

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ MOHAMED KHIDER - BISKRA



N° d'ordre:

Série:

FACULTÉ DES SCIENCES EXACTES ET DES SCIENCES DE LA
NATURE ET DE LA VIE

DÉPARTEMENT D'INFORMATIQUE

THÈSE

présentée pour obtenir le diplôme de

DOCTORAT EN SCIENCES

SPÉCIALITÉ : INFORMATIQUE

**TOWARD ENTERTAINMENT IN SWARM
ROBOTICS: A FOCUS ON ARTISTIC
DYNAMIC PATTERNS TRANSFORMATION**

par

Belkacem KHALDI

Soutenue le 04/11/2018, devant le jury composé de:

| | |
|--|------------|
| Mohamed Chaouki BABAHENINI, Professeur, Université de Biskra | Président |
| Foudil CHERIF, Professeur, Université de Biskra | Rapporteur |
| Salim CHIKHI, Professeur, Université de Constantine 2 | Examineur |
| Abdelouahab MOUSSAOUI, Professeur, Université de Setif 1 | Examineur |
| Kamel Eddine MELKEMI, Professeur, Université de Batna | Examineur |
| Salim BITAM, MCA, Université de Biskra | Examineur |

Abstract

In the last decade, a wide range of successful applications has been established to meet the increasing need for robot swarms in our daily lives. Bringing such technology into entertainment and artistic activities such as painting, music or even dancing, is a new challenge that will increase and intensify research efforts in this area to target a wide range of stakeholders. Inspired by this challenge, we aim to enrich existing researches in the field of robotics for entertainment, by studying how a large number of robots can be used in ceremonies. We believe that this can be achieved by focusing on a typical study of multi-robot systems, namely patterns formation. The latter, which is an important phenomena found in living organisms (e.g., animals and plants) and physical organisms (e.g., sand dunes or galaxies), is a challenge aimed at confronting the problem of organizing a group of robots in global formations or patterns. These formations could be either simple patterns such as circles, lines, uniform distribution within a circle or square, etc., or complex patterns consist of simple patterns.

In this thesis, we are interested in designing and synthesizing controllers for robotics swarm systems to achieve patterns in a self-organized manner. Our approach is taken from the inspiration of nature, especially from the bio-mechanical forces involved in the studies of the inner cells on the one hand, and from the topological metric revealed in studies of bird flocks on the other hand. In order to produce self-organized aggregating patterns with robots swarm in an effective manner, we have devised many experimental ARGoS-based simulations (**A**utonomous **R**obots **G**o **S**warmling simulator) that allow us to study multiple aspects of self-organized collective behaviors. One of the main problems we focus on to study such behaviors using swarms of robots includes models of formation control, models of self-organized aggregating patterns, and fault detection in swarm robots formation control models.

Keywords: Swarm Robotics, Pattern Formation, Self-Organized Aggregating Patterns, Virtual Viscoelastic Model, Exogenous fault detection.

Résumé

Au cours de la dernière décennie, un large éventail d'applications réussies a été établi pour répondre au besoin croissant des robots d'essaims dans notre vie quotidienne. Utiliser une telle technologie dans des activités récréatives et artistiques telles que la peinture, la musique ou même la danse, est un nouveau défi qui va accroître et intensifier les efforts de recherche dans ce domaine afin de cibler un nombre plus large de personnes intéressées. Inspiré par ce défi, nous visons à enrichir la recherche existante dans le domaine de la robotique en essaim, en étudiant comment utiliser un grand nombre de robots dans les cérémonies. Nous croyons que cela peut être réalisé en se concentrant sur une étude typique des systèmes multi-robots, à savoir la formation de patterns. Ces derniers, sont des phénomènes importants trouvés dans les organismes vivants (tels que les animaux et les plantes) et les organismes physiques (dunes ou galaxies), constituent un défi visant à résoudre le problème de l'organisation d'un groupe de robots en formation ou patterns complets. Ces formations peuvent être soit de simples motifs tels que des cercles, des lignes, une distribution uniforme à l'intérieur d'un cercle ou d'un carré, etc., ou des motifs complexes composés de motifs simples.

Dans cette thèse, on s'intéresse à la conception et à la synthèse de contrôleurs pour des systèmes des robots d'essaim afin d'obtenir des formes de manière auto-organisée. Notre approche est inspirée de la nature, en particulier des forces biomécaniques impliquées dans les études des cellules internes d'une part, et de la métrique topologique révélée dans les études des volées d'oiseaux d'autre part. Afin de produire efficacement des formes d'agrégations auto-organisées avec des robots, nous avons développé plusieurs simulations expérimentales basées sur le simulateur AR-GoS (Autonomous Robots **G**o Swarming), qui nous permettent d'étudier plusieurs aspects du comportement collectifs auto-organisés. L'un des principaux problèmes sur lesquels nous nous sommes concentrés pour étudier un tel comportement en utilisant des robots en essaims, inclut des modèles de contrôle de la formation, des modèles de formes d'agrégations auto-organisés et la détection de failles dans des modèles de contrôle de formation.

Mots-clés: Robotique de l'essaim, Formation de motifs, Modèles d'agrégation auto-organisés, Modèle viscoélastique virtuel, Détection de fautes exogènes.

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful.

Alhamdulillah, all praise be to Allah, Lord of the Worlds, and may the peace and blessings be on the most noble of Prophets and Messengers, our Prophet Muhammad, peace be upon him, and on his family and all of his Companions. I offer to Allah, all praises and gratitude for the strengths and His blessing in completing this thesis.

Foremost, I would like to express my sincere gratitude to my advisor Prof. Foudil Cherif for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the members of my dissertation committee: Pr. Mohamed Chaouki BABAHENINI, Pr. Salim CHIKHI, Pr. Abdelwahab MOUSSAOUI, Pr. Kamel Eddine MELKEMI and Dr. Salim BITAM for generously offering their time, support, guidance and good will throughout the preparation and validation of this document. Without their passionate participation and input, the validation survey could not have been successfully conducted.

My sincere thanks also goes to Dr. Fouzi Harrou and Dr. Ying Sun of the Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division at the King Abdullah University of Science and Technology (KAUST), Saoudi Arabia, for their support and guidance that gave to me, and for the effort done in collaboration with our laboratory to enrich the content of my dissertation. I am gratefully indebted to them for their very valuable comments on this thesis.

Last but not least, my acknowledgments also go to everyone in the LESIA laboratory, it was great to share ideas with all of you during the past five years.

Finally, I must express my very profound gratitude to my family: my lovely wife, my parents and to my brothers and sister for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. I am also grateful to my other family members and friends who have supported me along the way.

Thanks for all your encouragement!

Contents

| | |
|--|------------|
| Abstract | iii |
| Résumé | v |
| Acknowledgements | vii |
| 1 Introduction | 1 |
| 1.1 Introduction | 1 |
| 1.2 Motivation & Main Objectives | 3 |
| 1.2.1 Motivation | 3 |
| 1.2.2 Main Objectives | 4 |
| 1.3 Contributions and Related Publications | 4 |
| 1.3.1 Preview of Contributions | 4 |
| 1.3.2 Related Publications | 6 |
| 1.4 Dissertation layout | 7 |
| I Background and Related Works | 11 |
| 2 An Overview of Swarm Robotics | 13 |
| 2.1 Swarm intelligence (SI) - an inspiration of Natural Swarm Systems | 13 |
| 2.1.1 The Genius of Natural Swarm Systems | 13 |
| Bird flocking | 14 |
| Ants' colonies | 14 |
| Swarming of bees | 15 |
| Fish schools | 16 |
| 2.1.2 Swarm Intelligence Systems: Definition and Properties | 16 |
| 2.1.3 Natural Swarm Behavior based Meta-Heuristics Algorithms | 18 |
| 2.2 Swarm Robotics – Swarm Intelligence applied to Multi-robot systems | 18 |
| 2.2.1 Multi-robotics | 18 |
| 2.2.2 Swarm Robotics | 19 |
| 2.2.3 Potential Application of Swarm Robotics | 21 |
| 2.2.4 Swarm Robotics Problems Focus | 22 |
| 2.2.5 Involved projects and simulations | 24 |
| Swarm robotics involved projects | 24 |
| Swarm robotics simulation platforms | 24 |

| | | |
|-----------|---|-----------|
| 2.3 | Summary | 28 |
| 3 | Studies in Aggregating Patterns and Faults Detection within Swarm Robotics systems | 29 |
| 3.1 | Overview | 29 |
| 3.2 | Aggregation Patterns in Nature | 29 |
| 3.2.1 | Aggregation in Nature | 29 |
| 3.2.2 | Emergence of Patterns in Aggregating Natural Swarms | 30 |
| 3.2.3 | Case study: Flocking Patterns | 30 |
| | Models on Flocking Patterns | 31 |
| 3.3 | Aggregation Patterns in Swarm Robotics | 32 |
| 3.3.1 | Aggregation in Swarm Robotics | 32 |
| 3.3.2 | Emergence of Patterns in Aggregating Robotics Swarm | 32 |
| 3.4 | Approaches on Swarm Robotics Aggregation Patterns | 33 |
| 3.4.1 | Cue-Based Methods | 33 |
| 3.4.2 | Self-organized based Methods | 34 |
| | Probabilistic Approach | 34 |
| | Deterministic Approach | 35 |
| | Artificial Evolution Approach | 35 |
| | Morphogenesis Inspired Approach | 35 |
| | Artificial Physics Approach | 36 |
| 3.5 | Detecting Faulty Robots in Swarm Robotics Systems | 37 |
| 3.5.1 | Faults in Engineered Systems | 37 |
| 3.5.2 | Faults in Swarm Robotics Systems | 37 |
| | Fault Types | 38 |
| 3.5.3 | Fault Detection in Engineered Systems | 39 |
| | Legacy Approach | 39 |
| | Model-based Approaches | 40 |
| | Data-driven Approaches | 40 |
| 3.5.4 | Fault Detection in Robotics Swarm Systems | 41 |
| | Endogenous Approaches | 41 |
| | Exogenous Approaches | 42 |
| 3.6 | Summary | 42 |
| II | Self-Organized Patterns in Aggregating robots swarm | 45 |
| 4 | Material and Methods | 47 |
| 4.1 | Overview | 47 |
| 4.2 | Simulation platform (ARGoS) | 47 |
| 4.3 | Robotic Platform | 49 |
| 4.3.1 | Foot-bot | 50 |
| 4.3.2 | e-puck | 50 |

| | | |
|----------|---|-----------|
| 4.4 | On-board Sensing/Actuating System | 51 |
| 4.4.1 | Infrared range and bearing sensors | 52 |
| 4.4.2 | Infrared proximity sensors | 53 |
| 4.4.3 | Wheels Actuator | 54 |
| 4.5 | Methods | 55 |
| 4.5.1 | The Viscoelastic Model: A Bio-mechanic inspired Model | 56 |
| | Overview of the inner cell Bio-mechanics | 56 |
| | The swarm robots viscoelastic interaction model | 56 |
| 4.6 | Summary | 59 |
| 5 | Basic Geometric Formations | 61 |
| 5.1 | Overview | 61 |
| 5.2 | Methods | 61 |
| 5.2.1 | The Task & the Experimental Setup | 61 |
| 5.2.2 | Robot' Controller | 62 |
| | Proximity Control (PC) | 62 |
| | Repulsive Control (RC) | 63 |
| | Forward Dependent Angular Motion Control (FDAMC) | 64 |
| 5.3 | Algorithms & Results | 65 |
| 5.3.1 | Regular Shapes Formation within foot-bots | 65 |
| 5.3.2 | Circle Formation within e-pucks | 68 |
| | Circumscribed Circle Theory | 69 |
| | Circle Formation via the Virtual Viscoelastic Control Model | 69 |
| 5.4 | Performance Analysis | 72 |
| 5.4.1 | Performance Metrics | 72 |
| | Group Speed | 72 |
| | Mean Distance Error | 72 |
| 5.4.2 | Results within the Metrics | 72 |
| 5.5 | Conclusion | 74 |
| 6 | Topological Approaches in Self-organized Aggregating Patterns | 77 |
| 6.1 | Overview | 77 |
| 6.2 | Methods | 77 |
| 6.2.1 | The Task & the Experimental Setup | 77 |
| 6.2.2 | Robot' Controller | 78 |
| | The K-NN Topological Approach | 78 |
| | The DW-KNN Topological Approach | 79 |
| 6.3 | Results & Discussion | 82 |
| 6.3.1 | Self-organized Aggregation Patterns basing on the KNN approach | 82 |
| 6.3.2 | Self-organized Aggregation Patterns basing on the DW-KNN approach | 83 |
| 6.4 | Performance Analysis | 86 |

| | | |
|----------|--|------------|
| 6.4.1 | Performance Metrics | 87 |
| | Distance-Weighted Distribution Quality | 87 |
| | Aggregation Quality | 88 |
| | Dispersion Quality | 88 |
| | Averaged Mean Distance Error | 89 |
| 6.4.2 | Analysis in Normal Circumstances | 89 |
| 6.4.3 | The Effect of Sensory Noise on the Performnce of the DW-KNN Approch | 90 |
| 6.5 | Analysis of the Two Approaches within Different Noise Models | 93 |
| 6.5.1 | Effect of Uniform Noise on the Performance Proposed Ap- proach | 93 |
| 6.5.2 | Effect of Gaussian Noise on the Performance Proposed Approach | 94 |
| 6.5.3 | Effect of Mean Shift Noise on the Performance Proposed Ap- proach | 95 |
| 6.5.4 | Effect of Autocorrelated Noise on the performance proposed approach | 95 |
| 6.6 | Conclusions | 97 |
| 7 | Detecting Faulty Robots in Aggregating robots Swarms | 99 |
| 7.1 | Overview | 99 |
| 7.2 | the task and the main objective | 99 |
| 7.3 | PCA-based monitoring approaches | 100 |
| 7.3.1 | Feature extraction using PCA | 100 |
| 7.3.2 | PCA-based fault detection | 101 |
| 7.4 | Univariate statistical control charts | 102 |
| 7.4.1 | EWMA monitoring charts | 102 |
| 7.4.2 | Cumulative sum (CUSUM) charts | 103 |
| 7.4.3 | Combining PCA with CUSUM and EWMA charts | 104 |
| 7.5 | Results and discussion | 104 |
| 7.5.1 | PCA modeling | 105 |
| 7.5.2 | Detection results | 107 |
| | Case (A): Abrupt fault detection | 108 |
| | Case (B): Intermittent fault | 109 |
| | Case (C): Random walk fault | 110 |
| | Case (D): Complete stop fault | 110 |
| | Case (E): Drift failure detection | 112 |
| 7.6 | Conclusion | 113 |
| 8 | Conclusion and Future Works | 115 |
| 8.1 | Future Works | 117 |

| | |
|---|------------|
| A Algorithms and Listings | 119 |
| A.1 An Example of an XML based ARGoS Configuration File | 119 |
| A.2 The Overall KNN Control Algorithm Implemented in a Foot-Bot Robot | 122 |
| A.3 The Overall DW-KNN Control Algorithm Implemented in a Foot-Bot Robot | 125 |
| Bibliography | 127 |

List of Figures

| | | |
|------|--|----|
| 2.1 | A B C D | 15 |
| 3.1 | The three basic steering behaviors determining the motion of the boids. (A) Separation: reacting only to nearby flockmates (B) Alignment: steering towards the average heading direction of the local flock-mates. (C) Cohesion: moving toward the average position of the neighboring boids. (From http://www.red3d.com/cwr/boids/). | 31 |
| 3.2 | General time-independent fault forms Isermann [131]. (a) abrupt: fault arising suddenly in time, (b) incipient: fault developing slowly dur- ing the operation of a system, and (c) intermittent: fault occurring in discrete intervals. | 38 |
| 3.3 | Schematic description of a model-based fault detection approach [138]. | 40 |
| 3.4 | Schematic description of a data-driven based fault detection approach [138]. | 41 |
| 4.1 | Schematic description of the ARGoS architecture [153]. | 48 |
| 4.2 | A snapshot during a run-time based ARGoS simulation for Listing A.1. A foot-bot robot is placed in an area of $9m^2$ of surface surrounded by 4 walls. | 49 |
| 4.3 | (A) The real foot-bot robot. (B) a CAD model for the foot-bot robot. . . | 51 |
| 4.4 | (A) The real e-puck robot. (B) a CAD model for the e-puck robot. . . . | 51 |
| 4.5 | The relative position measurements within the communication system. | 52 |
| 4.6 | Position of IR proximity Sensors in (A) the foot-bot robot and (B)the e-puck robot. | 53 |
| 4.7 | A default ARGoS based configuration of IR proximity sensors. | 54 |
| 4.8 | Kinematics of a two wheel differential drive mobile robot. | 54 |
| 4.9 | A default ARGoS configuration of the wheels actuator device. | 55 |
| 4.10 | The sub-cellular Viscoelastic model of Jamali, Azimi, and Mofrad [162]. | 57 |
| 4.11 | The swarm robots Viscoelastic interaction model. | 57 |
| 4.12 | Schematic illustration of the studies to be carried out using the Vis- coelastic model. | 58 |
| 4.13 | Schematic illustration of the self-organized aggregation model. | 58 |
| 5.1 | The Overall Control Model | 62 |
| 5.2 | A model of the <i>viscoelastic</i> connection setup between two robots. | 63 |

| | | |
|-----|---|----|
| 5.3 | N foot-bots achieving different final basic geometric formations (b) when starting from initial positions (a) using our viscoelastic based model. | 67 |
| 5.4 | Rate of change of (A) v_i and (B) \hat{a}_i with respect to k_w during the last 200 time steps of a pentagon formation. | 68 |
| 5.5 | Rate of change of (A) $\hat{a}_i(t)$ and (B) $v_i(t)$ for 5 foot-bots ($rb0, rb1, rb2, rb3$ and $rb4$) performing a pentagon formation task. the x-axis of the plots represents the evolution of time step t | 68 |
| 5.6 | A circumscribed circle of a hexagon. | 69 |
| 5.7 | Four simulations scenarios of a swarm of N e-pucks forming a uniform circle via the virtual viscoelastic based control model. | 71 |
| 5.8 | Evolution in time of (A) Group Speed (GS) and (B) Mean Distance Error ($MDER$) during a formation of four different regular shapes by a N foot-bot robots under absence and presence of noise in the RAB device. The curves in the plots represents the median values obtained from 5 runs by each experiment. | 73 |
| 5.9 | Evolution in time of (A) Group Speed (GS) and (B) Mean Distance Error ($MDER$) during a circle formation by N e-puck robots under absence and presence of noise in the RAB device. The curves in the plots represents the median values obtained from 5 runs by each experiment. | 74 |
| 6.1 | Schematic description of the KNN control model. | 79 |
| 6.2 | Schematic description of the DW-KNN control model. | 80 |
| 6.3 | A schematic representation of SPH particle interactions within the influence domain governed by the kernel function. | 81 |
| 6.4 | Self-organized Aggregation Patterns with the KNN Topological Approach. | 84 |
| 6.5 | Self-organized aggregation of a swarm of N foot-bots running the DW-KNN topology | 86 |
| 6.6 | Self-organized aggregation of a swarm of N foot-bots running the DW-KNN topology with existing obstacles | 87 |
| 6.7 | Performance metrics results in absence of obstacles: (A) $F_{ag}(t)$ and (B) $AMDE(t)$ obtained from the overall controller implemented on $N = \{100, 150\}$ robots when taking into consideration different KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 25 runs with different initial configurations of robots. | 90 |

- 6.8 Performance metrics results in absence of obstacles: (A) $F_w(t)$, (B) $F_{ag}(t)$ and (C) $AMDE(t)$ obtained from the overall controller implemented on $N = \{50, 100, 150\}$ robots when taking into consideration different DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 25 runs with different initial configurations of robots. 91
- 6.9 Performance metrics results in presence of obstacles: (A) $F_w(t)$, (B) $F_{ag}(t)$ and (C) $AMDE(t)$ obtained from the overall controller implemented on $N = \{50, 100, 150\}$ robots when taking into consideration different DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 25 runs with different initial configurations of robots. 91
- 6.10 Performance metrics results: (a) $F_w(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on 100 robots when taking into consideration different DW-KNN topologies with different σ . In all plots, the x-axis represents the standard deviation of noise σ . Each box represents values obtained from 25 runs with different initial configurations of robots. The red squares show the median values, and the dashed red lines shows a linear least squares regression fit to these squares; the olive band shows a linear least squares regression fit to the mean of the 5 red squares for each topology. The linear least squares regression functions generated to fit all of these points are of the form $f(x) = c_1x + c_0$; the corresponding coefficients can be found in the tables to the right of each figure. 92
- 6.11 Performance metrics results for the KNN and the DW-KNN topological Aggregation Approaches with the uniform noise model: (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 5 runs with different initial configurations of robots. 94
- 6.12 Performance metrics results for the KNN and the DW-KNN topological Aggregation Approaches with the Gaussian noise model: (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 5 runs with different initial configurations of robots. 95

| | | |
|------|---|-----|
| 6.13 | Performance metrics results for the KNN and the DW-KNN topological Aggregation Approaches with the Mean Shift noise model: (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 5 runs with different initial configurations of robots. | 96 |
| 6.14 | Performance metrics results for the KNN and the DW-KNN topological Aggregation Approaches with the Auto-Correlated noise model where $a = 0.5$: (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 5 runs with different initial configurations of robots. | 97 |
| 7.1 | A flowchart of a PCA-based fault detection schemes. | 106 |
| 7.2 | Evolution in time of (a) $AMDE$, and (b) GS | 106 |
| 7.3 | Snapshots during ARGoS simulation of 6 foot-bots performing the VVC model at: (a) $t=0$, (b) $t=250$ and (c) $t=500$ | 107 |
| 7.4 | Three PCs capture 96% of information in the system. | 107 |
| 7.5 | Monitoring results of PCA- T^2 (a), PCA- Q (b), and PCA-EWMA charts(c) for the normal operation data. | 108 |
| 7.6 | Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $h = 0.19$) (c) charts in the presence of an abrupt fault in x_1 from sample 150 to 200 (Case (A ₁)). | 108 |
| 7.7 | Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $h = 0.19$) (c) charts in the presence of an abrupt fault in x_1 from sample 150 to 200 (Case (A ₂)). | 109 |
| 7.8 | Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $h = 0.19$) (c) charts in the presence of intermittent faults in x_1 between sample times [50 100] and [150 200] (Case (B)). | 110 |
| 7.9 | Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $H = 0.19$) (c) charts when the first robot performs a random walk from sample number 200 through the end of the testing data, Case (C). | 111 |
| 7.10 | Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $h = 0.19$) (c) charts when the first robot has completely stopped working between sample times 200-300, Case (D). | 111 |

| | |
|---|-----|
| 7.11 Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $h = 0.19$) (c) charts in the presence of a drift fault with slope 0.01 in x_1 from sample 150, Case (E). | 112 |
|---|-----|

List of Tables

| | | |
|-----|--|-----|
| 2.1 | Comparison of swarm robotics systems and multi-Robotics Systems. | 21 |
| 2.2 | Problems Classification in Swarm robotics. | 23 |
| 2.3 | Some of successful Swarm Robotics Projects. | 25 |
| 5.1 | Parameters and Constants relative to the regular shape formation model for the foot-bot robot. | 66 |
| 5.2 | Parameters and constants relative to the circle formation model for the e-puck robot | 70 |
| 6.1 | Parameters and constants used in the KNN approach | 83 |
| 7.1 | PCA-based EWMA and CUSUM fault detection procedures. | 105 |
| 7.2 | Data collected from the ARGoS simulation | 105 |
| 7.3 | False and miss detection rates for all monitoring charts. | 112 |

Chapter 1

Introduction

1.1 Introduction

Since a long time dating back to the middle of the twentieth century, robots have been effectively involved in many industrial sectors. The early use of industrial robots was ultimately aimed at automating most of the purely repetitive and mechanical tasks often performed by humans. This could be beneficial in a number of ways, from offering better degrees of consistency and much higher production speeds, to achieving significant degrees of accuracy. Today, most robots used in such industrial tasks are designed to perform well-defined operations at a given time and in a fixed location without any flexibility in their morphological re-configurability. As a result, this category of robots is only used in known environments and is limited to performing relatively simple tasks.

