_i^{trg} as follows:

$$\mathbf{v}_i^{trg} = v_0 * s(\|\mathbf{r}_{trg}^i - \mathbf{r}_i\|, l, d) * \frac{\mathbf{r}_{trg}^i - \mathbf{r}_i}{\|\mathbf{r}_{trg}^i - \mathbf{r}_i\|} \quad (6.1)$$

where v_0 is the desired flocking speed, \mathbf{r}_{trg}^i is the position of the target of agent i and \mathbf{r}_i is the position of agent i . $s(x, l, d)$ is the sigmoid function with parameter l and d which represent the boundaries of the sinusoidal pattern:

$$s(x, l, d) = \begin{cases} 0, & \text{if } x \in [0, l] \\ \sin\left(\frac{\pi}{d}(x - l) - \frac{\pi}{2}\right) + 1, & \text{if } x \in [l, l + d] \\ 1, & \text{if } x > l + d \end{cases} \quad (6.2)$$

This tracking term is then added to the desired speed from our APR model (Eq.5.6):

$$\tilde{\mathbf{v}}_i^d(t + 1) = \frac{\mathbf{v}_i(t)}{\|\mathbf{v}_i(t)\|} v^{flock} + \mathbf{v}_i^{press}(t) + \mathbf{v}_i^{att}(t) + \mathbf{v}_i^{frict}(t) + \sum_s \mathbf{v}_{is}^{wall,obst}(t) + \mathbf{v}_i^{trg}(t) \quad (6.3)$$

As before, this velocity is bounded by v^{max} according to the following expression:

$$\mathbf{v}_i^d = \frac{\tilde{\mathbf{v}}_i^d}{\|\tilde{\mathbf{v}}_i^d\|} \min\{\|\tilde{\mathbf{v}}_i^d\|, v^{max}\}, \quad (6.4)$$

where \mathbf{v}_i^d is the final desired velocity of an agent i and v^{max} is the maximum allowed velocity. This final velocity concatenates both the navigational feedback and the flocking behavior induced by the APR model which is what we wanted in the first place. This resulting model will be called **APRT** model for *asymmetric pressure regulation and tracking* model.

In this thesis, we assume that the **leader** is not static and behaves as one of the agents but without the alignment and the attraction terms. Indeed, as it is attracted by a target or heading in a known direction, it does not need to align with the rest of the swarm nor to stay close to them. However the pressure repulsion remains and acts as a fail-safe in case if the rest of the swarm would reach close enough to the leader. Its velocity $\tilde{\mathbf{v}}_{leader}^d$ can be described as follows:

$$\tilde{\mathbf{v}}_{leader}^d(t+1) = \frac{\mathbf{v}_{leader}(t)}{\|\mathbf{v}_{leader}(t)\|} v^{flock} + \mathbf{v}_{leader}^{press}(t) + \sum_s \mathbf{v}_{leader/s}^{wall,obst}(t) + \mathbf{v}_{leader}^{trg}(t) \quad (6.5)$$

All the terms here are the same as before, we just use the subscript “leader” instead of i as it is unique. Regarding the term $\mathbf{v}_{leader}^{trg}$, it is the same as in Eq. 6.1 but the parameters might be different and the target position is initially known by the leader. This target can be static or dynamic. Indeed because we leave the self propelling term for the leader, even though it reaches a static target, it will still wander around it without reaching a stable position. This perpetual movement is not trivial as it prevents the followers from falling into a dangerous equilibrium state where they would all struggle to reach the same position (the one of the leader) while being constrained by the pressure repulsion.

In addition to this perpetual motion, another aspect of this model acts in our favor and relies on the leadership definition. In our model, as we will see in the next section, any follower can become a leader depending on its position and its “rank” regarding the swarm and the actual leader. Thanks to this, all the agents in the swarm may have a different leader and hence will not aim toward the same position.

6.2.2 Incremental leadership

In order to define a model robust to communication failures but also prevent congested motion (especially in narrow paths), we propose to increase the number of leaders. Selecting the appropriate leader is a key aspect of our incremental leader follower strategy. From what we have built in the previous section, we allow each agent to have a specific leader potentially different from the actual leader. The structure we use to assign leaders is inspired from the *hierarchical leadership* structure (see [12]). We begin by giving this definition:

Definition 6.2.1 (Hierarchical leadership flock [12]). A $(k+1)$ -flock with adjacency matrix $A = [a_{ij}]$ is said to be under hierarchical leadership if the agents can be labeled $[0, 1, \dots, k]$, such that

- (i) $a_{ij} = a_{agent\ i\ led\ by\ j} \neq 0$ implies that $j < i$; and

(ii) if we define the leader set of each agent i by

$$\mathcal{L}(i) = \{j \mid a_{ij} > 0\}, \quad (6.6)$$

then for any $i > 0, \mathcal{L}(i) \neq \emptyset$

If so, the flock is called an **HL**-flock (Hierarchical Leadership).

From this definition we can see that all the agents are subject to some leadership except for agent 0 which is the actual leader. In Fig. 6.2, we can see three examples of flock structures, two **HL**-flocks and one non-**HL**-flock due to the fact that agent 2 has no leader.

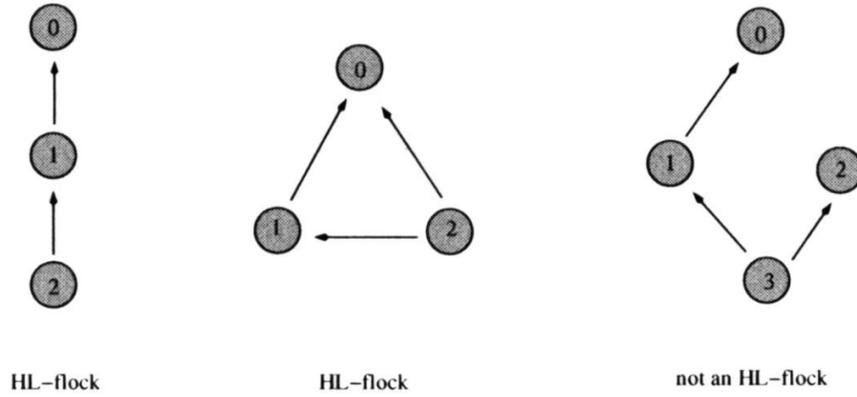


Figure 6.2: Two examples of **HL**-flocks and one example of a non-**HL**-flock. The arrow $i \rightarrow j$ represents that agent i is led by agent j . Credits: [12].

This structure is interesting and leads to good results as it is shown in [12] but is yet not suited for our objective. Indeed we observed in simulation that having multiple leaders could lead to unwanted situations where the follower would be dragged towards opposite directions. This situation cannot happen if the follower has only one leader. We propose to explore an approach where each agent follows a unique leader, which is selected has being the closest to the true leader. To do so, we introduce a hierarchy, inspired from HL-flock, computed as a distance from the true leader. We define this hierarchy by a value lr which is the topological distance to the true leader.

We formalize this proposition by the following definition characterizing our *incremental leadership* flocks.

Definition 6.2.2 (Incremental leadership flock). A N -flock with adjacency matrix $A = [a_{ij}]$ is said to be under incremental leadership if the agents can be ranked with a value lr_i in $[0, N - 1]$, such that

- (i) $lr = 0$ is reserved for the true leader and will never change
- (ii) $\forall j \in [0, N - 1]$ and $j \neq i$, if $a_{ij} = 0$ (i.e. no neighbors) then $lr_i = N - 1$

(iii) if agent i is the follower of the leader j then

$$lr_i = lr_j + 1 \quad (6.7)$$

(iv) each agent $i \neq 0$ has only one leader.

If so, the flock is called an **IL**-flock (Incremental Leadership).

According to this definition, we can infer an algorithm that will turn a flock into a **IL**-flock. The true leader is initialized with a rank equal to 0, $lr = 0$. The rest of the agents are all initialized to a rank equal to the number of agents in the system minus one as it is the maximum rank reachable. Because the system is decentralized and based on local interactions and asynchronous communications, an agent cannot know its exact rank instantaneously but will rather increment it according to the one of its neighbors. When the algorithm starts, each agent will check in its communication neighborhood if there are agents with a lower rank. If yes, this agent updates its rank to the lowest one of these neighbors plus one. It also uses this neighbor position as the target position and will hence head in this direction. This procedure is known as the *leader assignment* algorithm (LA) and is presented in Alg. 1. In this algorithm, \mathbf{q}_{trg}^i is the position of the target of agent i , \mathbf{q}_i is its position, \mathcal{N}_i is its communication neighborhood and lr_i is its leading rank.

Algorithm 1 Leader Assignment Algorithm (for agent i): $LA(i)$

Input: \mathcal{N}_i

Output: lr_i, \mathbf{q}_{trg}^i

```

1:  $lr_i \leftarrow N - 1$  ▷ leading rank is initialized by default to nb. of agents
2:  $\mathbf{q}_{trg}^i \leftarrow \mathbf{q}_i$ 
3: for  $j \in \mathcal{N}_i$  ordered by distance do ▷ loop on the neighbors
4:   if  $lr_j < lr_i$  then
5:      $lr_i \leftarrow lr_j + 1$  ▷ new level
6:      $\mathbf{q}_{trg}^i \leftarrow \mathbf{q}_j$  ▷ new target assignment
7:   end if
8: end for
9: return  $lr_i, \mathbf{q}_{trg}^i$ 

```

The overall algorithm that is presented in Alg. 2 consists of an infinite loop where all the agents compute their leader using the LA algorithm before applying the flocking strategy referred to as $FlockingMotion()$ in the algorithm. We call this algorithm the *incremental leader follower* algorithm (ILF).

With these algorithms, we allow each agent to become a leader as long as it is connected to a leader itself. This increases the number of leaders in a distributed way and we will see in the next section how it enhances the cohesion and thus prevents the fragmentation. In Fig. 6.3, we illustrate how the ILF algorithm works. We observe the resulting graph representation after applying the ILF algorithm on a swarm of 7 agents (lr values are represented by colors).

Algorithm 2 Incremental Leader Follower Algorithm

Output: lr_i, \mathbf{q}_{trg}^i

```
1: while True do ▷ infinite loop  
2:   for  $i \leftarrow 1$  to  $N$  do ▷ loop on the agents  
3:      $lr_i, \mathbf{q}_{trg}^i \leftarrow LA(i)$   
4:      $FlockingMotion()$   
5:   end for  
6: end while
```

We can clearly see that each agent has only one leader with a higher leading rank except for the actual leader (agent 1) which does not have any leader. In Fig. 6.4, we can see the same

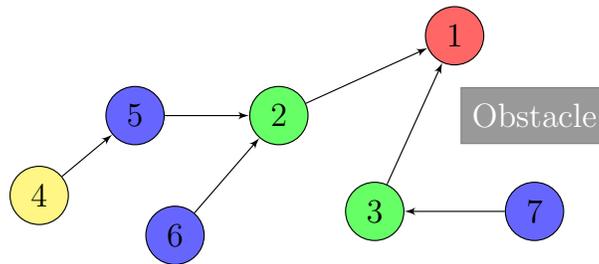


Figure 6.3: Representation of the graph with the incremental leader follower algorithm. The colors represent the leading rank with $lr = 0, lr = 1, lr = 2, lr = 3$ respectively for the red, green, blue and yellow colors.

concept in the simulator where the agents are moving in a narrow path. The red leading agent is the real leader ($lr = 0$), the only one having the true information about the target. The green ones ($lr = 1$) are the followers of the red agent but also leaders of the blue ones ($lr = 2$) themselves leaders of the yellow ones ($lr = 3$). It is important to note here that because of this environment, the communication is restrained especially between the leader and almost all the rest of the agents. Without such intermediate leaders, it would be less likely that agents would be able to follow the leader. If the information of the leader were globally known, like in [10] for example, the yellow agents would directly follow the red agent, it would increase the risk of hitting the wall before reaching the turn or lead them toward a deadlock depending on the rest of the environment. By allowing a follower to become a leader, we somehow propagate the target information through intermediate way-points and headings. Even though the information of the actual leader is not known by all the agents, they at least know where to go next in order to get closer to it. The ILF algorithm breaks down the global attraction towards the leader into a “*piecewise attractors*” which respects our aim to have decentralized models.



Figure 6.4: Swarm of 16 agents applying the ILF algorithm in a narrow corridor. The colors represent the different leading ranks.

6.3 Simulation and Comparison

In this section, we study the performances of the ILF algorithm applied to different leader-follower flocking models: the APRT model and the VATT model which is an extended version of the VAT model with the tracking component presented in Eq. 6.1. To do so, we present two scenarios. In the first one we compare the results of the ILF applied to the APRT and the VATT model. The idea is to study the impact of the ILF on both models. The second scenario compares the ILF algorithm to a regular leader follower model such as the one of Virágh et al. [1]. Both are applied to the APRT model.

A major aspect of these experiments is the complexity of the environment where agents evolve. It consists of a narrow S-shaped corridor as shown in Fig. 6.5. The green circle represents the entrance of the corridor and the red circle is the exit. The idea is to make the agents go through the corridor by using the ILF algorithm and a regular leader follower algorithm and measure the rate of success. **The rate of success is equal to one when all the agents have reached the red circle and zero otherwise.** We also measure the number of collisions between the agents. The environment being made of two non-convex obstacles resulting in an highly constrained path, the simulations might lead to collisions with the obstacles. Indeed, both APRT and VATT share the same interaction term regarding obstacles avoidance and this latter is not specifically designed for concave obstacles. In these experiments, 15 agents are simulated. Only one is the leader and knows the position of the exit of the corridor (the red circle in Fig. 6.5) and the other ones are followers. The leader is always initialized ahead of the followers in order to be able to go through the corridor

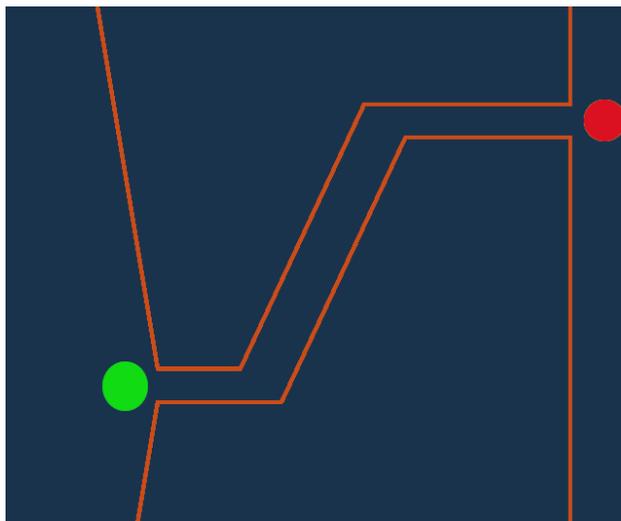


Figure 6.5: S-shaped environment.

without any trouble.