On the other hand, there are other tasks that should be performed in unstructured environments, which could change over time and require some degree of autonomy and flexibility that is not found in industrial robots. Examples of such tasks are the exploration and coverage of large areas, as well as environmental monitoring and natural disaster monitoring. To cope with such tasks and overcome the limitations of industrial robots, a new generation of robotic systems has been designed based on the theory of artificial intelligence introduced since the 1960s. In the first generations, a new field of robotics called mobile robotics [1] was used to investigate studies with robots able to detect and move around their environment. Simultaneous Localization and Mapping (SLAM) is one of the typical problems studied in mobile robotics. Other studies with this type of robots also include navigating through a field of obstacles by the use of path planning.

In the late of 1980s, multi-robot systems (MRS) were specifically introduced to address the lack of information processing capacity and many other aspects of single mobile robots that cannot perform special tasks like those needing cooperation and collaboration between groups of robots [2]. Since nature provides fascinating examples of team collaboration in performing collective behaviours, more research has begun to draw direct inspiration from biological studies giving rise to a new bio-inspired robotics discipline termed bio-inspired robotics [3]. Indeed, in this discipline several aspects of researches can be associated, most of each can be considered as a

sub-field in its own right. For example, few researches follow developmental robotics [4] to design cognitive robotics, which can be achieved by mimicking the way humans acquire cognitive abilities in learning. Others follow soft robotics [5] to cope with robots that are very flexible and adaptable to operate in environments where rigid robots fail. This type of robots is mainly inspired by how living organisms such as octopus thighs, elephant trunks and snake bodies move and adapt to their surroundings. Another class of bio-inspired robots is that of Legged robots [6], these robots belong to the category of mobile robots in which the locomotion is obtained using mechanical limbs. They often use legged animals, such as humans or insects, as sources of inspiration, making them more versatile and more suitable for traversing many complex terrains than wheeled robots.

The types of multi-robot systems described above, and which depict a group of simple physical robots collaborating on specific tasks, have also been very successful and have made great steps in many areas, such as cooperative transportation and aggregation, environmental monitoring, search and rescue missions, foraging and space exploration [7]. However, with emerging new challenges such as decentralization in control and self-organization as well as some tasks require large groups of autonomous robots to be efficiently executed, a significant interest in applying swarm intelligence techniques in multi-robotic systems has given rise to a new area of research called *swarm robotics* [8, 9].

The main idea behind swarm robotics is the study of how to coordinate large groups of relatively simple robots through the use of local rules. It focuses on studying the design of a large number of relatively simple robots, their physical bodies and their control behaviors to achieve a specific task that is beyond the capability of a single robot [10]. Swarm robotics is closely related to the idea of swarm intelligence and shares its interest in self-organizing decentralized systems. It offers several advantages for robotic applications such as scalability, flexibility and robustness due to redundancy [11].

In its early age, swarm robotic was involved in mimicking intelligent swarming behaviors of social animals such as foraging [12], aggregation [13], flocking [14], cooperative transport of objects [15], and self-organized patterns formation [16]. Recently, with the tremendous progress being made in this area, researchers are focusing primarily on how a swarm robot system can be involved in our real life. Today, they can be effectively involved in military (e.g., collective bomb detection, cooperative research and exploration), logistics (e.g., managing warehouses and products delivery to customers), agriculture (e.g., seeding, harvesting and grains storage) and emergency (e.g., rescuing in disasters), etc.

1.2 Motivation & Main Objectives

1.2.1 Motivation

Entertainment and art are among the applications domains that have lately attracted more attention in the swarm robotics literature. Bringing such technology to Artistic activities (such painting, music or even dancing...) is a new challenge that will increase and multiply efforts to target a large range of interesting public.

Opening or closing ceremonies, which have recently become international manifestations in every sport events, are till now an interesting artistic spectacles that attract thousands of viewers on live and millions of witnesses on TVs due to the high entertainments given to them. With the high increases of technology used in such events, more sophisticated artistic designs have been applied to show competition even between one ceremony to another. For example at the opening and closing ceremonies of the Beijing Olympic Games 2008, the one of London 2012, and the last one held in Brazil 2016, very interesting artistic patterns and designs have shown using a high lighting technology.

We think that these artistic designs and patterns, that have been generated using, in general cases a great number of human beings, can be regenerated using a large scale of robots swarm. We believe that this can be achieved by focusing on studying a typical task of swarm robotics systems, which is pattern formation. This last, which is an important phenomenon that exists in biological organisms (e.g., animals, plants) and physical entities (e.g., sand dunes or galaxies), is a challenge that aims to face a fundamental problem in which a group of swarm robots are aggregated into global formations or patterns that can be either simple patterns like circles, lines, uniform distribution within a circle or square, etc or either complex patterns composed of simple patterns.

The problem of pattern formations in swarm robotics is a challenging research that includes: (1) identifying the robots that will form the pattern, (2) maintaining the generated pattern as the robots move, (3) avoiding obstacles and collisions during movement, (4) treating pattern transformation or reconfiguration, and (5) coordinating multiple patterns while multiple groups of robots are forming independent patterns.

The task of pattern formation in swarm robotics has been widely studied basing on several approaches from Leader/Follower algorithms to Potential field algorithms to recently Biologic-inspired algorithms. However, we believe that there still exist a gape to fill, specially when we make attention to the amazing self-organized aggregating patterns observed in flock of birds and school of fish, and try to abstract a physical based model from the process behind them.

The research presented in this dissertation is motivated by the purpose of designing methods for self-organizing a large number of very simple weak mobile robots into aggregating patterns that might be later applied in the entertainment application domains. The work presented here summarizes some significant steps in the context

of self-organized aggregating patterns, which studies the cohesive and coordinated movement of a group of robots.

1.2.2 Main Objectives

In light of the above motivations, the general purpose of this dissertation is to advance state-of-the-art studies in aggregating patterns behaviors within robotics swarms.

The specific objectives of this thesis are summarized in the following notes:

- Conducting, in general terms, a literature review of the current studies in swarm intelligence and swarm robotics, and in a specific manner a detailed background about the approaches used in studying swarm robotics aggregating patterns, as well as the different methods used in detecting faults in such robotics systems.
- Studying generally the biologic process behind the mechanics involved in cell morphology, and particularly the bio-mechanics properties (i.e, elasticity and viscosity) of the cell cytoskelton that are involved in cell shaping and in the overall organization of cell's parts.
- Investigating the topological metric approach revealed in aggregation studies of birds flocking and fish schooling, through which we take inspiration to study aggregating patterns within swarm robotics systems.
- Developing a bio-mechanical inspired model that shall constitute the structure of the solutions, as well as a methodology for obtaining these solutions to study different aggregating patterns within robotics swarm systems.
- In light of the bio-mechanical inspired model and the revealed topological metric approach, designing different controllers that shall be implemented in a robotics swarm platform to investigate solutions to the aggregating patterns task.
- Showing that the proposed bio-mechanical inspired model is so flexible in a manner that can be successfully re-implemented in other robotics platform.
- Extending the model by proposing a faults detection method to cope with faults in robots that may lead in degradation of the overall performance of the entire swarm system.

1.3 Contributions and Related Publications

1.3.1 Preview of Contributions

The contributions shown in this dissertation can be summarized as follow:

- We provide a review study [17] in which a detailed overview about swarm robotics has been presented. The study aims to orientate readers, mainly those

newly coming to this field of research, in a manner that it gives a clarified picture about the swarm robotics research field by comparing it with the multi-robotics domain research. The study also highlights the major axis and the important projects, robotics platforms as well as the different simulators that have been addressed so far to investigate swarming behaviors using swarm robotic systems. New Interests coming to this topic of research can be easily guided through the different sections presented in the study. Up to date, the study has been cited nine (09) times in different journal and conference papers, including three (03) auto-citations.

- An artificial viscoelastic model based on the bio-mechanical properties of the cell-cytoskeleton has been proposed to study basic geometric pattern formations using a group of simple limited differential drive mobile robots. The virtual viscoelastic model has been implemented in simulated versions of foot-bot robots to successfully achieve different geometric configurations such as triangles, squares and pentagons. Additionally, the proposed bio-inspired model has been adapted to study circle formation using another differential driver mobile robots called e-puck robots [18]. This to show that the model is highly flexible to be adapted and implemented in any differential drive mobile robotics platform.
- In light of the topological metric approach revealed in studies of birds flocking patterns, two topological neighborhood approaches have been proposed to investigate swarm robotics aggregating patterns within the viscoelastic model. In the first topological approach, we suggest using a K-Nearest Neighboring (KNN) topological method to aggregate the robots [19]. Here, the artificial viscoelastic interactions between the robots are governed by the nearest K neighbors, meaning that each robot is interacting only with its K-closest teammates basing only on distances toward neighbors as the only factor in the aggregation process. In the other topological approach, we propose using a distance-weighted KNN (DW-KNN) method for the aggregation of the robots [20, 21]. With the DW-KNN topological approach, the distances towards the neighbors are weighted using a density estimation technique. Therefore, each robot selects its neighbors basing on both distances and densities of its teammates as two keys factors in the aggregation process. With the two proposed topological approaches, various aggregation patterns in presence and absence of obstacles have been achieved within tens of simulated versions of foot-bots robots.
- An analyze study under different aggregation metrics of performance, and in presence of noises in the robots sensors, has been further investigated to show the effect of different noise models in the performance of our virtual viscoelastic model [20, 22]. The analyze has been addressed within the two proposed topological approaches, and it shows that the DW-KNN approach performs better than KNN one.

- Additionally, a fault detection method has been suggested to enhance our model in order to cope with faults that may lead in a degradation in the performance of our artificial viscoelastic model [23, 24]. The method merges the flexibility of principle component analyses (PCA) models and the greater sensitivity of the exponentially-weighted average (EWMA) and cumulative sum (CUSUM) control charts to insidious changes. To this end, the method is tested and evaluated on a swarm of simulated foot-bot robots performing a circle formation task, via the viscoelastic control model.

1.3.2 Related Publications

To this end, the above preview of contributions has been so far reported in the following publications in academic journals and conferences:

1. Khaldi, B., & Cherif, F. (2015). **An Overview of Swarm Robotics: Swarm Intelligence Applied to Multi-robotics.** *International Journal of Computer Applications*, 126(2), pp. 31-37.
2. Khaldi, B., & Cherif, F. (2016, June). **A Virtual Viscoelastic Based Aggregation Model for Self-organization of Swarm Robots System.** *In Conference Towards Autonomous Robotic Systems (TAROS)*, Shiefield, UK. (pp. 202-213). Springer International Publishing.
3. Khaldi, B., & Cherif, F. (2016, November). **Swarm robots circle formation via a virtual viscoelastic control model.** *In 2016 8th International Conference on Modelling, Identification and Control (ICMIC)*, Media, Algeria. (pp. 725-730). IEEE.
4. Khaldi, B., Harrou, F., Sun, Y., & Cherif, F. (2017, May). **A measurement-based fault detection approach applied to monitor robots swarm.** *In 2017 6th International Conference on Systems and Control (ICSC)*, Batna, Algeria. (pp. 21-26). IEEE.
5. Khaldi, B., Harrou, F., Cherif, F., & Sun, Y. (2017). **Monitoring a robot swarm using a data-driven fault detection approach.** *Journal of Robotics and Autonomous Systems*, (97), pp. 193-203.
6. Khaldi, B., Harrou, F., Cherif, F., & Sun, Y. (2017, October). **A Distance Weighted-based Approach for Self-Organized Aggregation in Robot Swarms.** *In 2017 5th International Conference on Electrical Engineering (ICEE)*, Boumerdes, Algeria. (pp.1-6). IEEE.
7. Khaldi, B., Harrou, F., Cherif, F., & Sun, Y. (2017, November). **An Efficient Aggregation Topological Strategy for Self-Organized Patterns within Robots Swarm.** *In 2017 3rd International Conference on Electrical Engineering and Control Applications (ICEECA)*, , Constantine, Algeria. (pp. xx-xx). IEEE.

8. Khaldi, B., Harrou, F., Cherif, F., & Sun, Y. (2018). **Self-organization in Aggregating Robot Swarms: A DW-KNN Topological approach.** *Journal of BioSystems*, 165, pp. 106-121.

Publications 2, 4, 6 and 7 were all presented orally as full papers by the author himself at the respective conferences, held in Sheffield, UK, Batna, Boumerdes, and Constantine, Algeria.

Publication 3 was also presented orally as a poster by the author himself at the 8th International Conference on Modeling, Identification and Control (ICMIC), which took place in Media, Algeria.

The material in Publication 1 highly corresponds to the contents of Chapters 2 of this thesis. The preliminaries and related work material from all the publications has contributed so far to the content of Chapter 3.

The methods and the analyses sections presented in Publications 2 and 3 have loosely contributed to the content of Chapter 4 and Chapter 5.

The content of Chapter 6 is highly depends on the material, the methods, and the performance analyses sections discussed in Publications 6, 7, and 8.

Finally, all the material contents and most of the analyses studies that were depicted in Publications 4 and 5 have been a source of contribution for the contents of Chapter 7.

1.4 Dissertation layout

This thesis is organized into 7 chapters grouped into two main parts: (I) Background and Related Works and (II) Self-Organized Patterns in Aggregating robots swarm. The first part is composed of two chapters (Chapter 2 and Chapter 3), it puts the reader in the context of the thesis and gradually introduces to him the state of art and the related studies of the research presented in this dissertation.

More specifically, in Chapter 2, a detailed overview about the swarm robotics field is provided in a chronological manner. The chapter first starts by presenting the field of swarm intelligence and its main fundamental concepts, with highlighting some of the most natural swarming behaviors that led to the birth of swarm intelligence. It then introduces the swarm robotics domain as a specific multi-robotics sub-domain where the theory of swarm intelligence is applied, and with giving the main differences between the two research fields. Finally, it presents some of the potential application of swarm robotics and the main problems that are being addressed in this field, as well as mentioning some of the successful projects and the most simulation platforms known in the literature.

Chapter 3 digs into the main issues addressed in this thesis, and provides relevant studies on aggregating patterns and fault detection in robotic swarm systems. More specifically, the first part of this chapter presents studies on aggregating patterns in natural swarms, and then reviews how these studies have been used to address the

problem of pattern formation using robotic swarm systems. The chapter discusses mainly two kinds of studies in aggregating patterns: (1) a cue-based and (2) a self-organized based methods, with more focus on the second method. Therefore, the chapter emphasizes in more details the different self-organized based aggregation approaches that have so far used in the literature. The second part of Chapter 3 addresses the fault detection problematic within swarm robotic systems. The content of this part starts by providing a general overview about faults in engineered systems as well as the different approaches used in detecting faults in such systems, and then it projects that in the field of swarm robotics.

Second part of this thesis considers and provides solutions to the self-organized aggregating patterns issues addressed in this dissertation. It is composed essentially of four chapters (Chapter 4-7). Chapter 4 presents the materials and the methods that are used to synthesis controllers for aggregating patterns studies within robotics swarm. It specifically introduces the simulator and its architecture as well as the robotics platform adopted in our studies. Moreover, it discusses all the required materials in terms of the robot on-board sensing and actuating systems that are particularly used to implement the proposed aggregating patterns based controllers. Additionally, it introduces our swarm robotics viscoelastic interaction model as a bio-mechanics based model which takes inspiration from the bio-mechanics involved in inner-cells. Finally, it summaries the main swarming behavior studies investigated in this document.

In Chapter 5, we provide a simple solution to the formation control task for a swarm robotics system. The solution makes use of our virtual viscoelastic interaction model to achieve basic geometric formations such as triangles, squares, pentagons, or circles. The chapter first provides a description about the task to be performed by the robots, and the experimental setup settled for the task. It then presents and explains in details the overall robot controller used to perform the task. With this robot controller, two experimental studies using two robotics platforms have been investigated in this chapter. In the first study, we achieve regular geometric formations using foot-bots robots. We illustrate basically the implementation of the proposed overall controller in a foot-bot robot, and we assess the performance of the solution under different metrics of performance. In the second study, we adapt the control model to address a circle formation task using e-puck robots. This study shows the flexibility of our proposed overall control model to be adapted and implemented in any differential drive mobile robotics platform.

Chapter 6 extends the control model introduced in the previous chapter to the task of self-organizing aggregating patterns within robots swarm. In the beginning of this chapter, a description of the task to be performed as well as the experimental setup settled for the task is provided to the reader. Then basing on the topological metric studies revealed in birds flocking behavior, two extensions to the previous basic control model are discussed to investigate self-organized aggregating patterns. In the first extension, the idea of implicating our viscoelastic interaction model in a

KNN topological aggregation approach is proposed to study its impact in emerging self-organized aggregating patterns. Whereas in the second extension, our viscoelastic interaction model is involved in another topological aggregation approach called DW-KNN. Within the two topological aggregation approaches, a robot controller, implemented in a foot-bot robot, is explained in details with providing analyses studies under different metrics of performance. The chapter further provides analyses of the two topological approaches in presence and absence of different models of noises in the robot sensors, and also provides some performance results in presence and absence of obstacles.

Chapter 7 addresses the problem of fault detection in robots swarm. It provides an innovative exogenous fault detection method for monitoring robots swarm as a solution to such a problem. The chapter mainly reviews the PCA based monitoring approach by describing feature extraction using PCA, and highlighting the use of two conventional monitoring statistics, the T^2 and Q statistics, as keys factors in the PCA-based fault detection methods. The chapter also provides a detail explanation about univariate statistical control charts, such as CUSUM and EWMA, which have been widely used to monitor industrial processes. Later, solutions to the fault detection problem are proposed by combining PCA with CUSUM and EWMA charts. The solutions are evaluated and tested using a swarm of foot-bots performing a circle formation task, via our virtual viscoelastic control model. The implementation of the developed monitoring methods is described in details, and results within the proposed solutions show the performance of the fault detection techniques in detecting different kind of faults including abrupt, intermittent, random walk, complete stop, and gradual faults. Results also demonstrate that a significant improvement in fault detection can be obtained by using the proposed methods where compared to the conventional PCA-based methods (T^2 and Q).

Finally, Chapter 8 concludes the thesis and debates a number of potential perspectives and directions for future works.

Part I

Background and Related Works

Chapter 2

An Overview of Swarm Robotics

As an emergent research area by which swarm intelligence theory is applied to multi-robotics systems, swarm robotics is a very particular and peculiar sub-area of collective robotics that studies how to coordinate large groups of relatively simple robots through the use of local rules. It focuses on studying the design of large amount of relatively simple robots, their physical bodies and their controlling behaviors. Since its introduction in 2000, several successful experiments had been realized, and till now more projects are under investigations. For the aim to orientate the readers, mainly those who are newly coming to this research field, this chapter seeks to give an overview of this domain of research with highlighting the grand lines of its different main focuses areas that are under investigated.

2.1 Swarm intelligence (SI) - an inspiration of Natural Swarm Systems

Who among us haven't been amazed by the individually simple but collectively complex behavior exhibited by natural grouping systems including social insects such as: ant' colonies, termites, bees, wasps ... etc, and high order living animals such as: flocks of birds, fish schooling, and packs of wolves ...? Over time, scientists have trying, using a very interesting principle said by Albert Einstein: *"Things should be made as simple as possible, but not any simpler"* [25], to understand the underlying principles behind these amazing natural collective complex behaviors that are emerged from individual simple local interactions rules. The robustness, scalability, and distributed self-organization principles observed in these natural systems, have been deeply studied by scientists and their attempt to apply the insight gained through this research to artificial systems (e.g., massively distributed computer systems and robotics) has given rise to a new research topic called Swarm Intelligence (SI) [25].

2.1.1 The Genius of Natural Swarm Systems

In nature, collective complex behaviors exhibited by grouping of insects or animals are generally associated to the term of *"swarm"* [26]. This last and refereeing to Hinchey, Sterritt, and Rouff [27], is defined as: *"images of large groups of small insects*

in which each member performs a simple role, but the action produces complex behavior as a whole". This means that the observed complex or macroscopic behavior of the whole swarm system is produced from the combination of the simple (microscopic) behaviors of the numerous simple individual entities that constitute the swarm system. The entities have the ability to achieve significant results as a team resulting from their interactions with the environment, and their local interactions between each other [28]. In our daily life, there exist many kinds of natural grouping systems that can produce interesting and unexpectedly complex collective behaviors which have been become sources of inspiration for many research domains. The most relevant ones are presented in the following sections:

Bird flocking

Birds, especially starlings have the ability to coordinate their movement without any mistake as if an external force were driving them to achieve a well-timed ballet. They can exhibit an astounding collective complex behavior once they are flying together for food searching or long-distance migration (see Figure 2.1a). When they are flying, they can form large flocks which can move synchronously, fluidly and quickly with the possibility to expand, contract and change shape at any moment [29]. Birds also have efficient social interaction that enables them to [30]: (1) avoid collisions during flying even while they often change direction unexpectedly, (2) scatter and quickly regroup when reacting to external threats, and (3) keep away from predators. It has been observed that the amazing resulting coordination in movement when flock of birds fly is achieved through visual communication between them, the coordination maneuver can be initiated by any birds without any leader controller. It can be emerged naturally as each individual follows a few simple rules [7].

Ants' colonies

Since millions of years, ants have remarkably been succeeded to survive in different ecological environments wherein other livings did not. The secret behind this is that ants have been exceptionally demonstrated an incredible social organization between them (see Figure 2.1b). Ants have been geniusly stunned biologic researchers on how they collectively behave as a colony to solve problems unthinkable for individual ants. They are able to communicate, cooperate and divide daily responsibilities in order to accomplish their tasks such as finding the shortest path to the best food source, building architectural nests with tunnels and chambers, defending a territory from neighbors, or allocating workers to different tasks [30, 31]. The puzzling thing in these all tasks is that they are accomplished with: no generals command ant warriors; no one's in charge; no managers' boss ant workers. The queen does not supervise at all the activity of the colony; it acts no role except laying eggs. Even with half a million ants, a colony continues to function just fine with no management at all.

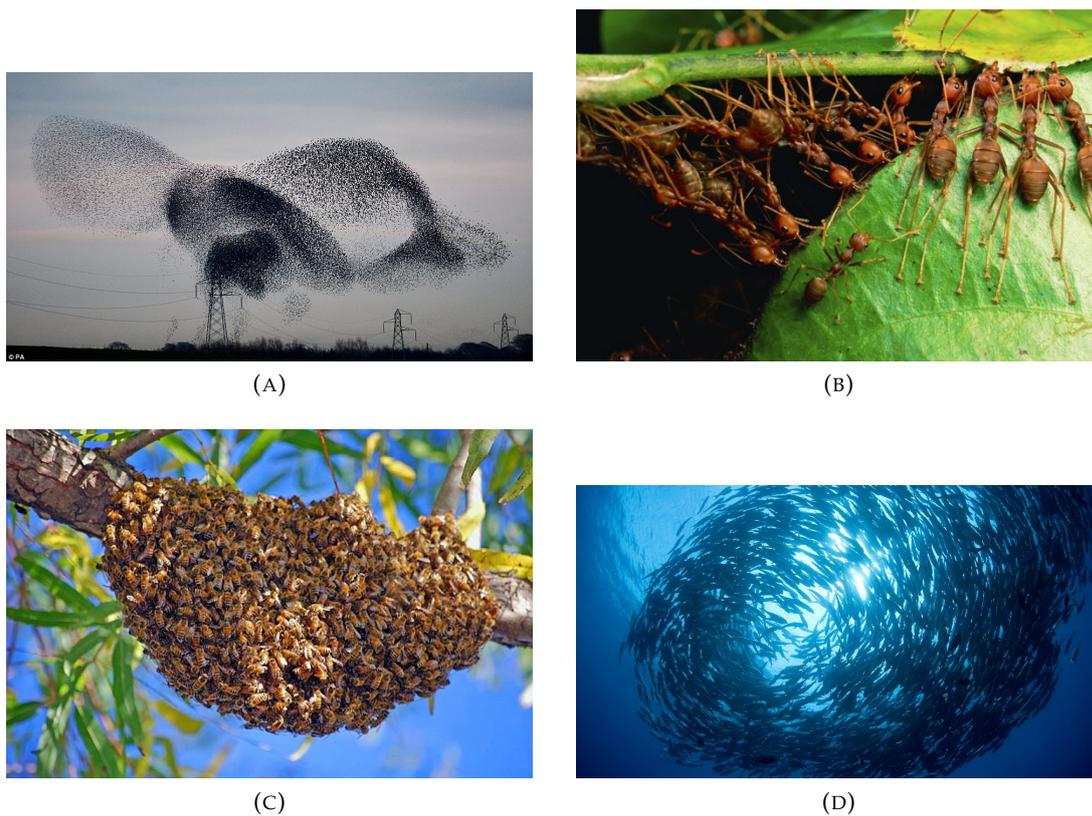


FIGURE 2.1: Examples of natural swarm systems: (A) Thousands of starlings flying together and forming a large flock¹. (B) Weaver ants pulling leaves so as to glue them together and build their nest². (C) Swarming bees, looking for a new nest site³. (D) Fish moving together in schools so as to better detect predators and evade their attacks⁴. Images used under the Creative Commons License.