6.3.1 APRT and VATT

From our previous observations, we have seen that the APR model had the capacity of deforming the swarm without affecting the cohesion which is a major asset when moving in tight spaces. On the contrary, the VAT model is more likely to form a compact swarm preventing fragmentation but increasing the risk of collisions especially in the environment we use here. The difference of behaviors can be seen in Fig. 6.6 and 6.7 where the ILF algorithm is being applied to both models, respectively APRT and VATT, in order to make them go through a narrow corridor. In these snapshots, the red agent corresponds to the true leader ($lr = 0$) and the other ones are the followers ($lr \neq 0$). We can clearly see the difference between them, especially with the shape of the fleet. The VATT model seems to be highly constrained leading all the agents to an extreme cohesion with an high risk of collision whereas for the APRT model the flock looks more elongated and less constrained with consequently a lower risk of collision. For both models, the optimized version, trained on the *generic* environment, has been used but the models have not been optimized on the S-shaped environment. As we mentioned before, we measure two metrics here:

- Success: All the agents reached the end of the corridor.
- Collisions: Count the number of collisions among agents.

In this experiment, the agents are all initialized in a fixed size circle before the entrance of the corridor (close to the green circle of Fig. 6.5). After a certain time, a way-point is set on the red circle (Fig. 6.5) which starts the ILF algorithm and attracts the true leader towards the exit of the corridor.

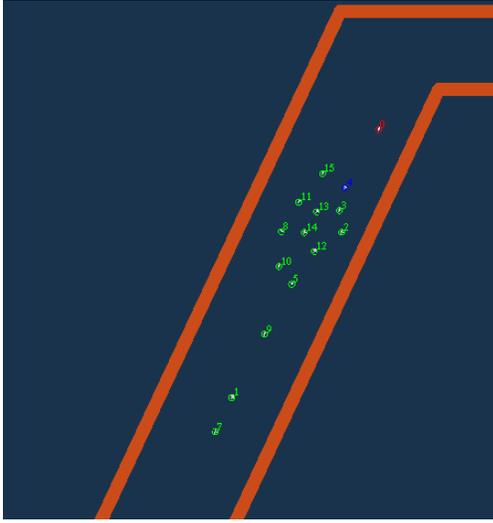


Figure 6.6: APRT with incremental leader follower



Figure 6.7: VATT with incremental leader follower

Table 6.1: Performances of VATT and APRT with the incremental leader follower algorithm.

	Rate of Success	Collisions
APRT	96%	0
VATT	83%	0.28 ± 0.67

The performances of the models are being evaluated over a hundred runs and are gathered in Tab. 6.1.

You can see the behavior of the swarm applying the ILF algorithm in the video attached to this URL: <https://youtu.be/zKcpsBHAmAg>.

From Tab.6.1, we can see that the APRT model has a better rate of success than VATT and produces no collision among the agents which is a fundamental property of the APR model. As a consequence, we can say that the APRT model seems better suited to the ILF algorithm than the VATT model. The ILF algorithm is yet not responsible for the collisions of the VATT model as we know from the previous chapters that it leads to collisions on its own. In addition, when we observe the simulations we can see that some collisions with the wall occur at the entrance of the corridor which appear to be a very challenging sequence and is relatively dependent of the initialization of the agents. This part is also responsible of most of the failure as the leader goes quickly inside the corridor and may leave behind all the agents while they reshape to fit in as it can be seen in Fig. 6.8. Beside this drawback, the APRT model seems to be promising with the new incremental leader follower algorithm and it is quite easy to think about the applications that could be based on it such as guided exploration or tracking of an external target for example.

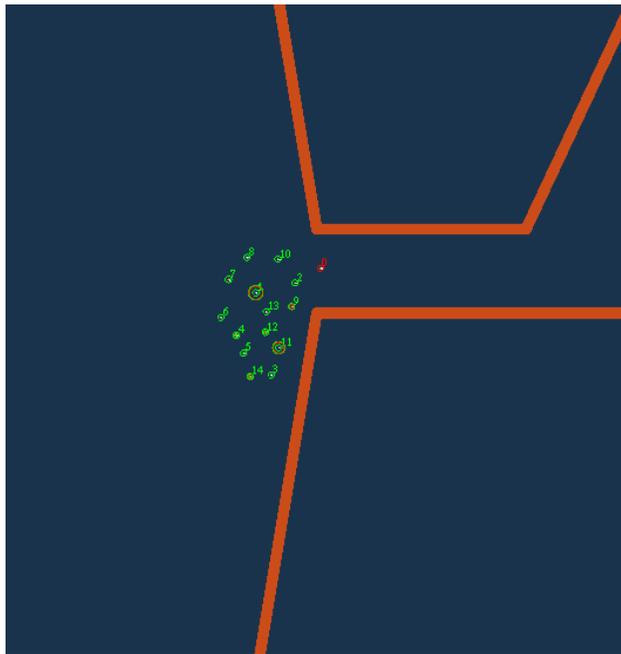


Figure 6.8: Swarm of drones at the entrance of the corridor (leader in red color)

6.3.2 ILF algorithm and classical leader follower algorithm

In order to validate the ILF algorithm, it must also be compared to a classical leader follower algorithm (noted CLF). For this, we use the control law developed by Virágh et al. in [1]. It consists of the same equation presented in Sec. 6.2.1 (see Eq. 6.1). The only difference is that the leadership is no longer distributed here which means that there is only one leader for all the followers and the interaction exists only if the leader is in the communication neighborhood of the follower. The difference can be seen in Fig. 6.9 where the ILF algorithm is applied resulting in multiple leadership layers whereas in Fig. 6.10, the CLF is applied and we can see only four agents in green being attracted by the leader in red. The rest of the swarm only relies on the attraction term of the APRT model to follow their peers.

So as to properly measure the performances of these two algorithms, the agents are initialized in the exact same position with a null velocity at each simulation. When the experiment starts, the way-point attracting the leader to the exit of the corridor is already set. Hence the swarm directly attempts to go into the corridor, following the leader. We have run the simulation a hundred time for both configurations. The rate of success and the number of collisions are gathered in Tab. 6.2. The results are quite significant with a rate of success of 100% for the ILF algorithm and 20% for the CLF algorithm. Not only is the ILF algorithm 400% more efficient than the CLF algorithm, but it also never fails to reach its goal. Last but not least, there is no collision, as it could be expected when using the APRT model.

From these results we can clearly state that the ILF algorithm has numerous advantages

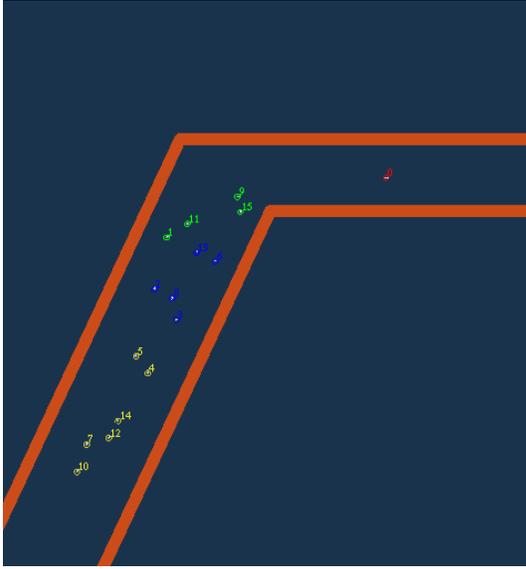


Figure 6.9: APRT with ILF algorithm.

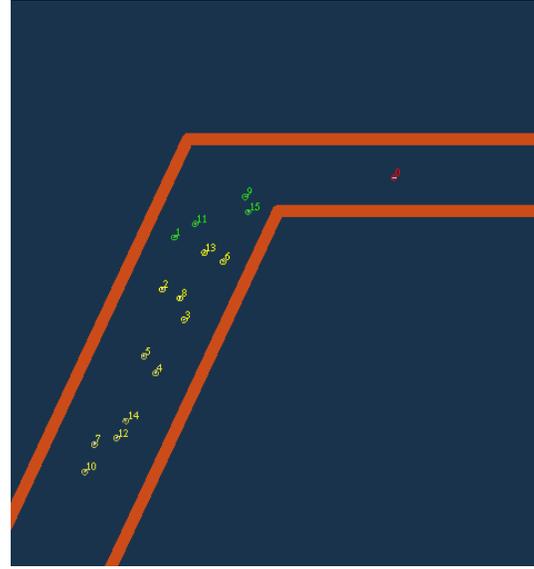


Figure 6.10: APRT with CLF algorithm.

Table 6.2: Performances of ILF and CLF algorithm with APRT model.

	Rate of Success	Collisions
ILF	100%	0
CLF	20%	0

namely its adaptability to existing flocking models (requiring minimal modifications) and the performances it produces in highly constrained environments. The propagation of the leadership allows to get rid of the limits regarding the number of followers and improve the robustness of the swarm.

7.1 Synthesis

The initial goal of this thesis was to understand flocking motion and develop new models based on local communications in swarm of UAVs. The physical environment within which the UAVs evolve in rapidly became an important aspect of this thesis after realizing the impact it had on the global behavior of the swarm. Eventually, the main objective during these three years has been to identify the different flaws of existing flocking models in highly constrained environments and develop new models with better performances. We believe this work shows that we managed to do so but it is important to remember that it relies on a specific simulator and the hypothesis we took remains to be confirmed with other frameworks and real UAVs. In this direction, note that the VAT and APR models have been recently implemented in the GAZEBO simulator with the IntelAero drone model, by Johan Faure, engineer in the Chroma team. First results, illustrated in Fig 7.1.a show that the main properties are kept and the simulated flocking is efficient. Moreover, the team has also started to implement and test flocking models with real PX4 Vision UAVs (see Fig 7.1.b). However, this is only the beginning of a new phase of evaluation of the different flocking models proposed.

To come back to this thesis, our main contributions can be summed up into the following threads:

1. We showed the limits of existing flocking models. In environments with many obstacles, state of the arts models appear to generate unwanted phenomena such as fragmentation and collision. These situations occur under specific yet realistic conditions that we fixed in our work such as the consideration of more realistic communications, the removal of any centralized information or the exclusive use of self-propelled aerial robots. This led to extend the flocking simulator of Virág in Chap. 3.
2. In order to cope with the limitations we highlighted in the previous point, we extended two flocking models from the literature. We managed to significantly improve their performance in various environments. This led to define the models VAT and VOS in Chap. 4. In particular, we added an attraction force to the initial model of Vásárhelyi



Figure 7.1: a. Snapshot of 5 IntelAero drones flocking together with the VAT model in the GAZEBO simulator b. Experiment of a 3-drones flocking by the Chroma team on April 2023 (CITI lab.)

et al. [2] making it more robust and less prompt to fragmentation (VAT). Regarding the VOS model, we merged both the model of Vásárhelyi et al. and the one of Olfati-Saber [10], taking advantage of their performances.

3. We developed a new flocking model mostly based on asymmetric interactions. It relies on a novel measure of pressure that concatenates crucial information regarding the neighborhood of an aerial robot. It also includes an adaptive attraction force that efficiently selects the agents used for the computation of the force. This model, called APR, leads to some serious improvements with a collision-free behavior in all the environments we designed.
4. From the perspective of developing flocking based applications, we proposed a leader-follower algorithm that aims at leading the flock toward a given destination. It benefits from an adapted hierarchical leadership model. This new algorithm called *incremental leader follower* can be easily stacked on all flocking models and proved to be much more efficient than a classical leader follower algorithm

Most of these contributions result from a huge empirical work that have been made possible by the use of the simulator developed by Virágh et al. [1]. All our simulations have been conducted in this simulator that we progressively enhanced in order to suit our different needs such as the integration of a communication model, the measure of the pressure or the on-line computation and display of the clusters. These functionalities, mostly visual, greatly helped us in the understanding of the behaviors of the aerial robots. Besides this aspect of the simulator, we also managed to integrate it into an optimization process that happened to be essential in the development of new flocking models. Through this step, we also learned how to properly measure and asses the performances of a model.

The work done during this three years has resulted in several models, in particular the *asymmetric pressure regulation* model which broadly answers to the main problematic of this thesis. New questions have yet arisen from this study like the stability of the control law or the different ways to interact with obstacles. These questions are essential and need to be studied further as to build more and more robust and efficient flocking systems.

7.2 Perspectives

As we progressed in this thesis, we thought about many enhancements and tools that could be developed to improve our flocking models but also the general framework including the simulator and the optimization strategy. We also encountered some obstacles that remain to be dealt with. The major perspectives of these work can be summarized into the following items:

- The *asymmetric pressure regulation* model, which is the most promising model, concatenates two asymmetric terms resulting in a global asymmetric control law. This asymmetric property makes it very difficult to prove the stability of the model. It appears that the model oscillates between symmetrical and asymmetrical phases depending on the configuration of the swarm. Studying this stability could be very interesting in order to prove the robustness of the APR model which has, until now, only be “proven” empirically.
- Integrating a communication model into the simulator has been one of the contributions of this thesis. It has yet showed us the limits of such models and the difficulties of accurately modeling radio waves. As explained in research articles such as the one of Vásárhelyi et al. [2], communications are seriously challenged in outdoor experiments and not as reliable as we could model it. Paring network and robotic simulators in a long-lasting way would be a major advance that would be beneficial for both communities.
- In all our models, we used the same term for the obstacles avoidance. This choice was made because of the good performances it led to on various models and situations. However, we saw at the end of this thesis that it is not flawless and some situations lead to collisions. This is clearly not acceptable and put at risks our wish for real experiments. By looking closely, these collisions happen in highly constrained environments and under harsh, almost unrealistic conditions. In such situations, we are confident that the UAVs we plan to use would not collide with any obstacles as they integrate fail-safes, taking over control to avoid any harm. However, this action usually interrupts the motion of the UAVs. This is why we need to develop safer obstacle avoidance methods, that better fit in the flocking behavior without impacting the agility it provides.

- Thanks to the *incremental leader follower* algorithm, we understood that multiple applications could benefit from the flocking. Even though we measured the performances of this algorithm, it still requires some investigation in order to prove its reliability and applicability. We are currently working on an advanced version where we limit the number of followers a leader can “have”. This could fluidify the motion and prevent dangerous situations where all the agents would be dragged by the real leader to a fixed position potentially leading to collisions.

Even though some aspects of the flocking remain to be deepened, we believe that the models we presented are ready to be used on swarm of UAVs. This validation step is crucial in the development of aerial systems but is yet very challenging and requires multiple contributions from the financial point of view as well as the technological one but also the legal aspect along with finding the appropriate facility where the UAVs could fly. We are currently on good tracks to start experimenting as we now have enough UAVs and an appropriate environment where they can fly safely.

While we listed the most interesting perspectives in our opinion, conducting real experiments, as we mentioned above, is obviously the ultimate validation that we can get. We believe that during these three years, we have paved the way to that end, by developing new models, underpinning the emergence of reliable, efficient, decentralized and bio-inspired behaviors.