Swarming of bees

One of the fantastic complex behaviors that are exhibited in bee colonies is what biologists called swarming (see Figure 2.1c). This spring phenomenon is considered as the natural mode of honey bee reproduction. It has been noticed that this behavior is proceeded once a bee colony outgrows its hive, leading to the departure of nearly half of workers with their mother queen to form new honey bee colony. The process is temporarily begun by forming a cluster (bivouac) on a tree branch as example from which a new site finding process is started [32]. Firstly, the environment is inspected by about 5 of the bees at the bivouac (scout bees) to search for new suitable nest site. Once the new site is discovered, the scout bees return to the bivouac where a waggle

¹http://i.dailymail.co.uk/i/pix/2013/11/27/article-2514252-19A6D97A00000578-863_972x511.jpg

²<http://4fs63j47srdk3eoozh18a6ij182.wpengine.netdna-cdn.com/wp-content/uploads/Mark-Moffett3.jpg>

³<https://www.explorenature.org/wp-content/uploads/2014/06/swarm1.jpg>

⁴http://newsletter.billbeardcostarica.com/wp-content/uploads/2015/02/sala-cocos10-baitball_18527_600x450.jpg

dance is performed to recruit some other scout bees which also can re-do the same actions until the desired new nest site is visited by a sufficient number of scout bees. At this moment the rest of the cluster is advertised that it's time to prepare for the entire migration to the new nest site. Biologists have observed that this amazing simultaneous process of migration is launched through 3 types of signals: (1) the shaking signal for activating the quiescent bees, (2) the piping signal for initiating the warm-up of the flight muscles and the (3) buzz running signal for the liftoff of bees [33].

Fish schools

When groups of fish swim together in a synchronized fashion, move in the same direction at the same speed, and turn simultaneously, yielding to form simple ellipsoids patterns to complicated vortex arrangements (see Figure 2.1d), they are exercising what biologist called schooling. The shape of patterns can be long term preserved even though individual fish are always coming and going [30]. Alike the emergent complex behaviors observed in birds flocking, fish schools are exhibiting their shows without having any form of a social leader. The school isn't perceived as a whole entity by individuals, further each individual fish is not intelligent enough to create such complex patterns by choice and even is lacking the information to know its location in the high density of its school. Fish in schooling have the ability to stay on a high density group without any common goals as an internal force is attracting them. They are governed by simple rules such as: desire for food, attraction to their own species, predator evading, collision avoidance, etc [34].

2.1.2 Swarm Intelligence Systems: Definition and Properties

It has been observed that the intelligence behind the collective ability of all these examples of natural swarm systems seems to be miraculous even for the biologists who know them well. To understand where this intelligence comes, it should be find answers to many fundamental questions such [31]: How does the complex behavior of a group is added up by the simple actions of individuals? How critical decisions are made by hundreds of honeybees about their hive even if many of them disagree? What enables high precision movements' coordination in a school of fish and flock of birds like a single, silvery organism even they change direction in a flash?

Searching answers to these questions yield scientists to deeply study these natural swarm systems where many common characteristics are shared between them. The most common idea resides on the high degree of coordination displayed when groups of individual swarms are performing their activities. The fascinating part is that in addition to none of those individual entities grasps the big picture, but each contributes to the success of the group. The groups also behave as a single entity, but the resulted collective behavior observed is the outcome of numerous individual actions performed at the same time. The individual entities are interacting locally with each other and their environment following a simple set of rules even their understanding

of the collective activity is weak or nonexistent (stigmergy) [29, 7, 31]. The groups are organized in the way that there isn't any leader controlling the entire entities, and there is no clear hierarchy among the individuals. This type of organization that apparently evades any predefined structure is called decentralized self-organization [29].

Inspired by these natural swarm systems and their amazing features, a new research field called Swarm Intelligence has been raised. This increasing domain' research which is firstly introduced in the context of cellular robotic systems by Beni and Wang [35], is considered as a sub-field of artificial intelligence based around on the study of collective behavior in decentralized, self-organized systems [36].

Although there is no a specific definition for swarm intelligence, we adopt heir the one denoted by Dorigo, Birattari, et al. [37]:

'The discipline that deals with natural and artificial systems composed of many individuals that coordinate using de-centralized control and self-organization. In particular, the discipline focuses on the collective behaviors that result from the local interactions of the individuals with each other and with their environment'.

So, a swarm intelligence system consists typically of a population of relatively simple agents (relatively homogeneous or there are a few types of them [37]) interacting only locally with themselves and with their environment, without having a global knowledge about their own state and of the state of the world. Moreover, the overall observed behavior is emerged in response to the local environment and to local interactions between the agents that follow often very simple rules [38]. A Swarm intelligence system has a fascinating dimension residing on its ability to act in a coordinated manner even with the absence of an external coordinator. Therefore, even though no individual is in charge of a group, the group still shows overall intelligent behavior.

From the above presentation, a Swarm intelligence system is characterized by a set of special features such as:

- **Robustness:** This feature is defined by Kitano [39] as a property that allows a system to maintain its functions despite external and internal perturbations. It means that the system should still perform even if some individuals fail.
- **Adaptiveness (Flexibility):** Adaptiveness is a basic biological phenomenon, whereby an organism becomes better suited to its habitat. This means that the system has the ability to adapt to any changing environment.
- **Scalability:** This means that the high levels of system functionality should be maintained even though the size of individuals is increased. The behavior of the whole swarm shouldn't be perturbed by adding a new individual which can only influence the behavior of a few others. In artificial systems, this is extremely significant since the performance of a scalable system can be increased by simply increasing the size without having the need to reprogram.

- **Self-organization (decentralized control):** Widely well-known in biological systems such as cells, organisms and groups that possess a large number of sub-units. The individual subunits are working as a group without neither local supervision nor central control.
- **Parallel functionality:** This is possible in a swarm system as different operations at different places at the same time can be performed by different individual entities. This helps to make an artificial swarm intelligence system more flexible, and enables it to powerfully self-organize and perform different aspects of a highly complex task.

2.1.3 Natural Swarm Behavior based Meta-Heuristics Algorithms

Natural swarm based theories have been applied to solve analogous engineering problems in several domains' engineering from combinatorial optimization to routing communication network as well as robotics applications, etc. (for a recent comprehensive review, readers can refer to [40]). The most well-known swarm based algorithms are: Ant Colony Optimization Algorithms (ACO), Particle Swarm Optimization Algorithms (PSO), Artificial Fish Swarm Algorithm (AFSA) and Bee based Algorithms. The ACO algorithm is inspired from the foraging behavior of ant colonies in finding shortest paths from their nests to food sources. The source of inspiration of PSO based algorithms comes especially from the behavior observed in bird flocking or fish schooling when they are moving together for long distances to search for food sources, whereas The AFSA algorithm is inspired from the collective movement observed in the different behaviors exhibited by fishes such as searching for food, following other fishes, protecting the group against dangers and stochastic search [41]. Bee based algorithms can be classified into three different main groups: (1) the honeybee' foraging behavior based algorithms, (2) the ones based on mating behavior in honeybee, and (3) the queen bee evolution process based algorithms (more details can be found in [42]).

2.2 Swarm Robotics – Swarm Intelligence applied to Multi-robot systems

2.2.1 Multi-robotics

Multi-robot systems (MRS) are born to overcome the lack in information processing capability and many other aspects of single robots that are not capable to deal with special tasks, which in order to be efficiently completed need cooperation and collaboration between group of robots [2]. Since its introduction in the late 1980s, various works (such as: cellular robotics, collective robotics, and distributed robotics) have been issued to describe group of simple physical robots collaborating together to perform specific tasks. MRS have also achieved a great success and made a great progress

in many areas such as cooperative transportation and aggregation, environmental monitoring, search-and-rescue missions, foraging, and space exploration [7].

In such tasks, even the simplicity in design and the low-cost in productivity, as well as the increase in capabilities, flexibility, and fault tolerance advantages gained when using multi-robots instead of a single one. However with the new arising challenges such as decentralization in control and self-organization, researchers in multi-robotic field begun to make attention to the increase progress known in swarm intelligence systems giving birth to the new sub-domain of research called “*swarm robotics*”.

2.2.2 Swarm Robotics

Swarm robotics is a very particular and peculiar sub-area of collective robotics in which swarm intelligence techniques are applied. The 2000 year has witnesses the first project “swarm-bot” [43] that has been marked as the real period of the development of swarm robotics. The project was shared by the inventor of ant colony algorithm Marco Dorigo, and it aimed to study new approaches to the design and implementation of self-organizing and self-assembling artifacts.

Dorigo et al. [43, 44] ones of the founders of swarm robotics gave a definition to this research domain as follow : “*Swarm robotics can be loosely defined as the study of how collectively intelligent behaviour can emerge from local interactions of a large number of relatively simple physically embodied agents*”. The main idea of the approach behind this field of research is to build relatively many small and low-cost robots that are supposed to accomplish the same task as a single complex robot or a small group of complex robots [45]. The approach also takes into account studying the design of robots (both their physical bodies and their controlling behaviors) in a way that a desired collective behavior emerges from the inter-robot interactions and the interactions of the robots with the environment [46]. Further, as the key properties (pointed out in [11]) of a typical SI system can be applied to either MRS and Swarm robot systems (SRS), a set of criteria has been highlighted by Şahin [47] to overcome the confusions raised about the use of the term “swarm” and the overlapping meanings applied to multi-robot research. Dorigo and Shahin’ set criteria that are not meant to be used as a checklist, rather they help evaluating the degree to which SR might be applied and how it might be different from other MRS, are described as follow [7]:

- **Autonomy:** A SR system is made up of autonomous robots that are able to physically interact with the environment and affect it.
- **Large number:** A SR system should be consisted of limited homogeneous groups of robots in which each group contains of large number of members. Hence, highly heterogeneous robot groups tend to fall outside swarm robotics.
- **Limited capabilities:** A SR system is composed of robots relatively incapable or inefficient to carry out tasks on their own but they are highly efficient when they cooperate.

- **Scalability and robustness:** A SR system should be scalable and robust. Increasing the number of unites will improve the performance of the overall system and on the other hand, reducing some units will not yield to a breakdown of the system.
- **Distributed coordination :** In SRS, the coordination between robots is distributed, each robot should only have local and limited sensing and communication abilities.

Based on this set of criteria, SRS are more beneficent than MRS which might be used whenever several robotic platforms are applied to achieve a mission. The main benefits when using SR reside on [45]: (1) the robustness feature explained by the coherency of the whole system when losing some robots. This can gain us money investment in hundreds of small swarm robots, rather than investing the same amount of money or greater in a single complex robot that can leads to the failure of the all over project if a single failure is persisted. (2) The flexibility feature enlightened by rather needing a hardware reconfiguration of complex robots to accomplish a task, the same task is achieved by coordinated swarm robots that are not essentially personalized to a given task. (3) The scalability feature described by the fact that relying only on local information, a swarm robotic algorithm can be applied unchanged to a group of any (reasonable) size.

Further, with the advances occurred in swarm robotics research domain and the continuous coming researches in MRS such as [11] “minimalist robotics”, “robot colonies” , “distributed robotics”, and “large-scale minimalist multi-robot systems”. Three sub-areas of swarm robotics have been proposed by Sharkey [11] to overcome the lack in clarity about the level to which biological inspiration continues to be applicable to swarm robotics, and about the possible interpretation of the communication and control methods used for single robots. The first distinguished sub-area is “scalable swarm robotics” in which decentralization and scalability are the key features of any control and communication mechanisms used in the system, otherwise possible constraints on the simplicity of single robots are ignored. The second sub-area is “Practical Minimalist swarm robotics” which as well as it underlines a decentralized control and communication mechanism, it also emphasizes the simplicity of the sensing abilities of the individual robots without interesting to take into account the recent biologic researches. While the third sub-area is “nature-inspired minimalist swarm robotics” which in contrast to the second sub-area, it embraces the self-organization feature - most commonly exhibited in biological systems – as constrains about the complexity of individual robots.

Additionally to the above differentiation and classification of swarm robotics and beyond the underlined confusion made in the above discussion, Tan and Zheng [9] published a recent research paper in which another differentiation study that differs swarm robotics from other multi-agent systems is undertaken. The study aims to overcome the confusion mad between the swarm robotics research area and the

other research areas such as multi-agent system, multi-robots system and sensor network. The comparison study highlights other critters than the classification done by Sharkey [11]. The table below as it's deducted from [9] summarizes these critters of differentiation.

TABLE 2.1: Comparison of swarm robotics systems and multi-Robotics Systems.

| | Swarm robotics systems | Multi-robotics systems |
|----------------------|---|--|
| Population Size | Variation in great range | Small |
| Control | Decentralized and autonomous | Centralized or remote |
| Flexibility | High | Low |
| Scalability | High | Low |
| Environment | Unknown | Known or unknown |
| Motion | Yes | Yes |
| Typical applications | Post-disaster relief, Military application, Dangerous application | Transportation Sensing, Robot football |

2.2.3 Potential Application of Swarm Robotics

Since the emergent of swarm robotics research field, several works have been issued to explain how we can benefit from the properties of swarm robotics systems that make them appealing in several potential application domains. Swarm robotics have been involved in many tasks [48] such as the ones demanding miniaturization like distributed sensing tasks in micro-machinery or the human body, those demanding cheap designs such as mining task or agricultural foraging task, those requiring large space and time cost and are dangerous to the human being or the robots themselves such as post-disaster relief, target searching, military applications, etc. Refers to Tan and Zheng [9] and Tan [48], swarm robotics is mostly used in:

- **Tasks covering large area:** Swarm robotics can be applied in tasks that require a large region of space. Heir, the robots are specialized for large coverage tasks (e.g. surveillance, demining, and search and rescue) and they are distributed in an unstructured or large environment (e.g. underwater or extraterrestrial planetary exploration) in which no available infrastructure can be used to control the robots. In such tasks, robot swarms are well-matched because they are able to: act autonomously without the need of any infrastructure or any form of external coordination, detect and monitor the dynamic change of the entire area, locate the source, move towards the area and take quick actions. Moreover the

robots, in such urgent situation, can aggregate into a patch in order to block the source as a temporary solution.

- **Tasks dangerous to robot:** In several dangerous tasks such as mine rescue and recovery, robots may be irretrievable after the task is accomplished. Thus, it's economically acceptable to use swarm robotics with simple and cheap individuals rather than using complex and expensive robots. Moreover, it's reasonably tolerable to apply swarm robots that provide redundancy for dealing with such dangerous tasks.
- **Tasks require scaling population and redundancy:** Swarm robotics can be also applied in situations where it is difficult or even impossible to estimate in advance the resources needed to accomplish tasks such as search and rescue, tracking, and cleaning. An example for this situation is: clearing oil leakage after tank accidents. Heir at the beginning of the task the population of swarm is highly maintained when the oil leaks fast and it's gradually reduced when the leak source is plugged and the leaking area is almost cleared. The solution needed in these cases should be scalable and flexible. Therefore, a robot swarm could be an appealing solution, robots can be added or removed in time without any significant impact on the performance to provide the appropriate amount of resources and meet the requirements of the specific task. This can be respected by the robustness feature of swarm robotics, and which is the main benefits from redundancy of the swarm.

2.2.4 Swarm Robotics Problems Focus

In the last decade, swarm robotics researches has known a significant progress due to the advantages gained when using such technology to solve many problems that are beyond the capabilities of classical multi-robots systems. The problems involves in swarm robotics research can be classified into [48]: those mainly based on the patterns (e.g. aggregation, cartography, migration, self-organizing grids, deployment of distributed agents and area coverage), those focused on the entities in the environment (e.g. Searching for the targets, detecting the odor sources, locating the ore veins in wild field, foraging, rescuing the victims in disaster areas and), and those mostly hybrid of the two previous problems (e.g. cooperative transportation, exploring a planet and navigating in large area).

Brambilla et al. [8] Illustrates another classification of the problems involved in swarm robotics based on the collective behavior problems focus. In Table 2.2 we summaries his study basing on giving: a short definition of the problem to be solved, its source of inspiration, the approaches used to model the problem, examples of the current researches that belongs to the problem, and finally the classification of the problem.

TABLE 2.2: Problems Classification in Swarm robotics.

| Problematic | Sources of inspiration | Modeling approaches | Research literatures samples | Classification |
|---|--|---|------------------------------|--------------------------------|
| Aggregation: Clustering swarm robots in a region of the environment. | <ul style="list-style-type: none"> Nature (e.g. Aggregation bacteria, cockroaches, bees, fish and penguins). | <ul style="list-style-type: none"> Probabilistic finite state machines. Artificial evolution. | [49, 50] | Spatially organizing behaviors |
| Pattern formation: Deploying robots in a regular and repetitive manner from which specific distances are kept between each other in order to create a desired pattern. | <ul style="list-style-type: none"> Biology (e.g. the spatial disposition of bacterial colonies and the chromatic patterns on some animals). Physics (e.g. molecules distribution and crystal formation). | <ul style="list-style-type: none"> Virtual physics-based design. | [16] | |
| Chain formation: Auto-Positioning robots to connect into two points. The chain that they form can then be used as a guide for navigation or for surveillance. | <ul style="list-style-type: none"> Foraging ants. | <ul style="list-style-type: none"> Probabilistic finite state machines. Virtual physics based design. Artificial evolution | [51, 52] | |
| Self-assembly and morphogenesis: Connecting physically swarm robots to each other to create structures (morphologies). | <ul style="list-style-type: none"> Ants (bridges, rafts, walls...). | <ul style="list-style-type: none"> Probabilistic finite state machines. Virtual physics based design. Artificial evolution | [53] | |
| Collective exploration | <ul style="list-style-type: none"> Social animals (ants, bees...). | <ul style="list-style-type: none"> Probabilistic finite state machines. Virtual physics-based design. Network routing | [54, 55] | |
| Coordination motion: Moving in formation similarly to schools of fish or flocks of birds. | <ul style="list-style-type: none"> Flocking in-group of birds. Schooling in group of fish. | <ul style="list-style-type: none"> Virtual physics-based design. Artificial evolution | [56] | |
| Collective transport: Cooperating in order to transport an object. | <ul style="list-style-type: none"> Cooperative carry prey in ant colonies. | <ul style="list-style-type: none"> Probabilistic finite state machines. Artificial evolution | [57] | |
| Consensus achievement: Reaching consensus on one choice among different alternatives. | <ul style="list-style-type: none"> Ants' decision between the shorter of two paths using pheromones. Bees' decision between the best foraging area and the best nest location. Aggregation in Cockroaches | <ul style="list-style-type: none"> Direct communication. Indirect communication. | [49, 58] | Collective decision making |
| Task allocation: Auto-distribution of swarm robots over different tasks To maximize the performance of the system. | <ul style="list-style-type: none"> Task allocation in ant and bee colonies. | <ul style="list-style-type: none"> Probabilistic finite state machines. | [59] | |

2.2.5 Involved projects and simulations

Swarm robotics involved projects

From the emergent of swarm robotics as a novel research domain, several successful projects have been created in order to face the challenges raised in this area of research. The most known projects are presented in Table 2.3.

Swarm robotics simulation platforms

Using plenty of physical robots in swarm robotics researches is hardly difficult to afford. Thus, computer simulations are developed to visually test the structures and algorithms on computer before engaging in real physical robots tests. The use of computer simulations, which are generally easier to setup less expensive, are normally faster and more convenient to use than physical swarms. In the section below we highlight the well-known widely used simulation platforms in swarm robotics researches.

- **Player/stage**

Player/stage⁵ is a combined package of free Software tools for robot and sensor applications developed by the international team of robotics researchers under the GNU license. Player component is one of the most widely used robot control interface in the world that provides a network interface to a variety of robot and sensor hardware. The control of robots can be programmed through multi-programming language that can be run in any computer with a network connection to the robot. Stage component is a multiple robot simulator interfaced to Player, it simulates a population of mobile robots moving in and sensing a two-dimensional 2D bitmapped environment.

- **Gazebo**

Gazebo⁶ is a simulator that extends Stage for 3D outdoor environments. It includes an accurate simulation of rigid-body physics. Hence, both realistic sensor feedback and possible interactions between objects can be then generated. Gazebo presents a standard Player interface in addition to its own native interface. In this way, the controllers written for Stage can be used in Gazebo and vice-versa.

- **UberSim**

The UberSim⁷ is a simulator developed at Carnegie Mellon for a rapid validation before uploading the program to real robot soccer scenarios. UberSim uses ODE physics engine for realistic motions and interactions. Although originally

⁵<http://playerstage.sourceforge.net>

⁶<http://gazebosim.org/>

⁷www.cs.cmu.edu/~robosoccer/ubersim

TABLE 2.3: Some of successful Swarm Robotics Projects.

| Project | Objective | Prototype |
|---|---|---|
| Open-source micro-robotic Project^a University of Stuttgart, Sergey Kornienko and University of Karlsruhe, Marc Szymanski, Ramon Estane | Develop a cheap, reliable and swarm-capable micro-robot that allows building a large-scale swarm system to investigate artificial self-organization, emergent phenomena, and control in large robotic groups. This research is important to understand underlying principle of information and knowledge processing, adaptation and learning for the design and development of very limited autonomous systems. | Jasmine  Cost: £80 Sensor: distance, light, bearing Motion/Speed: wheel, 50 cm/s Size: 3cm Autonomy: 1-2h |
| Swarm-bots^b project IRIDIA, Université Libre de Bruxelles | The project explores the design, implementation and simulation of self-organizing and self-assembling artifacts. The project after it was successfully completed in 2005; it has been extended by the swarmanoid project, a project that proposes a highly innovative way to build robots that can successfully and adaptively act in human made environments. The swarm-bot prototype has been also used in e-swarm project. | swarm-bot  Cost: N/A Sensor: range, bearing, light, camera, bump Motion/Speed: wheel, N/A Size: 12.7 cm Autonomy: 3h |
| E-puck education robot^c École Polytechnique Fédérale De Lausanne EPFL | The project develops a miniature mobile robot for education use. The robots have several features specialized for such purpose. The robots have a clean mechanical structure simple to understand, operate and maintain. The robots are cheap and flexible, and can cover a large spectrum of educational activities thanks to a large potential in sensors, processing power and extension. | e-puck  Cost: £580 Sensor: distance, camera, bearing, accelerometer, mic Motion/Speed: wheel, 13cm/s Size: 7.5cm Autonomy: 1-10h |
| R-one project^d Multi-Robot Systems Lab, Rice University | The project aims to provide an advanced low-cost mobile robots designed for research, teaching and outreach, the developed robots was successfully implicated in several projects such as multi robot manipulation, distributed approach for exploring and triangulating an unknown region, and distributed boundary detection. | R-one  Cost: N/A Sensor: distance, light, bump, IR, accelerometer, localization. Motion/Speed: wheel, 25cm/s Size: 11cm Autonomy: 4h |
| Kilobot project^e School of Engineering and Applied Sciences Wyss Institute for Biologically Inspired Engineering Harvard University | The project aims to design a robot system for testing the collective algorithms with a population of hundreds or thousands of robots. Each robot is made of low-cost parts and takes 5 min to be fully assembled. The system also provides several overall operations for a large swarm, such as updating programs, powering on, charging all robots and returning home. | Kilobot  Cost: £12 Sensor: distance, light Motion/Speed: vibration, 1cm/s Size: 3.3cm Autonomy: 3-24h |
| Khepera III robot^f K-Team Corporation | Produced by K-Team corporation, the robot provides a new standard tool for robotic experiments and demonstrations such as: artificial intelligence, navigation, multi-Agents System, real-time programming, control collective behavior, and advanced electronics demonstration. | Khepera III  Cost: N/A Sensor: range, bearing, camera, bump, IR, light,... Motion/Speed: wheel, 50cm/s Size: 13x7cm Autonomy: 8h |

^a<http://www.swarmrobot.org>^bwww.swarm-bots.org^c<http://www.e-puck.org>^d<http://mrsl.rice.edu/projects/r-one>^e<http://www.eecs.harvard.edu/ssr/projects/progSA/kilobot.htm>^f<http://www.k-team.com/mobilerobotics-products/khepera-iii>

designed for Soccer robots, custom robots and sensors can be written in C in the simulator and the program can be uploaded to the robots using TCP/IP.