Bibliography

- [1] Csaba Virágh, Gábor Vásárhelyi, Norbert Tarcai, Tamás Szörényi, Gergő Somorjai, Tamás Nepusz, and Tamás Vicsek. Flocking algorithm for autonomous flying robots. *Bioinspiration & Biomimetics*, 9(2):025012, May 2014. ISSN 1748-3182, 1748-3190. doi: 10.1088/1748-3182/9/2/025012. URL <https://iopscience.iop.org/article/10.1088/1748-3182/9/2/025012>.
- [2] Gábor Vásárhelyi, Csaba Virágh, Gergő Somorjai, Tamás Nepusz, Agoston E. Eiben, and Tamás Vicsek. Optimized flocking of autonomous drones in confined environments. *Science Robotics*, 3(20):eaat3536, July 2018. ISSN 2470-9476. doi: 10.1126/scirobotics.aat3536. URL <https://www.science.org/doi/10.1126/scirobotics.aat3536>.
- [3] @NatGeoUK. These birds flock in mesmerising swarms of thousands—but why is still a mystery., March 2021. URL <https://www.nationalgeographic.co.uk/animals/2021/03/these-birds-flock-in-mesmerising-swarms-of-thousands-but-why-is-still-a-mystery>. Section: Animals.
- [4] Intel Breaks Guinness World Records Title for Drone Light Shows in Celebration of 50th Anniversary, . URL <https://newsroom.intel.com/news/intel-breaks-guinness-world-records-title-drone-light-shows-celebration-50th-anniversary/>. Publication Title: Intel Newsroom.
- [5] Now that’s a big bird! Flock of thousands of starlings creates intimidating shapes to scare off prowling falcon \textbar Daily Mail Online, . URL <https://www.dailymail.co.uk/news/article-2144081/Now-thats-big-bird-Flock-thousands-starlings-creates-intimidating-shapes-scare-prowling-falcon.html>.
- [6] Xin Zhou, Xiangyong Wen, Zhepei Wang, Yuman Gao, Haojia Li, Qianhao Wang, Tiankai Yang, Haojian Lu, Yanjun Cao, Chao Xu, and Fei Gao. Swarm of micro flying robots in the wild. *Science Robotics*, 7(66):eabm5954, May 2022. ISSN 2470-9476. doi: 10.1126/scirobotics.abm5954. URL <https://www.science.org/doi/10.1126/scirobotics.abm5954>.

- [7] Relighting of the Tree Ceremony, . URL https://batman.fandom.com/wiki/Relighting_of_the_Tree_Ceremony. Publication Title: Batman Wiki.
- [8] Craig W. Reynolds. Flocks, herds, and schools: a distributed behavioral model. In *Seminal graphics*, pages 273–282. ACM, New York, NY, USA, July 1998. ISBN 978-1-58113-052-2. doi: 10.1145/280811.281008. URL <https://dl.acm.org/doi/10.1145/280811.281008>.
- [9] Tamas Vicsek, Andras Czirok, Eshel Ben-Jacob, Inon Cohen, and Ofer Sochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226–1229, August 1995. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.75.1226. URL <http://arxiv.org/abs/cond-mat/0611743>.
- [10] R. Olfati-Saber. Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, March 2006. ISSN 0018-9286. doi: 10.1109/TAC.2005.864190. URL <http://ieeexplore.ieee.org/document/1605401/>.
- [11] Gazebo, . URL <https://gazebo.org/home>.
- [12] Jackie (Jianhong) Shen. Cucker-Smale Flocking under Hierarchical Leadership. *SIAM Journal on Applied Mathematics*, 68(3):694–719, 2008. URL <http://www.jstor.org/stable/40233742>.
- [13] Andrea Cavagna, Alessio Cimorelli, Irene Giardina, Giorgio Parisi, Raffaele Santagati, Fabio Stefanini, and Massimiliano Viale. Scale-free correlations in starling flocks. *Proceedings of the National Academy of Sciences*, 107(26):11865–11870, June 2010. doi: 10.1073/pnas.1005766107. URL <https://www.pnas.org/doi/full/10.1073/pnas.1005766107>. Publisher: Proceedings of the National Academy of Sciences.
- [14] Kolbjørn Tunstrøm, Yael Katz, Christos C. Ioannou, Cristián Huepe, Matthew J. Lutz, and Iain D. Couzin. Collective States, Multistability and Transitional Behavior in Schooling Fish. *PLoS Computational Biology*, 9(2):e1002915, February 2013. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1002915. URL <https://dx.plos.org/10.1371/journal.pcbi.1002915>.
- [15] Sandrine Ngo, Francesco Ginelli, and Hugues Chaté. Competing ferromagnetic and nematic alignment in self-propelled polar particles. *Physical Review E*, 86(5):050101, November 2012. ISSN 1539-3755, 1550-2376. doi: 10.1103/PhysRevE.86.050101. URL <http://arxiv.org/abs/1206.3811>. arXiv:1206.3811 [cond-mat].
- [16] Fabian Schilling, Fabrizio Schiano, and Dario Floreano. Vision-based Drone Flocking in Outdoor Environments, February 2021. URL <http://arxiv.org/abs/2012.01245>. arXiv:2012.01245 [cs].

- [17] Viktor Walter, Nicolas Staub, Martin Saska, and Antonio Franchi. Mutual Localization of UAVs based on Blinking Ultraviolet Markers and 3D Time-Position Hough Transform. In 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), pages 298–303, August 2018. doi: 10.1109/COASE.2018.8560384.
- [18] Yitao Han, Liang Liu, Lingjie Duan, and Rui Zhang. Towards Reliable UAV Swarm Communication in D2D-Enhanced Cellular Network, February 2020. URL <http://arxiv.org/abs/2002.04897>.
- [19] Wang Shule, Carmen Martínez Almansa, Jorge Peña Queralta, Zhuo Zou, and Tomi Westerlund. UWB-Based Localization for Multi-UAV Systems and Collaborative Heterogeneous Multi-Robot Systems. Procedia Computer Science, 175:357–364, 2020. ISSN 18770509. doi: 10.1016/j.procs.2020.07.051. URL <https://linkinghub.elsevier.com/retrieve/pii/S1877050920317324>.
- [20] Qazawat Zirak, Dmitriy Shashev, and Stanislav Shidlovskiy. Swarm of Drones Using LoRa Flying Ad-Hoc Network. In 2021 International Conference on Information Technology (ICIT), pages 400–405, July 2021. doi: 10.1109/ICIT52682.2021.9491655.
- [21] Ariana Strandburg-Peshkin, Colin R. Twomey, Nikolai W. F. Bode, Albert B. Kao, Yael Katz, Christos C. Ioannou, Sara B. Rosenthal, Colin J. Torney, Hai Shan Wu, Simon A. Levin, and Iain D. Couzin. Visual sensory networks and effective information transfer in animal groups. Current Biology, 23(17):R709–R711, September 2013. ISSN 0960-9822. doi: 10.1016/j.cub.2013.07.059. URL [https://www.cell.com/current-biology/abstract/S0960-9822\(13\)00920-2](https://www.cell.com/current-biology/abstract/S0960-9822(13)00920-2). Publisher: Elsevier.
- [22] Iain D. Couzin, Jens Krause, Nigel R. Franks, and Simon A. Levin. Effective leadership and decision-making in animal groups on the move. Nature, 433(7025):513–516, February 2005. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature03236. URL <http://www.nature.com/articles/nature03236>.
- [23] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. Proceedings of the National Academy of Sciences, 105(4):1232–1237, January 2008. doi: 10.1073/pnas.0711437105. URL <https://www.pnas.org/doi/full/10.1073/pnas.0711437105>. Publisher: Proceedings of the National Academy of Sciences.
- [24] Francesco Ginelli and Hugues Chaté. Relevance of Metric-Free Interactions in Flocking Phenomena. Physical Review Letters, 105(16):168103, October 2010. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.105.168103. URL <http://arxiv.org/abs/1007.1783>.