- **USARSim**

USARSim⁸, shorted for Unified System for Automation and Robot Simulation, is a high fidelity multi-robot simulator originally developed for search and rescue (SAR) research activities of the Robocup contest. It has now become one of the most complete general purpose tools for robotics research and education. It is built upon a widely used commercial game engine, Unreal Engine 2.0. The simulator takes full advantage of high accuracy physics, noise simulation and numerous geometrics and models from the engine. Evaluations have shown that USARSim can simulate the real time robots well enough for researchers due to the high fidelity physics engine.

- **Enki**

Enki⁹ is an open source software released under the GNU license, it is a fast 2D physics based robot simulator written in C++. It is able to simulate the robot swarms hundred times faster on the desktop computer than real-time robots. It is also able to simulate the kinematics, collision, sensors and cameras of robots working on a flat surface. Enki is built to support several existing real robot systems, including swarm-bots and E-pucks, while user can customize their own robots into the platform.

- **Webots**

Webots¹⁰ is a development environment used to model, program and simulate the mobile robots available for more than 10 years. With Webots, the user can design the complex robotic setups, with one or several, similar or different robots with a large choice of pre-defined sensors and actuators. The objects in the environment can be customized by the user. Webots also provides a remote controller for testing the real robots. Until now, Webots robot simulator has been used in more than 1018 universities and research centers in the worldwide.

- **Breve**

Breve¹¹ is a free open-source software package which makes it easy to build 3D simulations of multi-agent systems and artificial life. Behaviors and interactions of agents are defined using Python. Breve uses ODE physics engine and OpenGL library that allows the observers to view the simulation in the 3D world from any position and direction. Users can interact at run time with the simulation using a web interface. Multiple simulations can interact and exchange individuals over the network.

⁸<http://sourceforge.net/apps/mediawiki/usarsim/>

⁹<http://home.gna.org/enki/>

¹⁰<http://www.cyberbotics.com/>

¹¹www.spiderland.org/breve/

- **V-REP**

V-REP¹² is an open source 3D robot simulator that allows creating entire robotic systems, simulating and interacting with dedicated hardware. V-REP is based on distributed control architecture. Each object/model can be individually controlled via an embedded script, a plugin, a remote API client, or a custom solution. In V-REP, Controllers can be written in C/C++, Python, Java, Lua, Matlab, Octave or Urbi and can be directly attached to the objects in the scene and run simultaneously in both threaded and non-threaded fashions. This makes it very versatile and ideal for multi-robot application. V-REP is used for fast algorithm development, factory automation simulations, fast prototyping and verification, robotics related education, remote monitoring, safety double-checking, etc.

- **ARGoS**

ARGoS¹³ was the official simulator of the Swarmanoid project. It is currently the main robot simulation tool for many European projects. ARGoS is a new pluggable, multi-physics engine for simulating the massive heterogeneous swarm robotics in real time. Contrary to other simulators, every entity in ARGoS is described as a plug-in, and it is easy to implement and use. In this way, the multiple physics engines can be used in one experiment, and the robots can migrate from one to another in a transparent way. Results have shown that ARGoS can simulate about 10,000 wheeled robots with full dynamics in real-time. ARGoS is also able to be implemented in parallel in the simulation.

- **TeamBots**

TeamBots¹⁴ is a collection of Java simulation for mobile robotics research. Some execution on mobile robots sometimes requires low-level libraries in C. TeamBots supports the prototyping, simulation and execution of multi-robot control systems and is compatible with the Nomad 150 robot by Nomadic Technologies and Cyb robot by Personal Robotics.

- **MORSE**

MORSE¹⁵ is a Blender Game Engine based simulator designed to provide a realistic 3D simulation of small to large environments, indoor or outdoor, with the ability to simulate one to tenths of autonomous robots. It comes with a set of robots base model (such as quadrotors, ATRV, Pioneer3DX, generic 4 wheel vehicle, PR2,...), with the possibility to add new ones.

¹²<http://www.coppeliarobotics.com/>

¹³<http://iridia.ulb.ac.be/argos/>

¹⁴www.teambots.org

¹⁵<https://www.openrobots.org/wiki/morse/>

2.3 Summary

Swarm robotics is a relatively new research area that takes its inspiration from swarm intelligence and robotics. It is the result of applying swarm intelligence techniques into multi-robotics. Although a number of researches have been proposed, it's still quite far for practical application. In the present chapter, an overview of swarm robotics has been given for a better understanding of this multi-robot domain' research and for clarifying the grand lines being focused on it. Interests that are newly coming to this topic of research can be easily guided through the different sections presented in this chapter.

Chapter 3

Studies in Aggregating Patterns and Faults Detection within Swarm Robotics systems

3.1 Overview

In this chapter, the main issues addressed in this dissertation are discussed. In particular, two focal concerns within robotic swarm systems are presented in details: (1) aggregating patterns and (2) fault detection. With the first concern, we provide first some of the studies in aggregating patterns in natural swarm systems, and then in spot of these studies, we review a number of relevant works on swarm robotics aggregating patterns. The works are classified into two main kinds of approaches: (1) a cue-based and (2) a self-organized based methods. We basically focus on the second method, and therefore we discuss in more details the different self-organized based aggregation approaches that have so far used in the literature. With the second concern, we address the problem of fault detection in swarm robotic systems. We first provide an overview about faults in engineered systems, and the different corresponding methods used for monitoring these faults. Finally, we project that in the field of swam robotics by reviewing the main fault detection approaches applied in such systems.

3.2 Aggregation Patterns in Nature

3.2.1 Aggregation in Nature

Aggregation (or gathering together) is a fundamental behavior that is observed in many biological organisms, such as social insects and group-living animals [60]. It is an important requirement for animal societies to accomplish complex swarming behaviors collectively. It can be helpful in different tasks like survival of individuals, avoidance of predators, increase of chances in finding foods, etc [61].

In nature, aggregation can be achieved using external gradients called cues, such as humidity for woodlice [62, 63] or temperature for honeybees [64]. This kind of

aggregation is mostly known as cue-based aggregation [65], where optimal zones are marked by a specific cue that initiate the aggregation process.

Aggregation can be also achieved in self-organized manner, where the aggregation process is enabled without any external cues [60]. Fascinating examples of such behavior can be observed in bird flocks, fish schools, and mammal herds [66]. In this kind of aggregation, no external cues are required to form aggregations, individuals rather form some random aggregating zones without any particular preference to their condition.

3.2.2 Emergence of Patterns in Aggregating Natural Swarms

In nature, marvelous self-organized patterns are observed in the collective behaviors of many biological organisms during their aggregation process. They are seen particularly in fish schooling, bird flocking, insect colonies and bacteria swarming [60, 67]. The observed patterns or shapes, which are defined as “*orders embedded in randomness*”, may look as spatial arrangements or temporal series, and its composing elements may seem identical or with variations [68]. Biologically, the process behind the creation and generation of these fascinating orders was termed “*pattern formation*”. It refers to the process through which a coherent set of associations between element’s states is formed and persists over some period of time, it captures the essence of self-organization and emergence in all kinds of systems [69]. An interesting question to be asked is how was these patterns generated or formed? An answer to this question, can make a fundamental distinction between those patterns that were created through order being imposed by some other external organization, or those that were created through the pattern being internally generated. Interesting on the ones being generated internally, it is believed that the spatiotemporal order at the group level of these self-organized patterns emerge only from simple local interaction rules among the lower-level components of the group. Moreover, these rules are specified by implying nonindependent individual decisions through local information transfer between group members [60, 70].

3.2.3 Case study: Flocking Patterns

Flocking patterns are one of the mesmerizing spatially self-organizing phenomena that are observed in a herd of animals of similar size and body orientation. They result often when a huge number of individuals move in mass or migrate in the same direction with a common group objective. Familiar examples of a such phenomenon can be seen in birds flocking and fish schooling. A “*murmuration*” of starlings in birds flocking, for example, is one of the most captivated phenomenon exhibited in winter in front of our eyes. It emerges when huge flocks of thousands of starlings fly in beautifully harmonized patterns, while maintaining group cohesion in highly uncertain environments[71]. While migrating, avoiding predation or wheeling above the roost; starlings are able to produce flocking patterns that are remarkably dynamic

and highly variable in shape [72]. Analogously, schooling of fish can produce instant patterns, change the formation of the patterns, and even recover to the initial formation when they are moving coordinately and responding to a predator attack [73].

Models on Flocking Patterns

Mathematically, several models have been developed to show how these complicated collective behaviors emerge by self-organization from a few simple interaction rules among individuals. One of the earliest and simplest individual-based models of birds flocking behavior refers to the Boids model of Reynolds [74]. In this model, the overall collective behavior emerges from the application of three heuristic rules (separation, alignment, and cohesion) (see Figure 3.1), which have been proven effective in many biological group behaviors [75]. Later, another popular yet collective behavior model

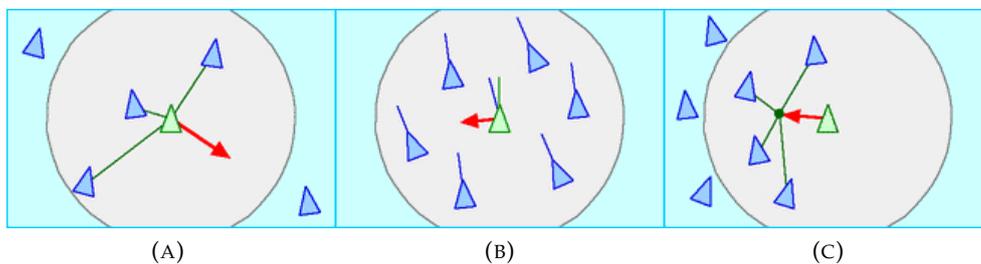


FIGURE 3.1: The three basic steering behaviors determining the motion of the boids. (A) Separation: reacting only to nearby flockmates (B) Alignment: steering towards the average heading direction of the local flock-mates. (C) Cohesion: moving toward the average position of the neighboring boids. (From <http://www.red3d.com/cwr/boids/>).

was proposed by Vicsek et al. [76]. The flocking agents of this model are described as self-propelled particles that are tending to move in the average direction of their neighbors in presence of external noise. Since then, several variants of the Visket model have been investigated in the literature [70]. Main directions involve models with no explicit alignment rule, models using additional cohesive terms, and models changing the noise term.

In other studies, researches turned to make more attention to the nature of aggregation patterns. In most of them, a set of Physical attraction/ repulsion (A/R) primitives are applied among individuals [77, 75]. As instance, in the A/R model of Gazi and Passino [77], a stable ring-shaped pattern has been analytically proven to be achieved in a finite time using a set of inter-molecule force laws. Cheng et al. [75] proposed an A/R model of self-driven particles to analyze the forming mechanisms of two distinct aggregation patterns: liquid-like and crystal-like clusters. It is observed that a radical transition between these two patterns depends only on the cutoff distance of the A/R function. In another study, Couzin et al. [78] investigated the spatial dynamics of flock of birds as well as fish schooling, and designed an A/R based self-organizing model that incorporate alignment rule to enhance group formation and cohesion among individuals.

Recently, Hildenbrandt, Carere, and Hemelrijk [79] demonstrated in their starDisplay model how to reproduce the flocking behavior of huge swarms of starlings (*Sturnus vulgaris*). The starling-like patterns generated by this model resemble remarkably not only qualitatively but also quantitatively, and the rules governing such phenomenon are the result of adding some specific rules of starling behavior (such as rolling during turning and moving above a roosting site) to the common rules of coordination based on separation, attraction, and alignment.

Moreover, the starDisplay model followed a topological range approach that was revealed in previous studies of birds flocking and fish schooling. Ballerini et al. [80, 81] showed empirically that the unpredictable, amazingly complex patterns formed by birds emerge from a topological distance approach rather than a metric distance approach, meaning that the birds only interact with their nearest six or seven neighbours rather than all of the neighbours in their field of vision.

Computer simulations predict that a significantly higher cohesion of the aggregation is achieved using a topological interaction rather than the standard metric one. Similarly, empirical results show that fish and elephants, for example, interact with only three or four neighbours [82].

3.3 Aggregation Patterns in Swarm Robotics

3.3.1 Aggregation in Swarm Robotics

On the basis of the biological studies cited in the previous section, aggregation is also a matter of interest in various swarm robotics studies. It is considered as a fundamental task that allows robot swarms to perform complex tasks, such as collective movement, self-assembly, and pattern formation, or to exchange information. It is a desired behavior that is being applied in multi-agent as well as swarm robots systems. Moreover, a lot of collective behaviors that are perceived in biological swarms and which some of them are possibly implemented in engineering of multi-agent and swarm robot systems emerge in aggregated swarms [83].

3.3.2 Emergence of Patterns in Aggregating Robotics Swarm

Self-organized patterns, one of the behaviors that emerge from swarms' aggregation, is an interesting characteristic of swarm robotics spatially organizing behaviors, which are recently taking an important interest in real world. It can be involved in several successful applications such as [84, 85]: surrounding an object or feature in the environment, forming a uniform distribution of robots in a given area for protecting the area or surveillance, election of a leader or follow-the-leader situations, gathering to share information or for some other tasks, removal of mines or bomb disposal, environmental exploration and mapping, formation of sensing grids, managing processes in a manufacturing unit, and carrying, moving and assembling objects.

Generally, the term of pattern in swarm robotics is used at least in two different area of research [86]. The first area focuses on the establishment, maintenance and reconfiguration of patterns; this is considered as a generalized form of a definition of a pattern formation [87]. In the same context, Bahceci, Soysal, and Sahin [88] defined the pattern formation problem as how a certain shape, such as a circle or a chain, might be formed and maintained by the coordination of a group of robots. The second area covers the global patterns that emerge from local interactions among individuals [89]. This is mostly similar to the self-organized patterns observed in flocking behaviors, where the resulting aggregation patterns are not explicitly planned.

Beyond this or that usage of the term pattern, the group of robots, and in order to perform its tasks, should be organized into global formations or patterns that might be varied from simple patterns to complex one. The formation of these patterns may refer either to [90]: geometric patterns challenged by the development of behaviors such that the desired geometric patterns are formed by individuals of swarm with a focus on the inter-interactions among them, or functional patterns dictated by the environment where the geometrical shape or size of the patterns formed are partially determined by the task at hand.

3.4 Approaches on Swarm Robotics Aggregation Patterns

In this section, we discuss the different approaches and the related works that have been addressed to study aggregation patterns within swarm robotics systems. As in the biological counterpart, most of the related studies that have been completed in the last two decades are categorized into cue-based aggregation and self-organized aggregation methods.

3.4.1 Cue-Based Methods

These methods are inspired from its biological counterparts where special signals or cues are used to activate the aggregation process.

In one study of cue-based aggregation by Kube and Zhang [91], a light source was used to aggregate a robot system around an object and to transport the object in a collaborative manner to another goal; by following simple behaviors (e.g., to find light and follow light), the aggregation task was completed with no explicit communication mechanisms. Holland and Melhuish [92] proposed a method based on an infra-red (IR) cue to regulate the size of an aggregate created by a robotics system that successfully allowed each robot to approximate the aggregate size and decide to join or leave the aggregate accordingly. Mermoud et al. [93] used a cue-based aggregation approach to address the problem of collective decision-making in swarm robotics system. In that study, the robots applied a probabilistic aggregation mechanism that first allowed them to aggregate in a good or bad location; then, based on the status of the location, the robots made a collective decision whether to keep or destroy the aggregate in that location.

Recently, one of the most successful cue-based aggregation models took inspiration from the collective behavior of honeybees, which prefer gathering where the temperature is 36°C. The BEECLUST model proposed by Kernbach et al. [64] was the first algorithm that mimicked this behavior; a gradual light source was used to generate clustering behavior in a swarm robotics system. It was proven to act robustly in many researches [94, 95, 96, 97]. Further, different variations of the model have been suggested to increase the performance of the aggregation process. For instance, Arvin et al. [98] proposed a new aggregation algorithm in which a dynamic velocity and a comparative waiting time were introduced to the original BEECLUST model, which contributed to a significant improvement in the aggregation time. Furthermore, a comparison between the original BEECLUST algorithm and two modified versions - called the vector averaging algorithm and the naive algorithm - showed that both the vector averaging and naïve algorithms outperformed the original BEECLUST model, and revealed that noise has less impact in the vector averaging method than the naïve one [96].

Later, those authors introduced a fuzzy-based aggregation approach to enhance the performance of BEECLUST in both computer-based simulations and real robot swarms [65]. Wahby, Weinhold, and Hamann [99] proposed another adaptive variant of BEECLUST, where the original algorithm was extended to adapt automatically to any light conditions. Recently, Vardy [100] proposed a model called ODOCLUST, which incorporated odometry as an additional capability to BEECLUST. The ODOCLUST variant achieved a fast and accurate aggregation without requiring high-fidelity odometry.

3.4.2 Self-organized based Methods

In self-organized based methods, aggregation patterns are achieved using simple local interaction rules among individuals. In the following sub-sections, we highlight the different approaches that have been so far proposed in the literature.

Probabilistic Approach

Most of the works in this approach used a probabilistic finite state machine to control the behavior of the swarm. For example, Garnier et al. [101] adopted a probabilistic approach, inspired by the cockroach model of Jeanson et al. [102], to achieve aggregation using a swarm of 20 physical Alice robots in homogeneous environments. A similar work by Correll and Martinoli [103] showed that when using probabilistic aggregation rules, a minimum combination of communication range and locomotion speed was needed to achieve a single aggregate cluster. Soysal and Sahin [104] suggested a probabilistic aggregation method in which a state-finite machine was used to combine a set of simple behaviors that included avoiding an obstacle, approaching, repelling, and waiting.

Deterministic Approach

In this approach, the robots generally build a connected visibility graph, and ensure that the graph is permanently maintained. Ando et al. [105] used an algorithm in this sense to study aggregation in a group of mobile robots with a limited sensing range. Later, the algorithm was generalized by Cortés, Martínez, and Bullo [106] to achieve an aggregation in arbitrarily high dimensions. The formation of the graph in these algorithms was based on the assumption that the robots were able to measure both the range and the angle of their neighbors. However, Gordon, Wagner, and Bruckstein [107] were able to achieve such an aggregation using only the angle measurement of the robot's neighbors. The aggregation performance of this last algorithm was later improved, by introducing an additional, crude range-sensing capability for differentiating whether neighboring robots were near or far [108]. In another study, De Genaro and Jadbabaie [109] used the Laplacian matrix to allow each robot to build its own proximity graph. The related control was fully decentralized, and simulated results demonstrated that the model was effective and even increased the connectivity of the entire swarm.

Artificial Evolution Approach

In some works, self-organized aggregation models have been approached using artificial evolution techniques. For instance, aggregation with simple robots, called s-bots, was studied by Trianni et al. [110]. In this study, general solutions to the aggregation problem were produced using an evolutionary robotics mechanism. The method was able to produce clustering behaviors with both static and dynamic behavioral strategies. With that model, Dorigo et al. [44] revealed that effective evolved controllers could be achieved for both aggregating and coordinated motion behaviors in a swarm of s-bots. In a similar setup, Soysal, Bahçeci, and ŞahİN [50] investigated the effects of a number of parameters, such as the robots' number, the size of arena, and the run time. In another study, Gauci et al. [111] proposed two algorithms - a reactive controller with no memory and a recurrent controller with memory- to study aggregation in a swarm of e-puck robots. The algorithms were based on a classical evolutionary programming technique, and used a simple binary sensor with a sensing range that proved sufficient to achieve an error-free aggregation. Results from both the simulation and experiments showed that aggregation toward one cluster was successfully achieved. However, a sufficiently long range in the binary sensor was needed to achieve an accurate aggregation.

Morphogenesis Inspired Approach

Biological morphogenesis, including its genetic and cellular internal mechanisms, has recently become a source of inspiration for many multi-robotics studies. This has given rise to the morphogenetic robotics [112] as a new emerging robotics research

field to study self-organization of swarm or modular robots. To address the problem of aggregation patterns in this context, Guo, Meng, and Jin [113] established a metaphor between multi-cellular systems and multi-robot systems to propose a decentralized GRN (Gene Regulatory Network) based algorithm for multi-robot shape construction. Through this GRN model, autonomous self-organization into different predefined shapes and adaptive self-reorganization under dynamic environments can be performed by multiple robots. Later, the authors proposed two extensions to the original model: by introducing a free-form shape representation to enable making more non-uniform rational B-spline complex 2D or 3D patterns [114], and through adopting an H-GRN (Hierarchical Gene Regulatory Network) model for adaptive multi-robot pattern generation and formation in changing environments [115].

Artificial Physics Approach

This approach, which belongs to the bio-inspired methods, takes inspiration from the observation of Physics. It was firstly introduced by Spears et al. [116] as a physicomimetics (or an artificial physics) framework. To control the behaviour of the whole swarm system, The framework makes use of virtual physics forces generated from the interactions of the robots. The framework was able, through using two types of physics force laws: Newtonian force law and Lennard-Jones force law, to drive large groups of aggregating agents moving into a desired formation such as a hexagonal lattice [117]. Further the framework is extended to handle moving formations through obstacle fields [118, 119].

Derived from this approach, different virtual physics forces laws have been so far applied in the literature. For example, Howard, Matarić, and Sukhatme [120] used virtual electric charges to model the deployment of robots into an unknown area, Moeslinger, Schmickl, and Crailsheim [121] applied repulsive and attractive virtual forces to investigate flocking behavior within swarm robotics. Gasparri, Priolo, and Ulivi [122] adopted also an attractive/repulsive virtual force model to study aggregation in a swarm of multi-robot systems based on local interaction. This model was later extended to cope with actuator saturation by Gasparri et al. [123] and to integrate obstacle avoidance by Leccese et al. [124]. In another work, Hashimoto et al. [125] suggested a control algorithm for a robotic swarm basing on the center of gravity of the local swarm, and this through making use of virtual forces, local forces and an advancing force laws.

Moreover, virtual spring based control models have recognized a significant interest in the last years. In these models, virtual spring forces are applied to maintain a desired distance among aggregated robots with providing some flexibility to the structure and smoothness to the movement. As instance in the work of Shucker and Bennett [126], a fully Distributed Robotic Macrosensor (DRM) control mechanism, involving a set of aggregation formation algorithms based on virtual spring mesh connectivity, is proposed to deploy a huge number of swarm robots system. The flexibility and the fault-tolerance of the aggregation formation mesh is guaranteed through

introducing a novel algorithm based upon an acute-angle test used to create a mesh of acute triangles. Bezzo and Fierro [127, 128] suggested a fully decentralized switched spring mesh model to investigate a multi-robots wireless communication navigation problem, which consists of moving the robots of the system in a swarming manner with maintaining communication connectivity while searching a moving target in a two-dimensional obstacle populated environment. The authors seek specifically to find the shortest path between a base station and one or more users (mobile target) that generate attractive potential fields around their center.

As an extension of virtual spring models, Virtual spring-damper based models have been also used in several studies. Some applications can be found in the work of Dewi, Risma, and Oktarina [129], where a wedge navigation formation is created by a flock of robots using simple virtual spring-damper model between the leader and the followers; the authors only use RF communication system and distance sensors between follower robots. In the work of Urcola et al. [130], the authors presented a virtual structure based on spring-damper elements to control a navigation system composed of leader and followers robots in formation movement. The navigation system can adapt the formation to the environment. Jeong and Lee [85] developed a virtual spring damper based dynamic model for an artificial swarm system, in which a dispersion and line aggregation formation algorithms are proposed to realize attractive and repulsive forces between the artificial agents and their neighbors; the dispersion algorithm is based on trigonal planner elements without a leader whereas the line formation algorithm use paired line elements with an interim leader.

3.5 Detecting Faulty Robots in Swarm Robotics Systems

3.5.1 Faults in Engineered Systems

In engineered systems, faults are among threats that could affect the dependability of any given system throughout its entire life cycle. In this context, the term fault was defined by Isermann [131] as *"an unauthorized deviation of at least one feature of the system from the acceptable, usual, standard condition"*. With this definition, a fault is considered as a state within the system, and the transition toward that state might be developed abruptly (stepwise), incipiently (drift-like) or intermittently (with interrupts) (see Figure 3.2). Relatively, one or more faults can cause a degradation in the performance of the system, and may lead to fail accomplishing the required function as expected. Consequently, a system that is possibly capable to continue operating in presence of faults even in a degradation performance manner, is then a fault tolerant system.

3.5.2 Faults in Swarm Robotics Systems

As discussed in section 2.2.3, swarm robotics systems can be beneficially applied in many real scenarios. Despite the robustness and the scalability that characterize them,

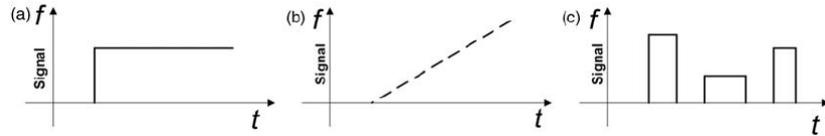


FIGURE 3.2: General time-independent fault forms Isermann [131]. (a) abrupt: fault arising suddenly in time, (b) incipient: fault developing slowly during the operation of a system, and (c) intermittent: fault occurring in discrete intervals.

they are also susceptible, similarly to engineered systems, to faults or external interferences that may lead in failing to accomplish the desired tasks successfully. [132, 133]. This can basically due to one or more faulty robots that lead to degrading the performances of the whole swam system. Practically, faults or failures that are needed to be addressed in the literature of swarm robotics are those causing disruption to the swarm's collective behaviour. These faults generally differ from one system to another and depend mainly on the behaviour that particular system is exhibiting at a particular time. In the context of this thesis, failures can be resulted from faults in the robot components such as bugs in its software controller or damages in its sensors/actuation devices. It can also due to topological faults such as damaged communication links between certain robots of the swarm.

Moreover, Timmis et al. [132] revealed that swarm robotics systems are less tolerance to partial failure robot(s) when compared to complete failure robot(s). As an example, a failure in a robot' component (i.e. motors) with the other sub-components functioning well, can effect the motion of the entire swarm. Bjercknes and Winfield [133] evaluated and analyzed the effect of such failure and other failures modes in a wireless connected robot swarms system both in simulation and real experimental platforms. The failures modes concerned in this analysis involved a robot' power failure mode, a robot' IR sensor failure mode, and a robot' motors failure mode. The reliability of robot swarms was analyzed with understanding which fault mode might be the serious source affecting the performance of the overall swarm. The study revealed that a potentially serious consequence of producing a partially failed robot that have an effect in the swarm movement is highly caused by robot(s) motor failures. Therefore, Bjercknes and Winfield [133] concluded that the consequence of partial robot failures should be critically considered when analyzing fault tolerance in robot swarms, and when designing robots swarm controllers.

Fault Types

Below are, details of some of the representative fault types for a simulated foot-bot robot basing on a cross-section of the literature that concerns this part of work.

Motor failure: In the Winfield and Nembrini [134] study, motor failure is by far the most damaging to collective behavior. In their study, motor failure is divided into two parts: (1) a complete failure of an individual's left motor and (2) a partial failure of the same motor. For a partial failure, the motor remains sensitive, but the associated

wheel rotates at half speed. In case of a complete failure, the engine completely stops and henceforward becomes unresponsive to its controller.

Sensor failure: Sensor failure is another common example used in research on fault-tolerant swarms [134, 135]. The exact details of a sensor failure vary from one robot to another, depending on their hardware. In the study of Winfield and Nembrini [134], it was again divided into a complete failure of the sensor and a partial failure of the sensor. In the event of a complete sensor failure, the range and bearing sensor (RAB), as well as its infrared proximity sensor (IR), failed completely and returned 0. In other words, the robot was completely unable to detect the presence from its neighbors or the walls of the arena.

Power failure: Again, a recurring example in fault tolerance work [134, 135]. A robot that experiences a power failure will stop moving completely and remain unresponsive to its environment. However, other robots in the swarm will still be able to detect its presence. This assumes that, in a physical system, each robot will be able to detect the presence of its neighbors regardless of the neighbor's response. This may not be the case for all systems, however, in cases where the robots are unable to communicate their presence for one reason or another, they will not be recognized by the swarm and will only become objects in the arena.

Software hang: The software crash was given as an example of a fault that requires exogenous, or at least partly exogenous, approaches to detect errors. Blocking the software locks a robot that performs the action it was performing at the last time it was operating normally. It should be noted that the robot allowed continuing broadcasting data to the swarm. This, again, assumes that robots can feel their neighbors independently, the justification of which is explained in the previous discussion of power failure.

3.5.3 Fault Detection in Engineered Systems

Fault detection is a binary decision process through which a decision is made about whether or not a fault has occurred in a system. To make such a decision, techniques in detecting faults generally use dependencies between variant measurable signals to extract information on possible changes that are caused by faults. In fact, fault detection has been a matter of interest for a long time in the field of engineered systems dating back to the beginning of 1980s. Different fault detection technics were applied so far in the literature.

Legacy Approach

First legacy techniques were based upon hardware redundancy. In this approach, special functional components of a system are duplicated in a way that they share the same input and therefore expect to produce the same output [136]. As consequence, any deviation in the output is then considered as a fault. Although by using hardware redundancy, faults are guaranteed to be detected in a fast, precise and dependable

manner; however, adding additional hardware components to the system increases cost and complexity, and therefore it is not practicable in many cases.

Model-based Approaches

As an alternative to the approach of hardware redundancy, model-based approaches tend to duplicate the redundant functional component via using a software model rather than a hardware component. Generally, a mathematical model is built based upon the experiences of a system's behavior (for an introduction see [137]). Then, any deviations resulting from the comparison between the actual behaviors and the predicted behaviors (derived from a mathematical model) are interpreted as symptoms of faults. The difference between the predicted and the observed value is called a residual. Figure 3.3 depicts a common basic structure that is followed by almost all model-based approaches. Unfortunately, deriving accurate models of monitored systems, can be difficult and time consuming.

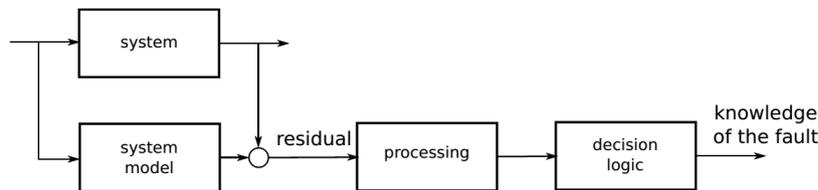


FIGURE 3.3: Schematic description of a model-based fault detection approach [138].

Data-driven Approaches

Data-driven implicit models are a suitable alternative in the absence of an explicit mathematical model, and if measurement signals are the only available resource for process monitoring. Alike the mathematical model-based approaches, data-based techniques also employ a model of the system. But in contrast, the model is learned from data gathered in the system rather than hand-crafting the model. Similarly, data-driven models efficiently extract useful features for the design of monitoring schemes, based on empirical models derived from the available process data. Such methods require minimal prior knowledge about process physics, but depends on the availability of quality input data.

A schematic description for data-driven fault detection approach is depicted in Figure 3.4. It can be considered as an extension to the model-based paradigm where, in order to learn a system model, a trainer module is added to exploit gathered input and output data of the system.

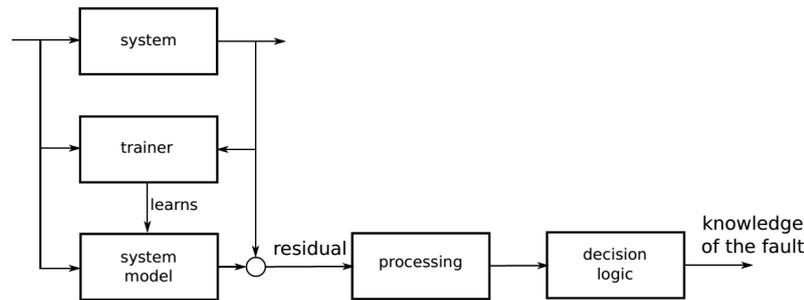


FIGURE 3.4: Schematic description of a data-driven based fault detection approach [138].

3.5.4 Fault Detection in Robotics Swarm Systems

Generally, faults in robot swarms are difficult to avoid and may result in serious system degradations [139]. Monitoring in swarm robotics has lately received special attention from researchers and practitioners in the field of safety engineering. Increased attention to fault detection and safety has led to the development of several fault detection techniques that can be grouped into two main families [140, 7]: endogenous and exogenous fault detection techniques. Both techniques can be developed using either a mathematical model or an empirical implicit model for fault detection.

Endogenous Approaches

Endogenous approaches are used to monitor each robot individually to reveal any faults. Several works report using such an approach; Skoundrianos and Tzafestas [141] used a local model neural network to diagnose faults in the wheels of a mobile robot. Yuan et al. [142] proposed a hybrid fault diagnosis approach based on Mittag-Leffler kernel (ML-kernel) support vector machine (SVM) and Dempster-Shafer fusion for wheeled robot driving system. Christensen et al. [143, 144] proposed a time-delay neural networks for automatic synthesis task-dependent fault detection modules in s-bot robots. Canham, Jackson, and Tyrrell [145] implemented an immune-based error detection method which takes inspiration from the negative selection process of the immune system on both a Khepra robot and a BAE system RascalTM robot. Mokhtar et al. [146] adapted a fault detection algorithm (called modified Dendritic Cell Algorithm mDCA) loosely inspired by the functioning of dendritic cells in the immune system. However, such approaches ignore the interaction between robots and therefore may result in a misleading diagnosis. For example, a robot might not detect anomalies in itself, such as a dead battery or a software bug, and it cannot even signal the rest of the swarm if an anomaly occurs in its communications hardware. In addition, these methods use only the data collected from, ignoring the data available in the whole swarm, which may result in the loss of pertinent information [140].

Exogenous Approaches

On the other hand, exogenous fault detection techniques were developed to inspect several robots simultaneously [147]. In other words, a robot could detect errors that arise in another robot's components by taking into consideration the available information of its neighborhood in the swarm [147]. Owens et al. [148], and Jakimovski and Maehle [149] proposed an AIS-based fault detection algorithm inspired by the T-Cell Receptor and intracellular signaling network mechanisms to detect anomalies within autonomous swarm systems. Christensen, OGrady, and Dorigo [139] proposed a firefly-inspired exogenous fault detection approach to detect inoperative robots in the swarm. Tarapore et al. [150] presented an AIS-based exogenous approach to detect faults in robotic swarm systems and tested its performance on different case studies that included aggregating, dispersing, flocking, and harming. Khadidos, Crowder, and Chappell [151] presented a model-based exogenous fault detection method based on broadcasting the sensor readings and motor speeds of robots to their neighbors. Millard, Timmis, and Winfield [152] proposed a run-time fault detection approach using an internal prediction model in each robot to compare with the real behavior of other robots in the swarm.

3.6 Summary

In this chapter, we reviewed the different approaches that have been used in the literature of swarm robotics to find solutions to the aggregating patterns and fault detection problems. The chapter delivered a strong image about the state of the art that puts the thesis in its context. First, basing on the aggregating patterns observed in nature, a number of studies in aggregating robots swarm have been discussed. These studies were classified into two main approaches: a cue-based and a self-organized methods. In the cue-based approach, the aggregation process is initiated using external gradients called cues or signals (e.g. light sources, sound signals, ... etc.), which are placed somewhere in the environment as specific marks that identify optimal zones where the aggregation should take place. In the self-organized approach, the aggregation process is activated somewhere in the environment without any requirement of external cues. Robots rather form some random aggregating zones without any particular preference to their condition. The self-organized approach has taken much focus in this chapter, and therefore more related studies have been discussed in the literature. Finally, the fact that swarm robotics systems are also susceptible, similarly to engineered systems, to faults or external interferences that may lead in failing to accomplish the desired tasks successfully, we reviewed a number of the relevant studies in swarm robotic fault detection approaches. The related works were classified into endogenous and exogenous fault detection methods. In the endogenous approach, each robot of the swarm has to be monitored individually to reveal any faults, i.e., each robot is responsible to detect its own faults. On the other hand, in

exogenous fault detection techniques, several robots have to be inspected simultaneously, and therefore an error that might arise in a robot's components could be detected by another robot via taking into consideration the available information of its neighborhood in the swarm.

Part II

Self-Organized Patterns in Aggregating robots swarm

Chapter 4

Material and Methods

4.1 Overview

In this chapter, we present the materials and the methods that will be used through this thesis to synthesis controllers for robotics swarm. Specifically, we introduce the ARGoS simulator and its architecture. We also present the robotics platform we adopted in our studies. Finally, we highlight the biological part from which we take inspiration for the proposed methods.

4.2 Simulation platform (ARGoS)

Today's robotics simulation platforms are providing realistic and fast prototyping models that could incorporate as much details of the reality as possible, and therefore make them much closer to the reality. Recently, one among the most used simulated robotics platforms that are destined to test and validate controllers for swarm robotics system is the ARGoS simulator [153].

ARGoS is an open source multi-robot simulator that was first developed within the EU-funded Swarmanoid project to study tools and control strategies for heterogeneous swarms of robots. Now, it is being a widely used simulation platform in many researches and projects that are dedicated to synthesis controls for swarming behaviors. ARGoS comes with all the tools needed for the development cycle of robot control code, from design to validation on real robots. Therefore, there is no difference between coding for simulation or reality. ARGoS can simulate large scales of heterogeneous swarms of robots in real time simulation. Its architecture is designed to be flexible in such way that custom features can be easily modified or added as plug-ins or modules, and that specific computational resources can be configured to fulfill the requirement of certain experiments [154, 153]. Moreover, in ARGoS multiple physics engines can be assigned to different parts of the simulated environment, and it can be also switched transparently while migrating the simulated robots from one engine to another.

The architecture of ARGoS is depicted in Figure 4.1, and below a short description about its main modules (For more related details, refer to [153, 155]):

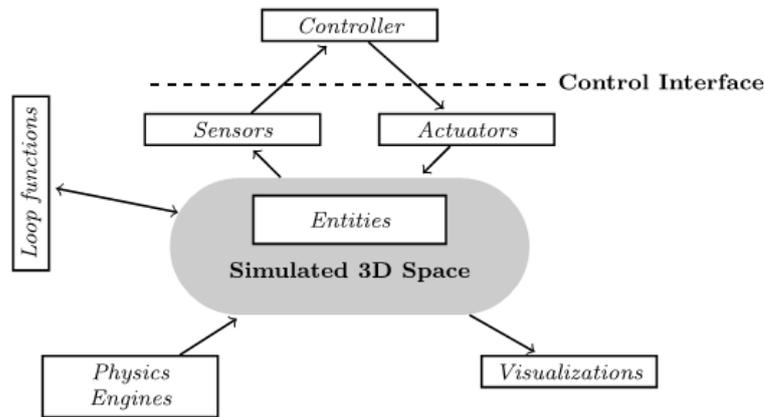


FIGURE 4.1: Schematic description of the ARGoS architecture [153].

- *Simulated 3D space*: This module is the core component of ARGoS architecture. It is a set of entities, which is considered as a central warehouse containing all the relevant information (i.e. position and orientation of robots or obstacles) about the state of the simulation.
- *Sensors and actuators*: *Sensors* are plug-ins that read the state of the *simulated 3D space*, with accessing only to specific types of entities to perform their calculations (i.e. a range sensor module needs to access information about embodied entities only). Analogously, *actuator* plug-ins update the state of the core component via writing into the components of a robot for example.
- *Physics engines*: With the *Physics engines* modules, the state of the embodied entities can be updated during an experiment by running multiple engines in parallel. This can be achieved by assigning each physics engine to a different part of the embodied entities.
- *Visualizations*: *visualizations* modules are rendering mechanisms that make an output representation of the state of the *simulated 3D space*. By default, ARGoS offers an interactive graphical user interface based on Qt and OpenGL. However, a high-quality rendering engine based on POV-Ray can also be used.
- *Controllers*: controller is a plug-in that interact with the core component through sensors and actuators. It is an implementation of the individual behaviour of an entity (i.e. a robot). Currently, it can be implemented in ARGoS using C++ or lua programming languages.
- *Loop functions*: *Loop functions* are functions hooks that are defined by user, and which can be placed in precise points in the simulation loop. For example, they can be defined at initialization time, or before and after the execution of the update phase. Through this feature and at each simulation step, the physics engine and its state can be queried and modified, while data could be also collected for further visualization and analysis.

Furthermore, ARGoS has a built-in models for several well known robots such as foot-bot, e-puck, kilobot and flybots, etc. To run an ARGoS based simulation experiment, two main components must be provided: a set of controllers and a configuration file. the controllers are user codes that include implementations of the individual behaviour of the robots, and optionally, specific functions to be executed in different parts of ARGoS to interact with the running experiment. Currently, they can be implemented using C++ or lua programming languages. The configuration file is an XML file in which a description of the structure of the simulated environment is provided, it contains all the required information to set up simulated entities such as the arena, the robots, the physics engines, etc.

Listing A.1 in Appendix A depicts an example of an ARGoS xml configuration file, in which each sub-component of the ARGoS architecture is configured. In this example, one foot-bot robot is arbitrary placed in an arena surrounded by four walls. The activated sensors/actuators of the foot-bot robot is highlighted in the corresponding sensors/actuators nodes. The physics engine of the different part of the entities configured in the xml file is set to *dynamics2d*. Finally, the visualization setup of the simulation is set to *open-gl*, with the possibility to switch to different visualization views using the three cameras that are configured within the visualization node. The corresponding simulation at run-time is illustrated in Figure 4.2.

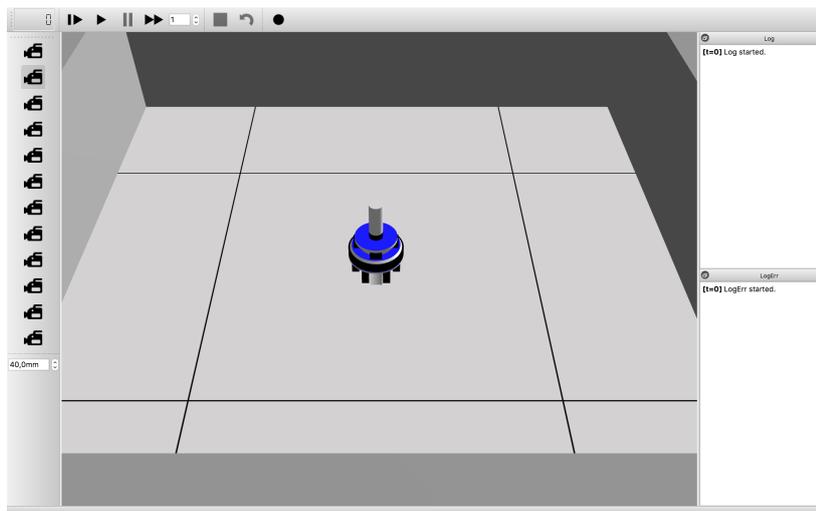


FIGURE 4.2: A snapshot during a run-time based ARGoS simulation for Listing A.1. A foot-bot robot is placed in an area of $9m^2$ of surface surrounded by 4 walls.

4.3 Robotic Platform

Throughout this thesis, we demonstrate and validate the results of our proposed models with simulated robots rather than real ones. While implementing models on real robots is an ultimate requirement to validate any proposed model. However, many advantages to work with simulated robots could be beneficial. For example,

we have to simulate the robot in various scenarios, and test its multiple versions of source code without damaging the real one. This definitively offers a low cost investigation and a shorter delivery times while compared to using real robots. Also, special specifications might be easily verified whether or not they are respected.

In order to implement and syntheses robotics controllers for our proposed models, a simulated version of the foot-bot robot (see Section 4.3.1) is mainly used in almost all the studies carried out in this thesis. Furthermore, a simulated version of the e-puck robot (see Section 4.3.2) is additionally used to investigate a circle formation problematic as an extra case study to show the possibility of re-implementing the models in different two-wheels differential robotic platforms. Both foot-bot and e-puck robots are widely involved in different collective swarming studies, and they are well implemented in the ARGoS simulator.

4.3.1 Foot-bot

The foot-bot (see Figure 4.3) is a two wheels differential mobile robot of about 17 cm of diameter and 29 cm of height, designed and built within the context of the SWARM-BOTS project. The foot-bot can move using a combination of wheels and tracks (called treels), and it comes with various sensors and actuators that allow interaction with the surrounding environment. Thereafter, a list of the most relevant ones used in swarm robotics studies:

- Twelve (12) RGB LEDs composing a ring that surrounds the robot body, and through witch colored patterns can be displayed to the other robots.
- An omni-directional camera that can be used to perceive colored objects displayed by other robots (up to a distance of approximately 50 cm).
- Four (4) ground sensors placed under the chassis can be used to perceive markers or holes on the ground.
- Twenty four (24) IR sensors for obstacle and proximity detection.
- A range-and-bearing communication device, called RAB, for exchanging messages between robots within a limited range.
- A gripper connector for allowing the foot-bot to perform physical connections such as gripping abjects or robot to robot connections.

4.3.2 e-puck

The e-puck (see Figure 4.4) is a wheeled cylindrical robot of approximately 7.4 cm of diameter and 5.5 cm of height. It was eventually designed by Mondada et al. [156] for educational researches purpose. Alike the foot-bot, the e-puck is equipped with a variety of sensors, and whose mobility is ensured by a differential drive system. The

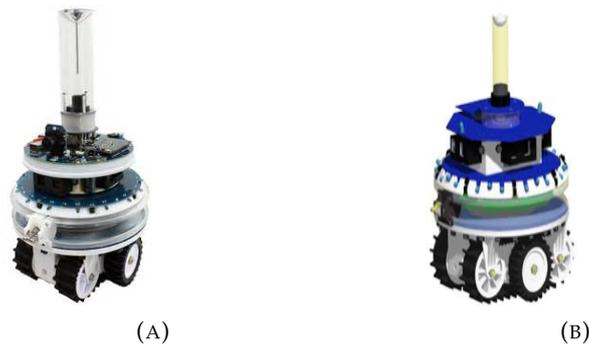


FIGURE 4.3: (A) The real foot-bot robot. (B) a CAD model for the foot-bot robot.

e-puck has nearly the same sensors/actuators of the foot-bot and below a list of the main ones used in most of the swarm robotics studies:

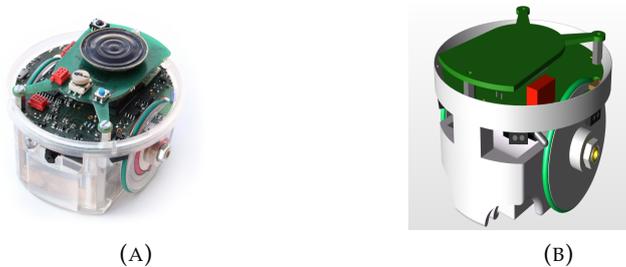


FIGURE 4.4: (A) The real e-puck robot. (B) a CAD model for the e-puck robot.