- [25] Vijay Kumar and De Rumi. Efficient flocking: metric versus topological interactions. page 9, September 2021. doi: <http://doi.org/10.1098/rsos.202158>.
- [26] H.G. Tanner. Flocking with obstacle avoidance in switching networks of interconnected vehicles. In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, pages 3006–3011 Vol.3, New Orleans, LA, USA, 2004. IEEE. ISBN 978-0-7803-8232-9. doi: 10.1109/ROBOT.2004.1307518. URL <http://ieeexplore.ieee.org/document/1307518/>.
- [27] Hong Shi, Long Wang, and Tianguang Chu. Virtual leader approach to coordinated control of multiple mobile agents with asymmetric interactions. Physica D: Nonlinear Phenomena, 213(1):51–65, January 2006. ISSN 01672789. doi: 10.1016/j.physd.2005.10.012. URL <https://linkinghub.elsevier.com/retrieve/pii/S0167278905004604>.
- [28] Pratik Mukherjee, Andrea Gasparri, and Ryan K. Williams. Stable motion and distributed topology control for multi-agent systems with directed interactions. In 2017 IEEE 56th Annual Conference on Decision and Control (CDC), pages 3455–3460, Melbourne, VIC, December 2017. IEEE. ISBN 978-1-5090-2873-3. doi: 10.1109/CDC.2017.8264165. URL <http://ieeexplore.ieee.org/document/8264165/>.
- [29] Arthur E. B. T. King and Matthew S. Turner. Non-local interactions in collective motion. Royal Society Open Science, 8(3):201536, March 2021. doi: 10.1098/rsos.201536. URL <https://royalsocietypublishing.org/doi/10.1098/rsos.201536>. Publisher: Royal Society.
- [30] Yuhai Tu and John Toner. How birds fly together: Long-range order in a two-dimensional dynamical XY model, June 1995. URL <http://arxiv.org/abs/adap-org/9506001>. arXiv:adap-org/9506001.
- [31] Hang Sun, Juntong Qi, Chong Wu, and Mingming Wang. Path Planning for Dense Drone Formation Based on Modified Artificial Potential Fields. In 2020 39th Chinese Control Conference (CCC), pages 4658–4664, July 2020. doi: 10.23919/CCC50068.2020.9189345.
- [32] Balsam Alkouz, Athman Bouguettaya, and Sajib Mistry. Swarm-based Drone-as-a-Service (SDaaS) for Delivery, April 2020. URL <http://arxiv.org/abs/2005.06952>.
- [33] Enrica Soria, Fabrizio Schiano, and Dario Floreano. Predictive Control of Aerial Swarms in Cluttered Environments. preprint, In Review, September 2020. URL <https://www.researchsquare.com/article/rs-82503/v1>.
- [34] Afzal Ahmad, Viktor Walter, Pavel Petráček, Matěj Petrlík, Tomáš Báča, David Žaitlík, and Martin Saska. Autonomous Aerial Swarming in GNSS-denied Environments with High Obstacle Density. May 2021. doi: 10.1109/ICRA48506.2021.9561284. Pages: 576.

- [35] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9): 1520–1533, September 2004. ISSN 1558-2523. doi: 10.1109/TAC.2004.834113.
- [36] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks, December 2002. URL <http://arxiv.org/abs/math/0212212>.
- [37] G. Wen, Z. Duan, H. Su, G. Chen, and W. Yu. A Connectivity-preserving flocking algorithm for multi-agent dynamical systems with bounded potential function. *IET Control Theory & Applications*, 6(6):813, 2012. ISSN 17518644. doi: 10.1049/iet-cta.2011.0532. URL <https://digital-library.theiet.org/content/journals/10.1049/iet-cta.2011.0532>.
- [38] Sungjoon Choi, Kyungjae Lee, and Songhwai Oh. A multi-agent coverage algorithm with connectivity maintenance. In *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 512–517, Daegu, November 2017. IEEE. ISBN 978-1-5090-6064-1. doi: 10.1109/MFI.2017.8170372. URL <http://ieeexplore.ieee.org/document/8170372/>.
- [39] Maria Carmela De Gennaro and Ali Jadbabaie. Decentralized Control of Connectivity for Multi-Agent Systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3628–3633, San Diego, CA, USA, 2006. IEEE. ISBN 978-1-4244-0171-0. doi: 10.1109/CDC.2006.377041. URL <http://ieeexplore.ieee.org/document/4177054/>.
- [40] Hao Fang, Yue Wei, Jie Chen, and Bin Xin. Flocking of Second-Order Multiagent Systems With Connectivity Preservation Based on Algebraic Connectivity Estimation. *IEEE Transactions on Cybernetics*, 47(4):1067–1077, April 2017. ISSN 2168-2267, 2168-2275. doi: 10.1109/TCYB.2016.2537307. URL <http://ieeexplore.ieee.org/document/7434002/>.
- [41] Wenhao Luo, Nilanjan Chakraborty, and Katia Sycara. Minimally Disruptive Connectivity Enhancement for Resilient Multi-Robot Teams. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11809–11816, Las Vegas, NV, USA, October 2020. IEEE. ISBN 978-1-72816-212-6. doi: 10.1109/IROS45743.2020.9340733. URL <https://ieeexplore.ieee.org/document/9340733/>.
- [42] E. W. Justh and P. S. Krishnaprasad. Equilibria and steering laws for planar formations. *Systems & Control Letters*, 52(1):25–38, May 2004. ISSN 0167-6911. doi: 10.1016/j.sysconle.2003.10.004. URL <https://www.sciencedirect.com/science/article/pii/S0167691103002949>.
- [43] Tamás Vicsek and Anna Zafeiris. Collective motion. *Physics Reports*, 517(3-4):71–140, August 2012. ISSN 03701573. doi: 10.1016/j.physrep.2012.03.004. URL <http://arxiv.org/abs/1010.5017>.

- [44] Francesco Ginelli. The Physics of the Vicsek model. The European Physical Journal Special Topics, 225(11-12):2099–2117, November 2016. ISSN 1951-6355, 1951-6401. doi: 10.1140/epjst/e2016-60066-8. URL <http://link.springer.com/10.1140/epjst/e2016-60066-8>.
- [45] Guillaume Gregoire, Hugues Chate, and Yuhai Tu. Moving and staying together without a leader. Physica D: Nonlinear Phenomena, 181(3-4):157–170, July 2003. ISSN 01672789. doi: 10.1016/S0167-2789(03)00102-7. URL <http://arxiv.org/abs/cond-mat/0401257>.
- [46] Yongnan Jia and Tamas Vicsek. Modelling hierarchical flocking. New Journal of Physics, 21(9):093048, September 2019. ISSN 1367-2630. doi: 10.1088/1367-2630/ab428e. URL <https://iopscience.iop.org/article/10.1088/1367-2630/ab428e>.
- [47] Reza Olfati-Saber and Richard M. Murray. DISTRIBUTED COOPERATIVE CONTROL OF MULTIPLE VEHICLE FORMATIONS USING STRUCTURAL POTENTIAL FUNCTIONS. IFAC Proceedings Volumes, 35(1):495–500, 2002. ISSN 14746670. doi: 10.3182/20020721-6-ES-1901.00244. URL <https://linkinghub.elsevier.com/retrieve/pii/S1474667015386651>.
- [48] J.A. Fax and R.M. Murray. Information Flow and Cooperative Control of Vehicle Formations. IEEE Transactions on Automatic Control, 49(9):1465–1476, September 2004. ISSN 0018-9286. doi: 10.1109/TAC.2004.834433. URL <http://ieeexplore.ieee.org/document/1333200/>.
- [49] Herbert G. Tanner, Ali Jadbabaie, and George J. Pappas. Flocking in Fixed and Switching Networks. IEEE Transactions on Automatic Control, 52(5):863–868, May 2007. ISSN 0018-9286. doi: 10.1109/TAC.2007.895948. URL <http://ieeexplore.ieee.org/document/4200874/>.
- [50] H.M. La and Weihua Sheng. Flocking control of a mobile sensor network to track and observe a moving target. In 2009 IEEE International Conference on Robotics and Automation, pages 3129–3134, Kobe, May 2009. IEEE. ISBN 978-1-4244-2788-8. doi: 10.1109/ROBOT.2009.5152747. URL <http://ieeexplore.ieee.org/document/5152747/>.
- [51] Weihua Sheng and Ronny Salim Lim. Hybrid System of Reinforcement Learning and Flocking Control in Multi-robot Domain. page 8, 2010.
- [52] Alexandre Bonnefond, Olivier Simonin, and Isabelle Guerin-Lassous. Extension of Flocking Models to Environments with Obstacles and Degraded Communications. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 9139–9145, Prague, Czech Republic, September 2021. IEEE. ISBN 978-1-66541-714-3. doi: 10.1109/IROS51168.2021.9635944. URL <https://ieeexplore.ieee.org/document/9635944/>.