- Four (4) RGB LEDs evenly disposed around the e-puck body and which can be lightened up independently.
- A bluetooth antenna to allow the robot to establish wireless radio communication with up to any eight (8) other bluetooth devices at the same time.
- Eight (8) Infrared emitters disposed approximately around the e-puck body, and which might be used to detect obstacles and receive encoded messages from other robots.
- An extension module for range and bearing communications, which is used to provide local communication capabilities to the robot [157] .
- A frontal RGB camera that can be used for visual perception.

4.4 On-board Sensing/Actuating System

In this section, we give more details about the on-board sensing/actuating systems used throughout this thesis.

4.4.1 Infrared range and bearing sensors

Both the foot-bot and the e-puck robots are equipped with a board that allows them to communicate locally without the need of any external reference (For more specifications about this device readers can refer to Roberts et al. [158] for the foot-bot robot and to Gutiérrez et al. [159] for the e-puck robot). With this communication system, a robot can communicate with its neighbors, measuring at the same time both the range and the bearing (orientation) of the sender (See Figure 4.5). The range and the bearing communication board is composed of 12 infrared sensors/actuators that allow sending and receiving messages within a communication range of $6m$ maximum and in 12 different directions. This communication range with which infrared signals might be sensed are adjustable in real time. The particularity of this infrared communication module is that the same message can be either sent in all directions, or specifically in one direction through setting witch sensors/actuator pair is used to send the message.

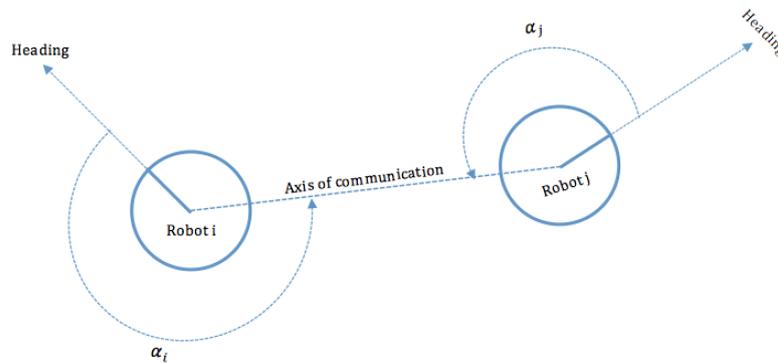


FIGURE 4.5: The relative position measurements within the communication system.

In ARGoS, the user can configure a range and bearing communication device as a module added to the list of sensors/actuators node in the xml configuration file. If any specific additional parameters are configured to this device, the behavior of the sensor will be perfect meaning that any message can be received within the configured communication range, and that the relative position (distance and bearing values) of the emitter can be precisely measured. However, several parameters are available to simulate more accurately the range and communication device as it is in the real world. For example, the user can set the *max_packets* parameter to limit the number of messages allowed to be received in one control time-step cycle. The user can also add Gaussian noise of the form $\mathcal{N}(0, \sigma^2)$ to the range and bearing measures through setting the *noise_std_dev* parameter. Loss of data during communication can be also simulated by setting the *loss_probability* parameter, meaning that packets are susceptible to be lost with the configured probability. Finally, it is also possible to reproduce in simulation the very noisy nature of the range measure of the physical robots by activating the *real_range_noise* parameter.

4.4.2 Infrared proximity sensors

The foo-bot and the e-puck robots are also equipped with IR proximity sensors that allow them detecting close objects. The particularity of these IR sensors is that it allow a robot to measure the proximity distance to an object relying on the principle of reflection. Note that a typical IR proximity sensor is composed of one IR optical transistor and one IR LED, both pointing to the same direction. The principle of reflection is that when an IR light signal emitted by an IR LED is reflected by an object, the reflected IR light is then sensed by an IR optical transistor. The intensity of the reflected IR light relies on how the object is close to the sensor and to the reflection property of the object' surface. The higher the reading of the reflected IR LED, the closer is the object in front of a sensor; the darker the painting of the object' surface, the harder is the intensity of the reflected signal.

For the foot-bot robots, there are 24 IR proximity sensors of a maximum range of 10cm, and which are equally distributed around the robot body (See Figure 4.6a). The reading of each one is composed of an angle and a value. The angle is measured in radians, and it corresponds to the relative position of the sensor with respect to the local x-axis of the robot. Whereas, the reading value part is in the range of [0,1], and it corresponds to the intensity of the reflected IR light. If no obstacle is detected the reading value is 0, while if obstacle is detected the reading will be greater than 0. As the robot gets closer to the obstacle the value will increase.

The e-puck robot has 8 IR proximity sensors that are distributed on its body (See Figure4.6b), and which have a maximum range around 7.5cm. The reading values of the sensors are ranged from 0 to 4095, while higher value indicates how close the object is in front of a sensor. Alike the foot-bot robot, the reading also contains angles orientations of the location of a sensor with regards to the x-ais of the robot.

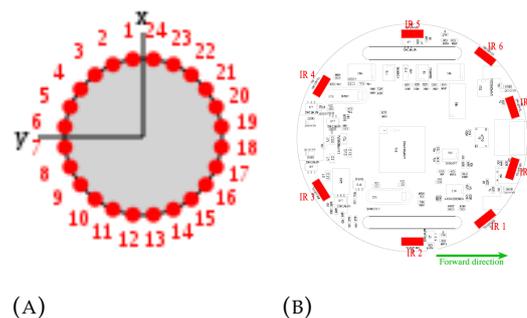


FIGURE 4.6: Position of IR proximity Sensors in (A) the foot-bot robot and (B)the e-puck robot.

In ARGoS, IR proximity sensors can be configured also as a node in the xml configuration file. Throughout this thesis, a default configuration of the sensors (See Fig.4.7) is used to synthesis obstacles avoidance controller for robots. This will give an ideal functioning to the proximity sensors. Although, this is not the case in real scenarios, we believe that using such a typical configuration for the proximity sensors is sufficient, since our objective is not to study the impact of IR proximity sensors on the

proposed models developed in this thesis.

29 `<footbot_proximity implementation="default" show_rays="true"/>`

FIGURE 4.7: A default ARGoS based configuration of IR proximity sensors.

4.4.3 Wheels Actuator

The foot-bot and the e-puck robots are considered as two wheels differential drive mobile robots, where each wheel is driven independently. Notice that the foot-bot robot moves using a treels' system composed of a set of wheels and tracks. However, it is treated like normal wheels. The forward motion of a differential drive mobile robot is achieved when driving both wheels at the same rate. In model-based control theory, the kinematics of a such kind of robots in the global coordinate frame (See Figure 4.8) is governed by Equation 4.1.

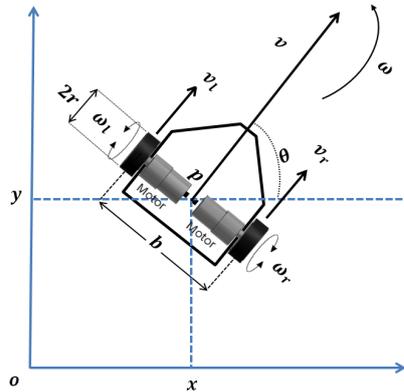


FIGURE 4.8: Kinematics of a two wheel differential drive mobile robot.

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (4.1)$$

where $p_i = [x \ y \ \theta]^T$ denotes the robot's position in the global frame, and $[v \ \omega]^T$ represents respectively the linear and angular velocities of the robot. These velocities can be written in function of the angular speed of the right and left wheels as follow [160]:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{b} & -\frac{r}{b} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix}, \quad (4.2)$$

where r is the wheel's radius and b is the axial distance between wheels. Since the forward velocities of the robot's wheels can be computed in function of their angular velocities as $v_r = r\omega_r$ and $v_l = r\omega_l$, then Equation 4.2 can be rewritten as follow:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix}, \quad (4.3)$$

and therefore, $\begin{bmatrix} v_r & v_l \end{bmatrix}^T$ can be deduced from Equation 4.3 as follow:

$$\begin{bmatrix} v_r \\ v_l \end{bmatrix} = \begin{bmatrix} 1 & \frac{b}{2} \\ 1 & -\frac{b}{2} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (4.4)$$

As consequence, the kinematics of a differential drive mobile robot in the robot coordinate frame is given by the equation follow:

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix}, \quad (4.5)$$

From Equation 4.5, the motion of the robot is controlled by giving both v_l and v_r as the control inputs. These velocities are generated by motors that are associated with the wheels. once these velocities are given, the robot motion will behave as follow: when the right wheel is driven at a higher forward velocity than the left wheel, the robot will turn left; in the opposite case the robot will turn right. While the wheels are driven in the opposite direction at the same forward velocity, the robot will then turn on the spot.

In ARGoS, this can be set by using the `set_velocity(v_l, v_r)` command. As instance, If a command is set to be `set_velocity(3,3)`, this will move the robot forward at 3cm/s. Also, other information about robot motion can be gotten such as: `distance_left` and `distance_right` which store the linear distance covered by the wheels in the last time step, `velocity_left` and `velocity_right` that store the current wheel velocity, and `axis_length` that represents the distance between the two wheels in cm. To be able to use this commands in ARGoS, the differential actuator device should be implemented in the ARGoS xml configuration file. The default configuration of this device is depicted in Figure 4.9.

24 `<differential_steering implementation="default"/>`

FIGURE 4.9: A default ARGoS configuration of the wheels actuator device.

4.5 Methods

Recall from section 3.4.2, Physics based approach is a bio-inspired method that takes inspiration from the observation of Physics. The approach was firstly introduced by Spears et al. [116] as a physicomimetics (or an artificial physics) framework, and it makes use of virtual physical forces to control the behaviour of the swarm robots

system. Taking insight from this approach, we introduce in the following subsection a new approach to study self-organized aggregating patterns within robotic swarms systems.

4.5.1 The Viscoelastic Model: A Bio-mechanic inspired Model

We propose a new artificial physics approach that takes inspiration from the biologic process behind the mechanics involved in cell morphology. In particular, we focus on the inner cell structure and the bio-mechanics known between its sub-cellular components such as cell membranes, cell cytoplasm and nucleus. More specifically, it is with these sub-components that a cell can respond to mechanical forces allowing it to move and change shapes.

Overview of the inner cell Bio-mechanics

Typically, the interior content of a cell is protected by a wall called the cell membrane. This barrier, in addition to the role that plays in separating the interior content of the cell from its surroundings, it provides also a mechanism to control substances that are entering and leaving the cell. The internal content of a cell is composed of a cytoplasm and a nucleus. The cell cytoplasm is a complex structure that can be described as a thick gel that fills the cell and envelopes the nucleus. Within it, a lot of cellular activities will take place, including several metabolic processes that are out of the context of this thesis. The cell cytoplasm is more than a fluid, it can be broken down to more than one organelle. We cite here the cytoskelton, which is considered as a polymer gel made of cross-linked actin filaments [161]. The cytoskelton organelle is involved in the cell shaping and the organization of the cell's parts, it also provides a frame for cell movement and cell division. Finally, the nucleus membrane is the brain of the cell that directs all sort of functions processed in the cell, and which contains DNA for replication and differentiation.

Despite the highly biological structure of a typical cell and those biological processes involved in it, we specifically focus here on the bio-mechanical properties of the cell cytoskeleton component. Bio-mechanical properties of a cell cytoskeleton, like elasticity and viscosity, are critical to the validity of any proposed model. Jamali, Azimi, and Mofrad [162] suggested a mathematical model to represent such properties basing on using viscoelastic interactions between the cytoskeleton membranes (See Figure 4.10). These viscoelastic interactions were modeled using *voigt* sub-units, in which springs are linear approximations to the elasticity of the inner cell and dampers are used to approximate the viscosity of the cytoskeleton.

The swarm robots viscoelastic interaction model

We take inspiration from the model of Jamali, Azimi, and Mofrad [162] and we adopted virtual viscoelastic links to model the interactions between the robots of the swarm. The idea here is not to re-generate the bio-mechanical processes of inner cells.

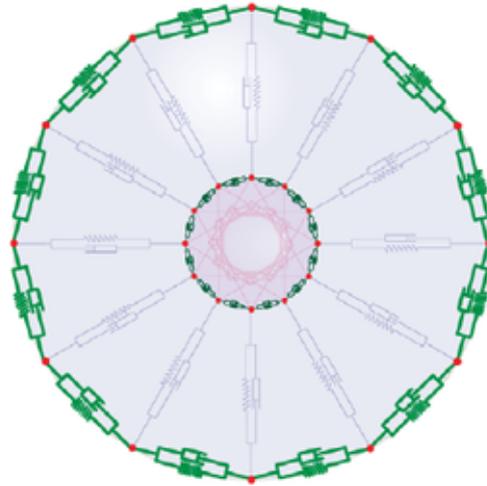


FIGURE 4.10: The sub-cellular Viscoelastic model of Jamali, Azimi, and Mofrad [162].

Figure 4.11 schematizes a model setup for the interactions between the robots of the swarm. Each robot is then affected by the virtual viscoelastic forces exerted by its neighbors. Furthermore, since the robots are placed in an inbound area surrounded by four walls in addition to some of our studies are carried while obstacles are placed in the area, an obstacle avoidance model should be then incorporated. For that we apply a repulsive control vector modelled as a potential field that is generated around each robot. The field had a strong influence when a robot was close to the potential field and a decreasing effect as the robot moved further away.

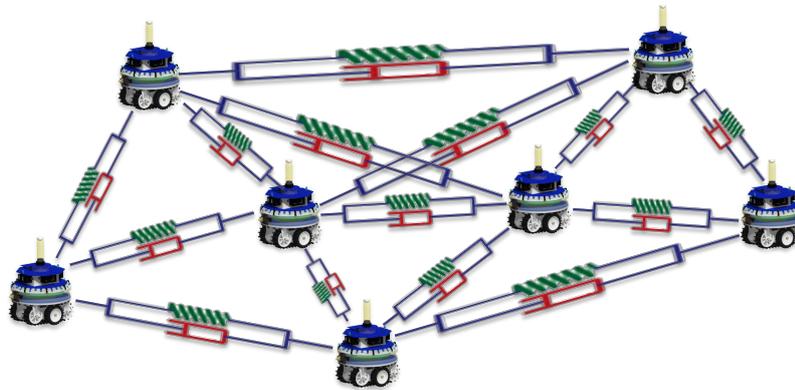


FIGURE 4.11: The swarm robots Viscoelastic interaction model.

With the proposed model, we investigated swarming behavior studies within robotics swarm (See Figure 4.12 for a schematic illustration of the studies handled using the viscoelastic proposed approach). In particular, we studied aggregating patterns by illustrating how a swarm robot system is able to achieve basic geometric formations and is capable to emerge self-organized aggregating patterns. To achieve self-organized aggregating patterns, we investigated the effect of two topological neighborhood approaches in the emergence of aggregating patterns. the first approach relies on a K-nearest neighboring method (KNN), and the second one relies

on a distance-weighted K Nearest Neighboring method (DW-KNN). In the KNN proposed aggregation approach, the robots interact with their K-nearest neighbors meaning that each robot pays attention to only its K closest teammates. In this case the distances towards neighbours are the key factor in the aggregating patterns task. However in the DW-KNN aggregation approach, the distances toward neighbours are weighted using a density estimation approach.

While swarm robotics systems are also susceptible to faults or external interferences that may lead in failing to accomplish the desired tasks successfully, we also investigated fault detection studies to enhance the proposed model.

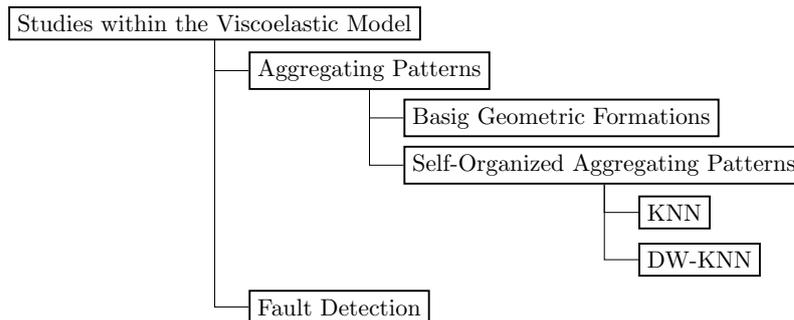


FIGURE 4.12: Schematic illustration of the studies to be carried out using the Viscoelastic model.

The general control model of the self-organized aggregation studies within robotics swarm is illustrated in Figure 4.13. The viscoelastic model is the crucial model in all the studies carried out in this thesis. It is achieved using the RAB sensors. The repulsive model accounts only for obstacle avoidance, and it is achieved using IR proximity sensors. this two sub-control models are used to actuate the robot' wheels. the details behind the implementation of the overall control model will be illustrated through out the upcoming chapters of this thesis.

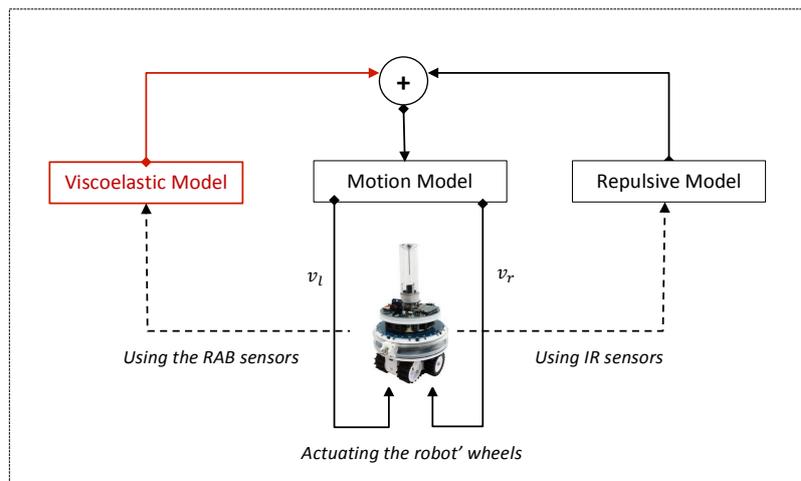


FIGURE 4.13: Schematic illustration of the self-organized aggregation model.

4.6 Summary

In this chapter, the materials and the overall methods that will be used in the remaining chapters of this thesis where reviewed in details. These materials and methods are essential for synthesizing controllers to study collective behavior within robotics swarm. We specifically introduced the ARGoS simulator and its main architecture, as well as the robotics platform (the foot-bot and the e-puck robots) to be adopted in the upcoming studies. Moreover, we discussed all the required materials in terms of the robot on-board sensing and actuating systems that will be particularly used to implement the proposed controllers. Additionally, we presented, basing on the biomechanical properties of the inner-cell, a viscoelastic interaction control model for the upcoming self-organized aggregating studies to be investigated in remaining content of this document.

Chapter 5

Basic Geometric Formations

5.1 Overview

In swarm robotics, many works have been addressed designing formation control models that are capable to sustain explicit connections among the robots while they are performing particular tasks [126, 130, 129, 85, 163]. However, most of these works represent robots as simple point models and even they use robot models, they generally apply fixed spring and damper coefficients to virtually connect the robots of the swarm.

In this chapter, we discuss how we applied our viscoelastic model to design a simple formation control model for robotics swarm system. We mainly report the basic geometric formations that can be achieved basing on intra virtual viscoelastic connectivity between the neighbors. The model is fully decentralized and it includes so much detail about the robot model, as well as it uses dynamic parameters for the spring and the damper coefficients. The basic geometric configurations, such as triangles, squares, pentagons, and circles, can be dynamically formed relying only on the relative distances and orientations estimations of the neighbors. These configurations are achieved and stabilized once the virtual viscoelastic forces that are exerted on the robots become equilibrated.

5.2 Methods

5.2.1 The Task & the Experimental Setup

The shape formation task presented in this chapter is carried out within robots that didn't have any knowledge about the dimensions of the arena. The robots have to self-organize themselves into basic geometric formations such as equilateral triangles, squares, octagons, or circles. Robots are supposed to be simple with local information, and each robot is able to sense its nearby mates within its field vision.

The experiments take place in an inbound space surrounded by four (04) walls. Initially, N robots are arbitrary dispersed in the arena and heading random orientations. Once the experiment starts, robots exert virtual viscoelastic forces among themselves for the accomplishment of the task they are looking for. In order to collectively achieve the desired task, the robots can share knowledge using IR range and

bearing communication device presented in Subsection 4.4.1. While the robots are moving, they also avoid the four walls that surround the arena using IR proximity sensors presented in Subsection 4.4.2.

Two different experimental studies have been investigated to self-organize robots swarm into the desired geometric formations. We basically used foot-bot robots to study regular shapes formation and later we address the problem of circle formation using e-puck robots.

5.2.2 Robot' Controller

As discussed in Subsection 4.5.1, the robots basically use our self-organized based model to achieve all the tasks carried out in this thesis. Specifically in this chapter, we focus on the basic behavior that emerges from the execution of the model. The model can drive the robots to perform regular geometric configurations, which can be helpful in deploying a large scale of robots swarm.

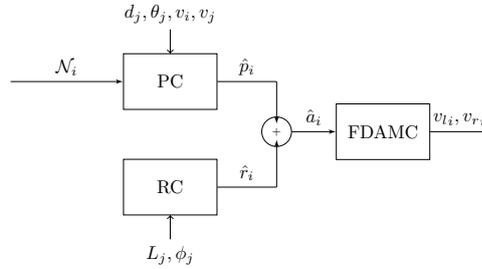


FIGURE 5.1: The Overall Control Model

As stated in Section 4.5, our self-organized based model follows a design method that is based on the artificial physics framework of Spears et al. [117]. According to this method, robots of the swarm exert virtual forces on each other. Figure 5.1 illustrates the overall control model implemented in the foot-bots. Mainly, we define a virtual force \hat{a}_i that governs the movement of each robot. This force is used to actuate the robot's wheels using a new proposed motion control (MC) called *Forward Dependent Angular Motion Control*. The virtual force, \hat{a}_i , results from a summation of two virtual forces computed using two sub-controllers as follow:

$$\hat{a}_i = \hat{p}_i + \hat{r}_i. \quad (5.1)$$

where \hat{p}_i is a proximity control (PC) that encodes viscoelastic rules, and \hat{r}_i is the repulsive control (RC) that encodes repulsive potential rules. The details behind the PC, RC, and MC sub-controllers is described in the subsections below:

Proximity Control (PC)

The proximity control is responsible for computing the virtual viscoelastic force $\hat{p}_i = [x_{\hat{p}_i} \ y_{\hat{p}_i}]^T$ exerted on each robot. Figure 5.2 illustrate a model setup between

two foot-bots interacting with a viscoelastic model, the force that results from this interaction is computed as follow:

$$\hat{p}_i = \sum_{j \in \mathcal{N}_i} \hat{p}_{ij}, \quad (5.2)$$

$$\hat{p}_{ij} = k_{ij}^s (d_{ij} - d_0) + k_{ij}^d (v_i - v_j), \quad (5.3)$$

where \mathcal{N}_i is the robot' neighbors, k_{ij}^s is the spring constant, d_{ij} is the displacement vector that represents the current length of the spring between two interacting robots, d_0 is the equilibrium length of the spring, k_{ij}^d is the damping coefficient, and v_i, v_j are respectively the forward velocities of the focal robot i and its nearby mate j .

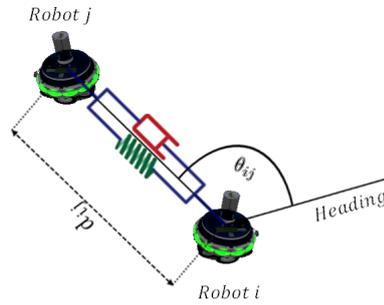


FIGURE 5.2: A model of the *viscoelastic* connection setup between two robots.

Most of the related works use fixed constants for the spring and the damping parameters. In our model and in order to reach always a stable situation and reduce the oscillation caused by such systems, the spring coefficient k_{ij}^s and the damping coefficient k_{ij}^d are computed according to the equations below:

$$k_{ij}^s = \frac{k_s}{\sqrt{d_{ij}}}, \quad (5.4)$$

$$k_{ij}^d = k_d \sqrt{k_{ij}^s}, \quad (5.5)$$

where k_s and k_d are gain constants. The output of the proximity control is the viscoelastic force vector $\hat{p}_i = \begin{bmatrix} x_{\hat{p}_i} & y_{\hat{p}_i} \end{bmatrix}^T$.