- [53] Gábor Vásárhelyi, Csaba Virágh, Gergő Somorjai, Norbert Tarcai, Tamás Szörényi, Tamás Nepusz, and Tamás Vicsek. Outdoor flocking and formation flight with autonomous aerial robots, July 2014. URL <http://arxiv.org/abs/1402.3588>.
- [54] Nikolaus Hansen. The CMA Evolution Strategy: A Tutorial, April 2016. URL <http://arxiv.org/abs/1604.00772>.
- [55] Alessandro Attanasi, Andrea Cavagna, Lorenzo Del Castello, Irene Giardina, Tomas S. Grigera, Asja Jelić, Stefania Melillo, Leonardo Parisi, Oliver Pohl, Edward Shen, and Massimiliano Viale. Information transfer and behavioural inertia in starling flocks. *Nature physics*, 10(9):615–698, September 2014. ISSN 1745-2473. doi: 10.1038/nphys3035. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4173114/>.
- [56] Ugo Lopez, Jacques Gautrais, Iain D. Couzin, and Guy Theraulaz. From behavioural analyses to models of collective motion in fish schools. *Interface Focus*, 2(6):693–707, December 2012. doi: 10.1098/rsfs.2012.0033. URL <https://royalsocietypublishing.org/doi/10.1098/rsfs.2012.0033>.
- [57] Eric Klavins. Communication Complexity of Multi-robot Systems. In Jean-Daniel Boissonnat, Joel Burdick, Ken Goldberg, and Seth Hutchinson, editors, *Algorithmic Foundations of Robotics V*, volume 7, pages 275–291. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-642-07341-0 978-3-540-45058-0. doi: 10.1007/978-3-540-45058-0_17. URL http://link.springer.com/10.1007/978-3-540-45058-0_17. Series Title: Springer Tracts in Advanced Robotics.
- [58] Rémy Grünblatt. From WiFi Performance Evaluation to Controlled Mobility in Drone Networks. page 125.
- [59] Bradley Fraser, Robert Hunjet, and Claudia Szabo. Simulating the effect of degraded wireless communications on emergent behavior. In *2017 Winter Simulation Conference (WSC)*, pages 4081–4092, Las Vegas, NV, December 2017. IEEE. ISBN 978-1-5386-3428-8. doi: 10.1109/WSC.2017.8248117. URL <http://ieeexplore.ieee.org/document/8248117/>.
- [60] Rémy Grünblatt, Isabelle Guérin Lassous, and Olivier Simonin. A distributed antenna orientation solution for optimizing communications in a fleet of UAVs. *Computer Communications*, 181:102–115, January 2022. ISSN 01403664. doi: 10.1016/j.comcom.2021.09.020. URL <https://linkinghub.elsevier.com/retrieve/pii/S0140366421003571>.
- [61] Heng Li, Jun Peng, Weirong Liu, Jing Wang, Jiangang Liu, and Zhiwu Huang. Flocking control for multi-agent systems with communication optimization. In *2013 American Control Conference*, pages 2056–2061, Washington, DC, June 2013. IEEE. ISBN 978-1-4799-0178-4 978-1-4799-0177-7 978-1-4799-0175-3. doi: 10.1109/ACC.2013.6580138. URL <http://ieeexplore.ieee.org/document/6580138/>.

- [62] Pierre Laclau, Vladislav Tempez, Franck Ruffier, Enrico Natalizio, and Jean-Baptiste Mouret. Signal-Based Self-Organization of a Chain of UAVs for Subterranean Exploration. *Frontiers in Robotics and AI*, 8:614206, April 2021. ISSN 2296-9144. doi: 10.3389/frobt.2021.614206. URL <https://www.frontiersin.org/articles/10.3389/frobt.2021.614206/full>.
- [63] Overview — The MORSE Simulator Documentation, . URL <https://www.openrobots.org/morse/doc/stable/morse.html>.
- [64] Carlo Pinciroli, Vito Trianni, Rehan O’Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Mauro Birattari, Luca Maria Gambardella, and Marco Dorigo. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295, December 2012. ISSN 1935-3812, 1935-3820. doi: 10.1007/s11721-012-0072-5. URL <http://link.springer.com/10.1007/s11721-012-0072-5>.
- [65] Okumura Y. Field strength and its variability in VHF and UHF land-mobile radio service. *Rev. Electr. Commun. Lab.*, 16:825–873, 1968. URL <https://cir.nii.ac.jp/crid/1572824499680561152>.
- [66] George F. Riley and Thomas R. Henderson. The ns-3 Network Simulator. In Klaus Wehrle, Mesut Güneş, and James Gross, editors, *Modeling and Tools for Network Simulation*, pages 15–34. Springer, Berlin, Heidelberg, 2010. ISBN 978-3-642-12331-3. doi: 10.1007/978-3-642-12331-3_2. URL https://doi.org/10.1007/978-3-642-12331-3_2.
- [67] András Varga and Rudolf Hornig. AN OVERVIEW OF THE OMNeT++ SIMULATION ENVIRONMENT. In *Proceedings of the First International ICST Conference on Simulation Tools and Techniques for Communications Networks and Systems*, Marseille, France, 2008. ICST. ISBN 978-963-9799-23-3. doi: 10.4108/ICST.SIMUTOOLS2008.3027. URL <http://eudl.eu/doi/10.4108/ICST.SIMUTOOLS2008.3027>.
- [68] T. Rasheed. Wireless Mesh Network Simulation Framework for Omnet++ Wireless Mesh Network Wireless Mesh Network Wireless Mesh Network Simulation Simulation Simulation Simulation Framework for Omnet++ Framework for Omnet++ Framework for Omnet++ Framework for Omnet++. 2009. URL <https://www.semanticscholar.org/paper/Wireless-Mesh-Network-Simulation-Framework-for-Mesh-Rasheed/f30a4eee07dde977d1efcaf300d55ff285c453eb>.
- [69] Emerson Marconato, Mariana Rodrigues, Rayner Pires, Daniel Pigatto, Luiz Querino Filho, Alex Pinto, and Kalinka Castelo Branco. AVENS – A Novel Flying Ad Hoc Network Simulator with Automatic Code Generation for Unmanned Aircraft System. January 2017. doi: 10.24251/HICSS.2017.760.