Repulsive Control (RC)

the repulsive control accounts for computing the virtual repulsive force $\hat{r}_i = \begin{bmatrix} x_{\hat{r}_i} & y_{\hat{r}_i} \end{bmatrix}^T$ that push away the robot from other robots or fixed obstacles. In our model, obstacle and collision avoidance are handled using a repulsive potential force [164]. The repulsive potential force acts only in a zone nearby the obstacle, and it has a strong influence on the robot when this one is very close to the obstacle and has a degrading influence when the robot is far away. While a robot is near to more than one obstacle, the robot is pushed away by the total repulsive forces \hat{p}_i results from the

sum of the repulsive forces \hat{r}_{ij} exerted by the set of obstacles \mathcal{O}_i , i.e.,

$$\hat{r}_i = \sum_{j \in \mathcal{O}_i} \hat{r}_{ij}. \quad (5.6)$$

Referring to Gill and Zomaya [164], each repulsive potential force \hat{r}_{ij} is computed using the negative gradient of the potential repulsive energy U_{ij} as follow:

$$U_{ij} = \begin{cases} \frac{1}{2}k_r \left(\frac{1}{L_j} - \frac{1}{L_0}\right)^2, & L_j \leq L_0 \\ 0, & \text{elsewhere} \end{cases} \quad (5.7)$$

$$\hat{r}_{ij} = -\nabla U_{ij}, \quad (5.8)$$

$$\hat{r}_{ij} = \begin{cases} k_r \left(\frac{1}{L_j} - \frac{1}{L_0}\right) \left(\frac{1}{L_j^2}\right), & L_j \leq L_0 \\ 0, & \text{elsewhere} \end{cases} \quad (5.9)$$

Where k_r is a scaling constant, L_j is the distance between the robot and the nearest edge of obstacle j , and L_0 is the obstacle influence threshold.

Forward Dependent Angular Motion Control (FDAMC)

The FDAMC is used to control the speed by which the robot on consideration will move. The FDAMC converts the virtual force control vector $\hat{a}_i = [x_{\hat{a}_i} \ y_{\hat{a}_i}]^T$ into a forward speed v_i and angular speed ω_i , then it transforms these speeds into forward speeds of both left and right wheels of the robot.

We base our FDAMC from the variable forward speed motion control (VMC) proposed in [14], by using this motion control the robots move at a variable forward speed and a variable angular speed. In VMC, the forward speed v_i and the angular speed ω_i are directly proportional to x and y components. We differ our FDAMC from this work by scaling the angle formed by the vector \hat{a}_i to get the angular speed, then the linear speed is gotten as a function of the angular speed as follows:

$$\omega_i = k_\omega \left(\frac{\alpha_i * 180}{\pi}\right), \quad (5.10)$$

$$v_i = \frac{v_{max}}{\sqrt{|\omega_i| + 1}}, \quad (5.11)$$

where $\alpha_i = \text{atan2}(y_{\hat{a}_i}, x_{\hat{a}_i})$ and k_ω is a gain constant. v_{max} is the maximum linear speed allowed for the robot.

To achieve a real robot' motion control, the angular speed should be limited within $[-\omega_{max}, \omega_{max}]$, whereas Equation 5.11 guaranties that the linear speed v_i is always within the range $[0, v_{max}]$. As we are dealing with two wheels differential drive mobile robots, v_i and ω_i should be converted into signals that actuate the robot' left and right wheels. For that, we use the differential drive model illustrated in Equation 4.4 of Subsection 4.4.3

5.3 Algorithms & Results

5.3.1 Regular Shapes Formation within foot-bots

In this subsection, we study how a swarm of foot-bots achieve regular shapes formations through implementing the above overall control. We illustrate in Algorithm 5.1 how the simulation scenario has been implemented in the foot-bot robot, and we summarize the main events that occur during the simulation.

The algorithm yields the foot-bot at every time step to move toward the final calculated target goal resulted from the force \hat{a}_i . The force is computed by summing the two virtual forces, \hat{p}_i and \hat{r}_i , that are implemented respectively in the PC function and the RC function. The velocity in which the robot moves towards the final target is implemented in the FDAMC function according to Equations 5.10, 5.11, and 4.4. The swarm system becomes stabilized where the forces are equilibrated, and at this moment the shape is automatically formed. The robots continue rotating on themselves to maintain the formed shape.

Algorithm 5.1: Regular Shape Formation

```

1 Initialize:  $d_0, L_0, b, k_\omega, k_s, k_r, v_{max}$ 
2 begin
3   for every time step do
4      $\mathcal{N}_i \leftarrow \text{senseNeighbors}()$ 
5      $\hat{p}_i \leftarrow \text{PC}(\mathcal{N}_i)$ 
6      $\mathcal{O}_i \leftarrow \text{senseObstacles}()$ 
7      $\hat{r}_i \leftarrow \text{RC}(\mathcal{O}_i)$ 
8      $\hat{a}_i \leftarrow \text{AddVector}(\hat{p}_i, \hat{r}_i)$ 
9      $\text{setRobotWheelsSpeed}() \leftarrow \text{FDAMC}(\hat{a}_i)$ 

```

To evaluate the performance of the proposed solution, we have addressed several simulation experiments using up to 12 foot-bots. We simulate the experiments using the ARGoS simulator presented in the previous chapter (See Subsection 4.2).

To achieve the PC, the foot-bot uses its RAB device to send and receive the linear speed to/from nearby robots. With the RAB device, the foot-bot is able to measure the relative range and bearing (d_{ij} and ϕ_{ij}) of the j^{th} neighboring robot. The PC assumes that a robot can perceive the range and bearing of its neighboring robots within a given range D_r . For achieving the RC, the foot-bot uses its IR proximity sensors to get the distance L_j of the closet detected obstacle and its corresponding angle θ_j . The linear speed of the left and right wheels of the robot, and which are computed using the FDAMC, is then actuated through the $\text{set_velocity}(v_l, v_r)$ command.

To allow testing the dynamic behavior of the swarm system, we investigated various Argos-based simulation experiments under different values of the implicated parameters (such as d_0, L_0 and k_ω). Table 5.1 summarizes the values of the parameters and constants related to the simulation of Algorithm 5.1.

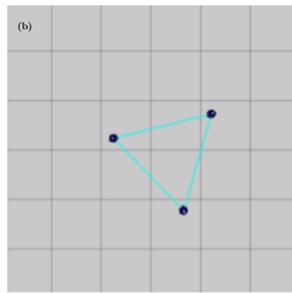
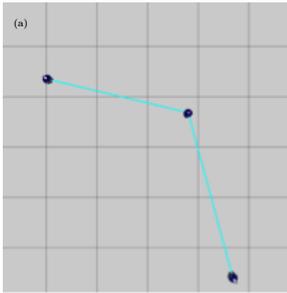
TABLE 5.1: Parameters and Constants relative to the regular shape formation model for the foot-bot robot.

| Parameter | Description | Value |
|----------------|----------------------------------|-----------------|
| b | Inter-wheels distance | 0.14 m |
| v_{max} | Maximum forward speed | 0.10 m/s |
| ω_{max} | Maximum angular speed | 180° /s |
| d_0 | Equilibrium length of the spring | 2 m |
| L_0 | Obstacle influence threshold | 0.1 m |
| k_r | Obstacle scaling constant | 1.75 force unit |
| k_d | Damper gain constant | 1.25 force unit |
| k_ω | Angular speed gain | 1.5° /s |
| k_s | Spring gain constant | 1.9 force unit |
| D_r | Maximum perception range | 5 m |

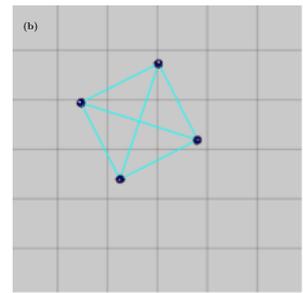
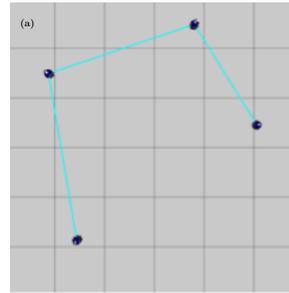
The illustrations in Figure 5.3 show various scenarios with 3 to 12 foot-bots that are randomly spread in two-dimensional space, and which finally tend, based on the number of detected neighbors, to dynamically form basic geometric regular polygons. As instance, an equilateral triangle is formed by three swarm robots (See Figure 5.3a), a square is shaped by four swarm robots (See Figure 5.3b), a pentagon and a hexagon configurations are achieved simultaneously by 5 robots (See Figure 5.3c) and 6 robots (See Figure 5.3d). Whereas a heptagon and an octagon shapes are constructed by 7 robots (See Figure 5.3e) and 8 robots (See Figure 5.3f). Our model can also achieve other configurations, where a robot can be positioned at the center of the regular geometric shape being formed. Situations such that are obtained basing on the balance of the virtual forces mainly influenced by the actual position of the robots. Some configurations like that are shown in Figure 5.3g, Figure 5.3h and Figure 5.3i.

The basic geometric regular shapes shown in these figures are strongly related to the parameters used in the simulation. For example the initial length, d_0 , imposes a unique distances between robots and hence it controls the size of the formed shape. The FDAMC angular speed gain, k_ω , controls the angular speed of the robots and this impose a direct influence on the linear speed of the right and the left wheels of the robots. This may affect the entire stabilization of the formed shape. To test the range of values of k_ω in which the swarm system performs in a stable manner, we evaluated the behavior of five simulated foot-bot robots (rb_0, rb_1, rb_2, rb_3 and rb_4) during Pentagon formation. For each robot and for each value of k_ω , we averaged the virtual force \hat{a}_i and the linear speed v_i obtained by running the simulation two hundred (200) time steps from the experiment that the Pentagon formation is stabilized. The values of k_ω investigated in the study are within the range of [0.1, 3.0].

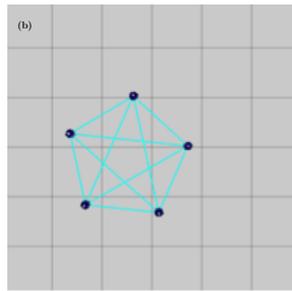
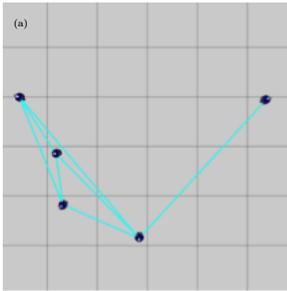
Figure 5.4a and Figure 5.4b show us that the most k_ω values in which the pentagon formation is very stable are those within the range [0.7,1.8], where all the robots of the swarm have almost the same average of \hat{a}_i and v_i . We call this range the success *ful_range* and any value out of this range lets the swarm system in an oscillatory situation. The best range that belongs to the success *ful_range* might be noticed in the



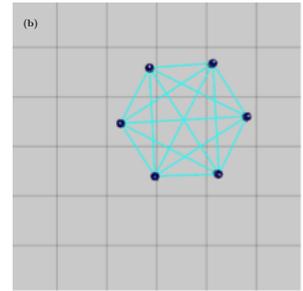
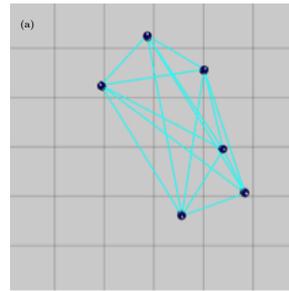
(A) 3 foot-bots forming an equilateral triangle.



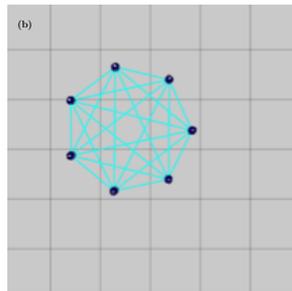
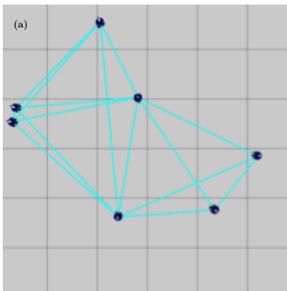
(B) 4 foot-bots forming a square.



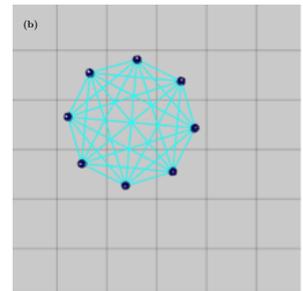
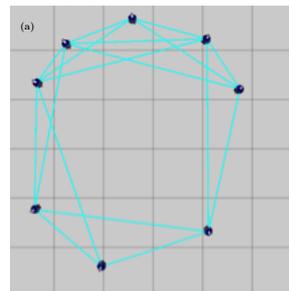
(C) 5 foot-bots forming a regular pentagon.



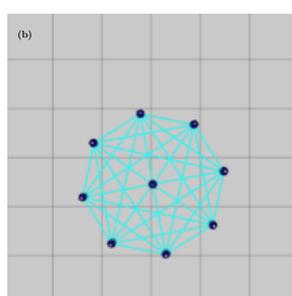
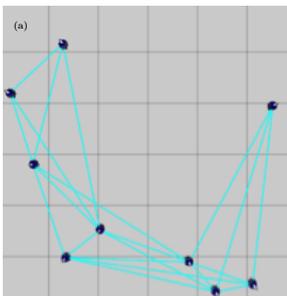
(D) 6 foot-bots forming a regular hexagon.



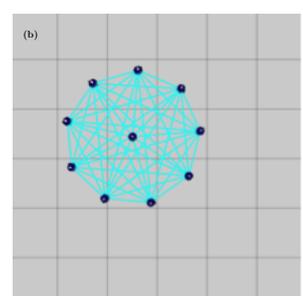
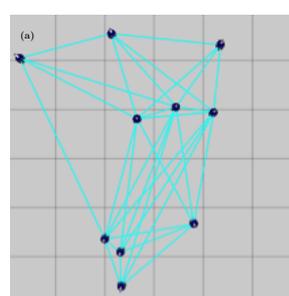
(E) 7 foot-bots forming a regular heptagon.



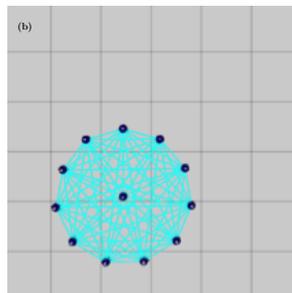
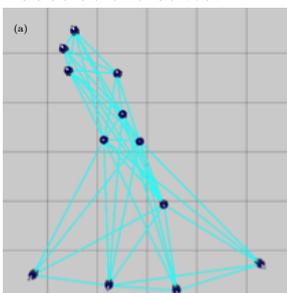
(F) 8 foot-bots forming a regular octagon.



(G) 9 foot-bots forming a regular octagon with a robot at the center.



(H) 10 foot-bots forming a regular nonagon with a robot at the center.



(I) 12 foot-bots forming a regular hendecagon with a robot at the center.

FIGURE 5.3: N foot-bots achieving different final basic geometric formations (b) when starting from initial positions (a) using our viscoelastic based model.

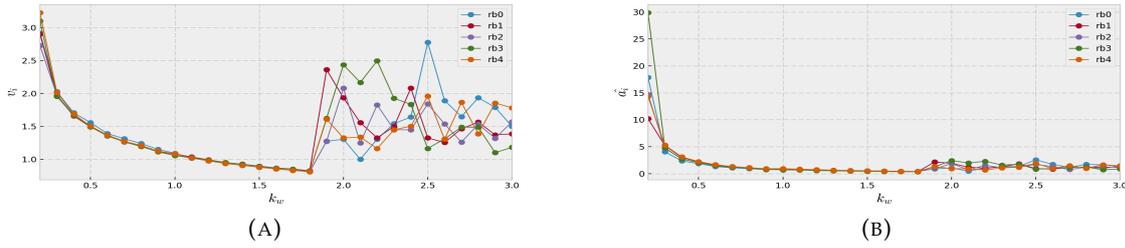


FIGURE 5.4: Rate of change of (A) v_i and (B) \hat{a}_i with respect to k_w during the last 200 time steps of a pentagon formation.

two figures as the range in which the robots have practically the minimum averages, we call this range the *best_range* and it is within the values of [1.2,1.8]. Any value (k_w) belongs to the *best_range* leads to better computing the linear speed v_i and hence a well calculating of the virtual force \hat{a}_i . For $k_w = 1.5$, Figure 5.5a and Figure 5.5b analyze the rate of change of the virtual force, \hat{a}_i , and the robot linear speed, v_i , during a full Pentagon formation. The figures show that both v_i and \hat{a}_i are stabilized for all the robots from the time step ($t = 400$), at this moment \hat{a}_i tend to zero and v_i have a minimum stable value.

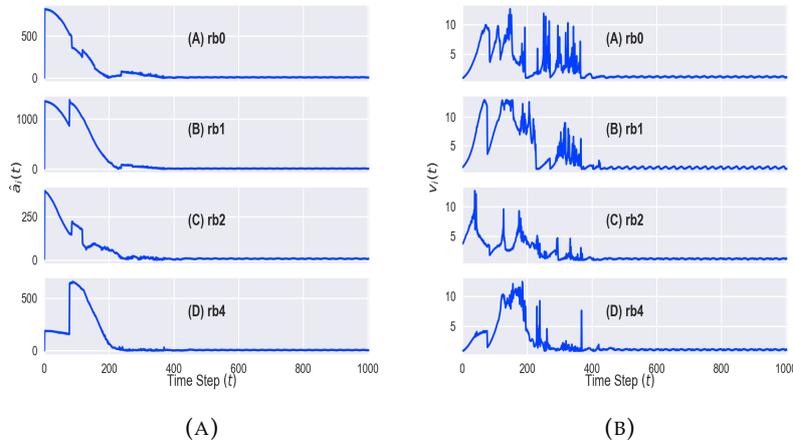


FIGURE 5.5: Rate of change of (A) $\hat{a}_i(t)$ and (B) $v_i(t)$ for 5 foot-bots ($rb0$, $rb1$, $rb2$, $rb3$ and $rb4$) performing a pentagon formation task. the x-axis of the plots represents the evolution of time step t .

5.3.2 Circle Formation within e-pucks

In this study, we adapt the formation control model proposed in the previous subsection in a way to dynamically self-organize a swarm of e-pucks into an uniform circle formation. The circle formed by robots is governed by two strict rules (a) almost all robots should be placed on the circle boundary (b) a robot might be positioned at the center of the circle. The model suggested in this study is decentralized and scalable by

which a circle can be dynamically formed using mainly a combination of virtual viscoelastic interactions as attractive primitives amongst neighbours and circumscribed circle theory.

Circumscribed Circle Theory

In geometry, any circle that connects all the vertices of a given polygon is called the circumscribed circle or circumcircle of a polygon. For specific polygons such as regular ones, there exists a circle that can be circumscribed about any regular polygon. Such circles are characterized by the following properties (See Figure 5.6 for a hexagone case):

- The center of the circumcircle is the same as the center of the regular polygon.
- The radius R of the circumcircle is also the radius of the polygon.
- The Side length $w = 2 * R * \sin(\pi/n)$, where n denotes the number of sides.

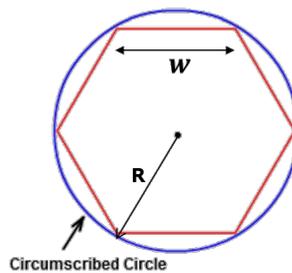


FIGURE 5.6: A circumscribed circle of a hexagon.

Circle Formation via the Virtual Viscoelastic Control Model

Alike the regular shapes formations task achieved with the foot-bots, to achieve a circle formation every e-puck of the swarm is subject to the virtual driven force control presented in Equation 5.1.

Notice that the predominant force is the viscoelastic force (\hat{p}_i) since the \hat{r}_i force accounts only for avoiding collision amongst robots, and since we choose a wake k_r and a short obstacle influence threshold.

The idea behind our control model is to model every neighbouring link (as well as every polygon side) via virtual viscoelastic link. Since the side length of any regular polygon of n sides circumscribed by a circle of radius R , can be computed as given in the previous subsection. Then the relation between radius R and the equilibrium spring length constant d_0 is computed as follow:

$$d_0 = 2 * R * \sin(\pi/n). \quad (5.12)$$

By using the combination of the circumscribed circle theory with mainly the viscoelastic control model, the e-pucks will achieve global circle formation whatever the

number of the robots being implicated. Algorithm 5.2 summarizes the main events occurred during the simulation. The algorithm is an adaptation of Algorithm 5.1 in a such way that the spring length, d_0 , of the viscoelastic force is computed at every time step using Equation 5.12.

Algorithm 5.2: Circle Formation

```

1 Initialize:  $R, L_0, b, k_\omega, k_s, k_r, v_{max}$ 
2 begin
3   for every time step do
4      $\mathcal{N}_i \leftarrow \text{senseNeighbors}()$ 
5      $d_0 \leftarrow 2 * R * \sin(\pi / |\mathcal{N}_i|)$ 
6      $\hat{p}_i \leftarrow PC(\mathcal{N}_i)$ 
7      $\mathcal{O}_i \leftarrow \text{senseObstacles}()$ 
8      $\hat{r}_i \leftarrow RC(\mathcal{O}_i)$ 
9      $\hat{a}_i \leftarrow \text{AddVector}(\hat{p}_i, \hat{r}_i)$ 
10     $\text{setRobotWheelsSpeed}() \leftarrow \text{FDAMC}(\hat{a}_i)$ 

```

As in performing regular shapes formation within foot-bots, to achieve the virtual viscoelastic based circle formation control model with e-puck we use both the RAB and the proximity sensors. The RAB is used specifically to compute the \hat{p}_i force by communicating the linear speed to the neighboring robots as well as measuring their relative range and bearing values. The proximity sensors are used to compute the \hat{r}_i force. Finally, the total force \hat{a}_i is transformed into signals to actuate the e-puck' wheels.

Several simulation experiments have been addressed under the ARGoS simulator. Table 5.2 summarizes the values of the different parameters and constants used in the simulation. The illustrations in (Fig. 5.7) show some scenarios with 5 to 8 robots that are randomly spread in arena of $10 * 10m$, and which finally tend, based on the number of detected neighbors, to dynamically form a circle within or without positioning a robot at the center of the circle.