- [70] Nicola Roberto Zema, Angelo Trotta, Guillaume Sanahuja, Enrico Natalizio, Marco Di Felice, and Luciano Bononi. CUSCUS: An integrated simulation architecture for distributed networked control systems. In 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), pages 287–292, January 2017. doi: 10.1109/CCNC.2017.7983121.
- [71] Atheer L Salih, M Moghavvemi, Haider A F Mohamed, and Khalaf Sallom Gaeid. Flight PID controller design for a UAV quadrotor.
- [72] Maurício R. Silva, Elitelma S. Souza, Pablo J. Alsina, Deyvid L. Leite, Mateus R. Morais, Diego S. Pereira, Luís B. P. Nascimento, Adelardo A. D. Medeiros, Francisco H. Cunha Junior, Marcelo B. Nogueira, Glauberto L. A. Albuquerque, and João B. D. Dantas. Performance Evaluation of Multi-UAV Network Applied to Scanning Rocket Impact Area. Sensors, 19(22):4895, January 2019. ISSN 1424-8220. doi: 10.3390/s19224895. URL <https://www.mdpi.com/1424-8220/19/22/4895>. Number: 22 Publisher: Multidisciplinary Digital Publishing Institute.
- [73] Zhenhui Yuan, Jie Jin, Lingling Sun, Kwan-Wu Chin, and Gabriel-Miro Muntean. Ultra-Reliable IoT Communications with UAVs: A Swarm Use Case. IEEE Communications Magazine, 56(12):90–96, December 2018. ISSN 1558-1896. doi: 10.1109/MCOM.2018.1800161. Conference Name: IEEE Communications Magazine.
- [74] T.S. Rappaport. Wireless Communications: Principles and Practice, 2nd Edition, volume 2. December 2001. ISBN 978-0-13-042232-3.
- [75] İlker Bekmezci, Ozgur Koray Sahingoz, and Şamil Temel. Flying Ad-Hoc Networks (FANETs): A survey. Ad Hoc Networks, 11(3):1254–1270, May 2013. ISSN 15708705. doi: 10.1016/j.adhoc.2012.12.004. URL <https://linkinghub.elsevier.com/retrieve/pii/S1570870512002193>.
- [76] Sunil Srinivasa and Martin Haenggi. Path Loss Exponent Estimation in a Large Field of Interferers, January 2012. URL <http://arxiv.org/abs/0802.0351>.
- [77] Govind Sati and Sonika Singh. A REVIEW ON OUTDOOR PROPAGATION MODELS IN RADIO COMMUNICATION. 4(2):5.
- [78] Kalyanmoy Deb and Himanshu Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. IEEE Transactions on Evolutionary Computation, 18(4):577–601, August 2014. ISSN 1089-778X, 1089-778X, 1941-0026. doi: 10.1109/TEVC.2013.2281535. URL <http://ieeexplore.ieee.org/document/6600851/>.
- [79] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2): 182–197, April 2002. ISSN 1941-0026. doi: 10.1109/4235.996017. Conference Name: IEEE Transactions on Evolutionary Computation.

- [80] Guillermo Campos Ciro, Frédéric Dugardin, Farouk Yalaoui, and Russell Kelly. A NSGA-II and NSGA-III comparison for solving an open shop scheduling problem with resource constraints. *IFAC-PapersOnLine*, 49(12):1272–1277, 2016. ISSN 24058963. doi: 10.1016/j.ifacol.2016.07.690. URL <https://linkinghub.elsevier.com/retrieve/pii/S2405896316309661>.
- [81] Julian Blank and Kalyanmoy Deb. Pymoo: Multi-Objective Optimization in Python. *IEEE Access*, 8:89497–89509, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2990567.
- [82] R.O. Saber and R.M. Murray. Flocking with obstacle avoidance: cooperation with limited communication in mobile networks. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 2, pages 2022–2028, Maui, Hawaii, USA, 2003. IEEE. ISBN 978-0-7803-7924-4. doi: 10.1109/CDC.2003.1272912. URL <http://ieeexplore.ieee.org/document/1272912/>.
- [83] M. Lindhe, P. Ogren, and K.H. Johansson. Flocking with Obstacle Avoidance: A New Distributed Coordination Algorithm Based on Voronoi Partitions. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1785–1790, Barcelona, Spain, 2005. IEEE. ISBN 978-0-7803-8914-4. doi: 10.1109/ROBOT.2005.1570372. URL <http://ieeexplore.ieee.org/document/1570372/>.
- [84] Luciano da F. Costa. Further Generalizations of the Jaccard Index, November 2021. URL <http://arxiv.org/abs/2110.09619>.
- [85] Boldizsár Balázs, Gábor Vásárhelyi, and Tamás Vicsek. Adaptive leadership overcomes persistence–responsivity trade-off in flocking. *Journal of The Royal Society Interface*, 17(167):20190853, June 2020. ISSN 1742-5689, 1742-5662. doi: 10.1098/rsif.2019.0853. URL <https://royalsocietypublishing.org/doi/10.1098/rsif.2019.0853>.
- [86] Wenwu Yu, Guanrong Chen, and Ming Cao. Distributed leader–follower flocking control for multi-agent dynamical systems with time-varying velocities. *Systems & Control Letters*, 59(9):543–552, September 2010. ISSN 01676911. doi: 10.1016/j.sysconle.2010.06.014. URL <https://linkinghub.elsevier.com/retrieve/pii/S0167691110000782>.



FOLIO ADMINISTRATIF

THÈSE DE L'UNIVERSITÉ DE LYON OPERÉE AU SEIN DE L'INSA LYON

NOM : BONNEFOND

Date de soutenance : 14/06/2023

Prénoms : Alexandre

Titre : Flocking Models Based on Local Communications: From Theory to Simulations

Nature : Doctorat

Numéro d'ordre : 2023ISAL0040

École doctorale : Informatique et Mathématiques, n° 512

Spécialité : Informatique

Résumé :

Les flottes de robots aériens sont désormais utilisées pour de multiples applications telles que la livraison ou encore la surveillance. Cependant, contrôler un grand nombre de drones ou UAVs (unmanned aerial vehicles) demeure un défi important. Dans cette thèse, nous étudions des stratégies de flocking, directement inspirées de la nature et reposant sur un modèle décentralisé où les entités concernées interagissent localement par le biais de communications. En analysant les performances de modèles de flocking existant dans des environnements très contraints (par des obstacles), nous identifions les potentielles limites de ces modèles et nous proposons de les adapter pour les rendre plus robustes. Dans ces travaux, nous considérons les communications sans fil comme étant le seul moyen d'accéder aux informations des voisins, ainsi nous intégrons un modèle de communication réaliste au simulateur de flocking de Viragh et al. [1].

L'analyse des contraintes au sein des flockings nous conduit à développer un nouveau modèle introduisant des interactions asymétriques et capable de faire évoluer les agents (UAVs) dans des environnements très contraints sans générer de collisions, nommé APR (Asymmetric Pressure Regulation).

Notre dernière contribution consiste en un algorithme de type leader-follower distribué permettant à tout agent follower de devenir un leader. Cela renforce considérablement la cohésion de la flotte et donc favorise le succès de missions consistant à traverser des environnements complexes comme des tunnels.

Mots-clés : Flocking, UAV, Simulation, Communication

Laboratoire(s) de recherche : Centre of Innovation in Telecommunications and Integration of Service (CITI)

Directeur de thèse : Olivier SIMONIN, Isabelle GUERIN-LASSOUS

Président du jury : Nicolas BREDECHE

Composition du jury :

Enrico NATALIZIO, Isabelle FANTONI, Martin SASKA, Olivier SIMONIN, Isabelle GUERIN-LASSOUS