TABLE 5.2: Parameters and constants relative to the circle formation model for the e-puck robot

| Parameter | Description | Value |
|----------------|------------------------------|-----------------|
| b | Inter-wheels distance | 0.053 m |
| v_{max} | Maximum forward speed | 0.13 m/s |
| ω_{max} | Maximum angular speed | $180^\circ/s$ |
| L_0 | Obstacle influence threshold | 0.1 m |
| k_r | Obstacle scaling constant | 1.75 force unit |
| k_d | Damper gain constant | 1.25 force unit |
| k_ω | Angular speed gain | $1.5^\circ/s$ |
| k_s | Spring gain constant | 1.9 force unit |
| D_r | Maximum perception range | 5 m |
| R | Radius of the wanted circle | 1.75 m |

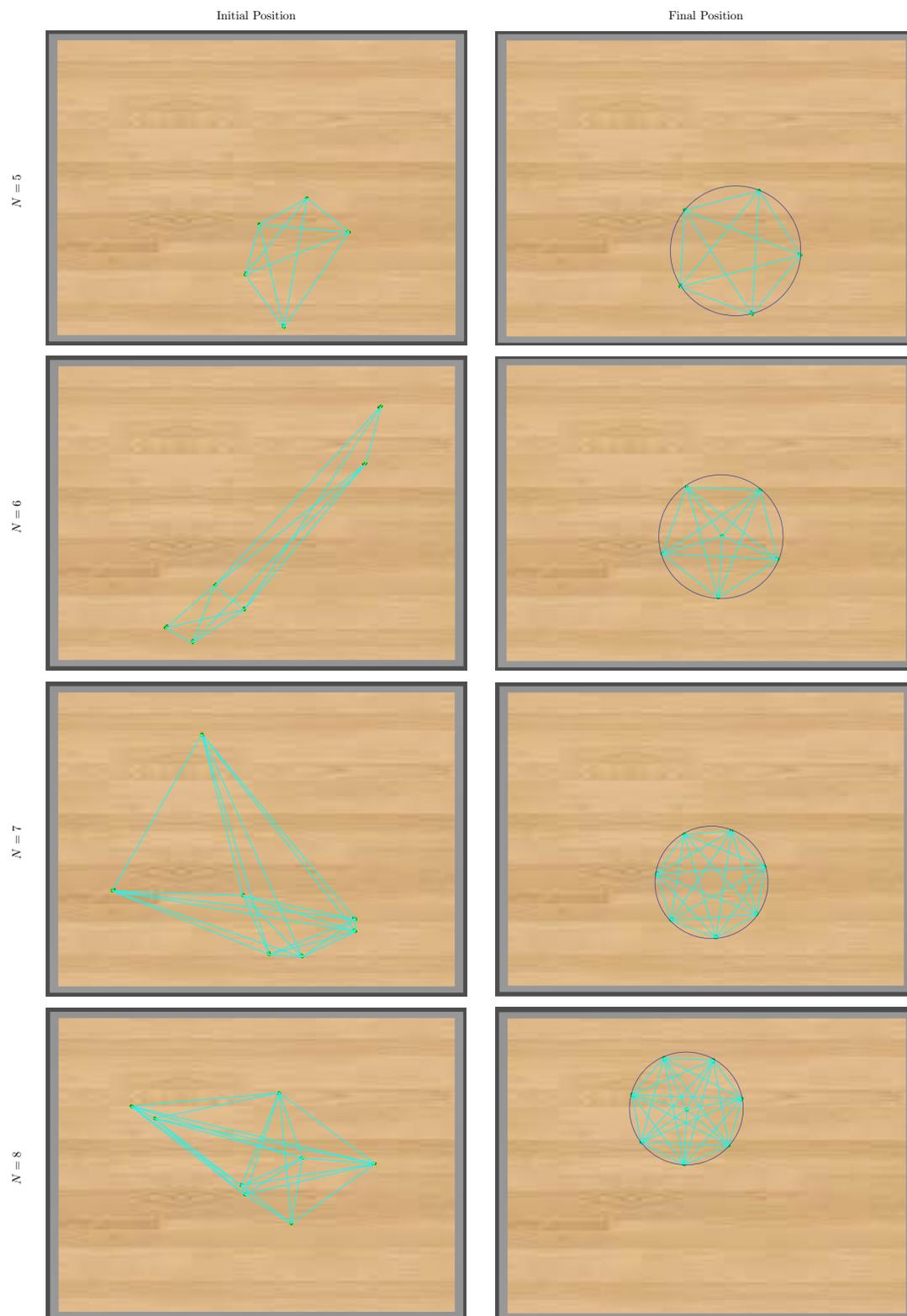


FIGURE 5.7: Four simulation scenarios of a swarm of N e-pucks forming a uniform circle via the virtual viscoelastic based control model.

5.4 Performance Analysis

In this section, we analyze the performance of the proposed control models using the following metrics.

5.4.1 Performance Metrics

Group Speed

The Group Speed GS is the magnitude of the velocity of the center of mass of the robots [165]; it is usually used to get an idea on how the speed of the entire swarm is evolved in time. The formula of this metric is as follow:

$$GS(t) = \frac{\|(\vec{C}[t] - \vec{C}[t-1])\|}{T}. \quad (5.13)$$

Here \vec{C} is the center of mass of the group of robots sampled at t and $(t-1)$, T is the sample period.

Mean Distance Error

The Mean Distance Error ($MDER$) is used mostly in multi-robots formation and flocking algorithms to get a value of how the robots are well positioned. To compute the $MDER$ [165], we take into considerations neighbors that are at a distance $\delta < (1.5 * dd)$. Then the formula of getting $MDER$ is:

$$MDER(t) = \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{|\mathcal{N}_{i,\delta}[t]|} \sum_{r_j \in \mathcal{N}_{i,\delta}[t]} (d_{ij}[t] - dd) \right) \quad (5.14)$$

Where N is the number of the robots, dd is a desired inter robot distance and in our case $dd = d_0$, $\mathcal{N}_{i,\delta}[t]$ is the neighbors of the robot that satisfy δ at sample time t , and $d_{ij}[t]$ is the distance between the robot i and the robot j .

5.4.2 Results within the Metrics

In order to evaluate the performance of the proposed models within the metrics presented in the previous subsection, we conduct a number of ARGoS based experimental simulations using a swarm of foot-bots and a swarm of e-pucks, performing respectively a regular shape formation task and a circle formation task. The simulations include studies in both absence and presence of noises in the RAB device of each robot. In absence of noises, the robots are supposed to perform in a perfect way meaning that all the sensors, in particular the RAB ones, are set to be ideal with no noises. While in presence of noises, noises are added to the true measures of the RAB sensors, and they are modeled using a Gaussian model of the form $\mathcal{N}(0, \sigma^2)$. This can be configured in the ARGoS simulator by setting the standard deviation parameter, `noise_std_dev`, to a σ value. In our experimental simulations with noisy sensors,

the $noise_std_dev$ parameter is set to 0.01. The very noisy nature of the range measure of the physical robot is also reproduced in simulation through activating the $real_range_noise$ parameter. Further, the probability of losing packets during communications between robots is taken into account by setting the probability of lost data parameter, $loss_probability$, to 0.03 meaning that 3% of packets communicated between a robot and its neighbors is set to be lost.

In Figure. 5.8a and Figure. 5.8b, we plot the results obtained from the simulation of four different regular shapes (Square, Pentagon, Hexagon, and Heptagon) achieved by N foot-bots. The figures show respectively the evolution of GS and $MDER$, in presence and absence of noises, for the regular geometric case studies on focus. As stated in the figures and regardless the presence or absence of noises in the RAB device, a stable GS and a minimum steady $MDER$ are achieved for the entire regular shapes being formed, leading to conclude that the proposed model is enough effective to be scaled to a large number of robots.

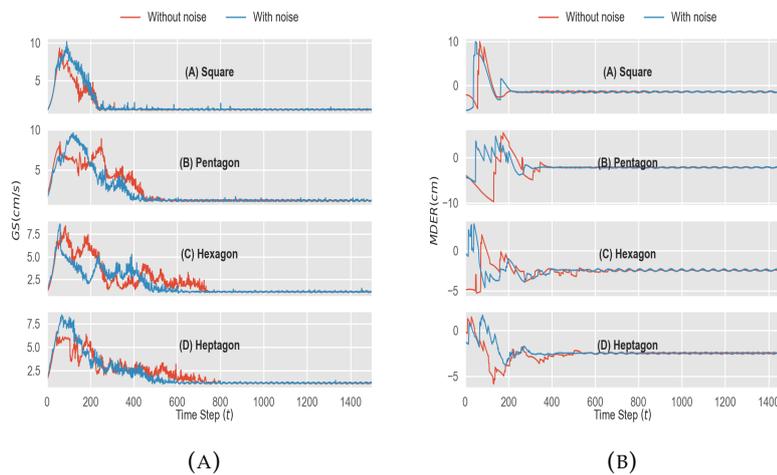


FIGURE 5.8: Evolution in time of (A) Group Speed (GS) and (B) Mean Distance Error ($MDER$) during a formation of four different regular shapes by a N foot-bot robots under absence and presence of noise in the RAB device. The curves in the plots represents the median values obtained from 5 runs by each experiment.

Figure. 5.9a and Figure. 5.9b demonstrate performance results from data collected during simulating different groups of N e-puck robots accomplishing a circle formation task. Alike the performance results obtained with the foot-bot robots, the e-puck robots are able to achieve accurate results in both presence and absence of noises in their RAB sensors. The results shown in the figures illustrate that the robots are capable to accomplish the circle formation task when the group achieve a stable GS and a minimum steady $MDER$ in almost all the simulation case studies. This shows that our proposed viscoelastic control model is appropriate to different 2-wheeled robotics platforms, which could be used to accomplish the desired task.

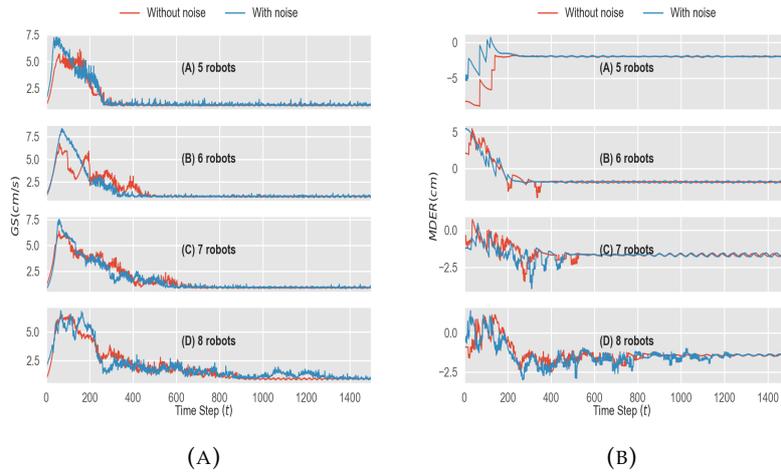


FIGURE 5.9: Evolution in time of (A) Group Speed (GS) and (B) Mean Distance Error ($MDER$) during a circle formation by N e-puck robots under absence and presence of noise in the RAB device. The curves in the plots represents the median values obtained from 5 runs by each experiment.

5.5 Conclusion

Pattern formation is one of the challenging aspects in swarm robotics self-organization problems, which currently knows an important interest in today's real world. In this chapter, we provided a simple solution to such problem via using our viscoelastic interaction model. With this model, we do not treat robots as point-masse particles like the most proposed virtual spring based formation control did, we consider each robot of the swarm as a simple two wheels differential-drive mobile robot whose geometric configuration is highlighted as in section 4.4.3.

We mainly reported a swarm robots formation control model that is based on intra virtual viscoelastic connectivity between the neighbors. The model can dynamically achieve regular geometric formations such as squares, pentagons, and octagons using only the distance and orientation estimations among the neighbors. The model is fully decentralized and scalable by which the number of the robots being implicated can dynamically form a desired regular basic geometric configuration. The model has been implemented in a simulated version of the real foot-bot robot using the ARGoS simulator. In addition, a number of analyses studies within the implemented model have been done to assess the stability of the desired regular geometric configurations, and to test the parameters involved in the simulation. We further demonstrated that the proposed viscoelastic formation model could be adaptable to achieve a circle formation task by another robotics platform called e-puck.

Based on the equilibrium of the virtual viscoelastic forces exerted on the robots, the proposed models could position the robots at equal angular ranges of the desired regular geometric shape as well as on the boundary of the desired circle, with or without positioning a robot at the center of the wanted configuration. The performance

of the formation control models presented in this chapter were evaluated using two metric of performances: the group speed GS and the mean distance error $MDER$. Performance results, in both presence and absence of noises in the RAB sensors of the robots, showed the accuracy of our overall viscoelastic interaction model.

Chapter 6

Topological Approaches in Self-organized Aggregating Patterns

6.1 Overview

Self-organized aggregating patterns that are observed in front of our eyes in flocking of large numbers of starlings might be achieved by a simple interaction rules among neighbors. Empirical analyses of such studies revealed that, a restricted number of 6 to 7 members amongst the entire ones available in the field vision of a given bird, are only associated in the interaction process. This is referred as a topological metric distance while compared to the metric distance that takes all the neighbours on the field vision.

Taking inspiration from this topological biological study, this chapter proposes two topological aggregation approaches to study self-organized aggregating patterns within robotics swarm. The chapter particularly extends our viscoelastic interaction model proposed in the previous chapter to cope with large scale of swarm robot system. We specifically provide studies in self-organized aggregating patterns while implicating our viscoelastic model in a K-Nearest Neighbors (KNN) topological approach and a Distance-Weighted K-Nearest Neighbors (DW-KNN) topological approach. Within the two topologies cases studies, we investigate a number of experiments in presence and absence of obstacles, and we further assess the performance of the two approaches within four metrics of performance, in presence and absence of different noise models in the RAB sensors.

6.2 Methods

6.2.1 The Task & the Experimental Setup

The experiment carried out in this chapter takes inspiration from the topological distance approach revealed in studies of birds flocking and fish schooling. Empirical results in these studies showed that the unpredictable, amazingly complex patterns

formed by birds emerge from a topological distance approach rather than a metric distance approach [80, 81]. This has been explained as the interaction between birds is shown to be only with the nearest six to seven neighbors rather than all of the neighbors in their field of vision. Computer simulations predict also that a significantly higher cohesion of the aggregation is achieved using a topological interaction rather than the standard metric one.

In our experiment, the task of the robots consists in forming self-organized aggregating patterns that emerge through topological neighborhood approaches without using any cues. We consider an area with or without obstacles surrounded by four walls, containing a swarm of N foot-bot robots that are initially distributed in random positions and heading arbitrary directions.

In more details, the arena where the experiments take place is a rectangular area of $(10 * 6)m^2$ surface. In the experiments, noise is added to the range and bearing measurements of the RAB device. Noise is modeled as a Gaussian distribution with zero mean and 0.01 standard deviation. The loss of packets during communication is also taken into consideration by setting the loss probability to 3%.

In order to study self-organized patterns in aggregating foot-bot robots swarm, we implicated our proposed viscoelastic model in two topological approaches, and we studied the impact of that in self-organizing aggregating patterns. In particular, we studied the implication of the viscoelastic aggregation model in the KNN and the DW-KNN topological methods. In the KNN approach, foot-bot robots are aggregated basing on the closest k neighbors, meaning that distances toward neighbors are the only factor taken into account during the aggregation process. While in the DW-KNN approach, distances toward neighbors are firstly weighted, and then robots are aggregated relying on the closest k weighted distances. Moreover, with the two proposed topological aggregation approaches we studied how K , the number of neighbors that are involved in the aggregation process, can play a role in emerging self-organized aggregating patterns.

6.2.2 Robot' Controller

The K-NN Topological Approach

In this approach, the overall self-organized aggregation control model that is implemented in each foot-bot robot is illustrated in Figure 6.1. In this control, the proximity sub-control (PC) model presented in the previous chapter (See Subsection 5.2.2), and which is modeled using virtual viscoelastic interactions, is applied among the K-NN robots, and not among all the neighbors in the field vision of the robots. To achieve this topology, robots are interacting only with their K closest neighbors, where K represents how many neighbors a robot should interact with.

Formally, let $\mathcal{N}_i(t)$ whose cardinality is denoted by N be the neighbors of a given

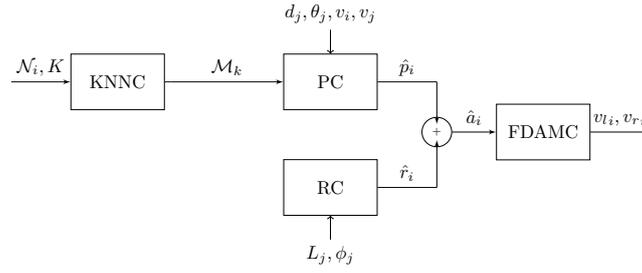


FIGURE 6.1: Schematic description of the KNN control model.

robot R_i within a range D_r at time t , and moreover let $\mathcal{M}_K(t)$ be the K closest neighbors to the robot R_i at the same time t . Then, the proximity control vector \hat{p}_i of Subsection 5.2.2 is re-computed as follows:

$$\hat{p}_i = \sum_{j \in \mathcal{M}_K(t)} \hat{p}_{ij}, \quad (6.1)$$

where \hat{p}_{ij} is the virtual viscoelastic force generated between to interacting robots, and which is computed as in Equation 5.3.

In order to identify the K -nearest neighbors, $\mathcal{M}_K(t)$, the foot-bot sorts the neighbors, $\mathcal{N}_i(t)$, by the nearest K distances, where $K \in \{1, 2, \dots, N-1\}$ refers to how many neighbors are taken into account. Note that the cardinality of the neighborhood, $|\mathcal{M}_i(t)|$ is K , and $j \in \mathcal{M}_i(t) : d_{ij} \leq d_{im}, \forall m \in \mathcal{M}_i(t)$. In the case where $K = N-1$, the mesh is the all-to-all connected network.

By relying on this topological approach, the distance toward neighbors is the only key factor taken into account in the aggregation process. Therefore, the basic regular geometrical configurations achieved in the previous chapter could be used as a base arrangement for the overall self-organized aggregating patterns achieved by this topology.

In certain swarm applications, where inter-agent distance is not the only factor on the collective behavior of the swarm, additional properties such as density could have a crucial effect. For example to drive a large number of robots from one area to another, the density of the robots could play the primary role while the inter-robot distance would play only a secondary role as a metric useful for proximity control or collision avoidance. In the next subsection, the idea of using a Distance-Weighted K Nearest Neighboring (DW-KNN) topology is proposed to study self-organized aggregating patterns as an emergent swarming behaviors within robot swarms.

The DW-KNN Topological Approach

In this subsection, a new topological distance approach is proposed to study self-organization in aggregating robot swarms. We use the density of robots in the swarm as an additional factor and define a new neighboring relationship based on the Distance-Weighted KNN approach. With the DW-KNN topology, virtual viscoelastic connections are dynamically created and destroyed among the weighted KNN rather

than the unweighted KNN. We present in Figure.6.2 an illustration scheme of the overall self-organized aggregating pattern control model within the DW-KNN topological approach.

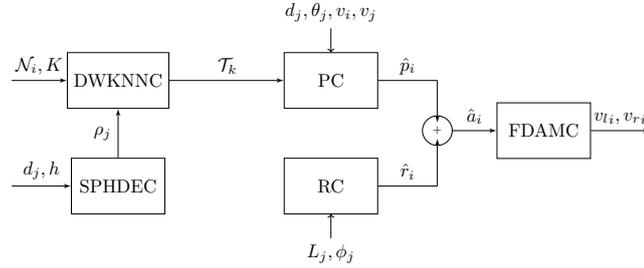


FIGURE 6.2: Schematic description of the DW-KNN control model.

To achieve the proximity control model using this approach, a topological neighborhood strategy is applied first by each robot R_i to decide which neighbors among those available are taken into account when arranging the robots. Then, a mesh of virtual viscoelastic links is built between the robots of the topological neighborhood [19, 18]. In this case, The proximity model of each robot R_i is computed as follows:

$$\hat{p}_i = \sum_{j \in \mathcal{T}_k(t)} \hat{p}_{ij} \quad (6.2)$$

where $\mathcal{T}_k(t) \in \mathcal{N}_i(t)$ is the set of topological neighbors at time t . The details about how the robot identifies this set is discussed bellow:

- First, a robot R_i computes its density ρ_i based on an SPH density estimation technique, which should be immediately communicated to its neighbors.
- Second, upon receiving the densities (ρ_j) from neighbors, a weighted-distance function w_{ij} is applied to weight the distances to the neighbors.
- Finally, based on w_{ij} , the set \mathcal{T}_k is identified by sorting the neighbors in order of the nearest K -weighted distances, where K refers again to how many neighbors are taken into account.

The SPH density estimation technique and the weighted-distance-based function we use in our study are discussed in the following subsections.

1. SPH Density Estimation Model

SPH is a mesh-free Lagrangian method, in which the state of the simulated system is represented by employing a finite set of disorder discrete particles in a way that makes both a fixed order to organize the particles and a generated mesh to represent the connectivity of the particles unnecessary [166]. SPH has been firstly introduced in computational astrophysics studies and is applied in simulating compressible flow problems [167]. One of the main features of the SPH technique is that at any given point in the simulation domain Ω , a property

of a particle i can be approximated relying on a summation of an interpolation kernel function W with h as the smoothing length [168]. A schematic representation of this system is presented below in Figure 6.3.

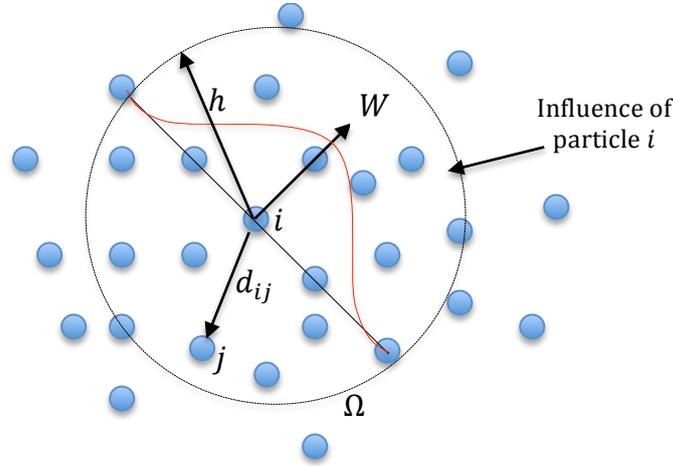


FIGURE 6.3: A schematic representation of SPH particle interactions within the influence domain governed by the kernel function.

In the SPH approach, interpolation is used to approximate physical quantities that are moving with particles; here we approximate the density as a physical quantity moving with the robots. The robot density ρ_i is evaluated as the weighted sum of distances over its proximate robots within a particular range $D_r = 2h$ [169]:

$$\rho_i = \sum_{j \in \mathcal{N}_i} W(d_{ij}, h), \quad (6.3)$$

The weight functions used in this work are the M4 cubic spline functions truncated at $2h$ [167]:

$$W(d_{ij}, h) = \sigma \begin{cases} \frac{1}{4}(2-q)^3 - (1-q)^3, & 0 \leq q < 1 \\ \frac{1}{4}(2-q)^3, & 1 \leq q < 2 \\ 0, & q \geq 2 \end{cases} \quad (6.4)$$

where $q = d_{ij}/h$ and $\sigma = 10/(7\pi h^2)$ is a normalized constant. The computed density is communicated to the neighbours of the robots.

2. Distance-Weighted K-Nearest neighbors

DW-KNN is a very popular acronym in machine learning. It is a classification method that is specifically used to assign a label to a new query based on weighting closer neighbors more heavily, according to their distances from the query [170]. Taking inspiration from this technique, and upon receiving the SPH densities from the neighbors, a robot R_i weights the distances to its neighbors

as

$$w_{ij} = \rho_j d_{ij}, \quad (6.5)$$

where ρ_j denotes the density received from the j^{th} neighbor. Further, the robot builds a DW-KNN connectivity, $\mathcal{T}_k(t)$, by sorting the neighbors in order of the nearest K -weighted distances, where $K \in \{1, 2, \dots, N-1\}$ refers to how many neighbors are taken into account. The cardinality of the neighborhood, $|\mathcal{T}_k(t)|$ is K , and $j \in \mathcal{T}_k(t) : w_{ij} \leq w_{im}, \forall m \in \mathcal{T}_k(t)$. In the case where $K = N-1$, the mesh is the all-to-all connected network.

By using the function in Equation 6.5 where both the distance and the density are applied as equally key factors, a neighbour robot R_j located far away from the robot R_i with a heavy density could have a greater impact than one located near R_i but with a weak density.

6.3 Results & Discussion

6.3.1 Self-organized Aggregation Patterns basing on the KNN approach

The main pseudo code highlighted in Algorithm 6.1 is executed at every time step t by each individual foot-bot. The overall detailed algorithm is reported in Appendix A.2. The pseudo code uses essentially the force resulted from the PC function to define the target goal to be achieved by the robot. The PC function returns the total virtual viscoelastic force exerted by the K nearby robots. The target goal might be influenced by a total repulsive potential force generated from detected obstacles or robots en collision and which is calculated in the RC function. The algorithm then computes the required velocities to be set to the wheels of the robot using the $FDAMC$ function. The swarm system becomes stabilized where the forces are equilibrated, and at this moment emerging shapes are automatically being created basing on the value of the aggregation parameter K .

Algorithm 6.1: KNN based Self-Organized Aggregating Pattern Algorithm

```

1 Initialize:  $d_0, L_0, b, k_\omega, k_s, k_d, k_r, v_{max}, K$ 
2 begin
3   for every time step do
4      $\mathcal{N}_i \leftarrow \text{senseNeighbors}()$ 
5      $\mathcal{M}_k \leftarrow \text{KNNC}(\mathcal{N}_i, K)$ 
6      $\hat{p}_i \leftarrow \text{PC}(\mathcal{M}_k)$ 
7      $\mathcal{O}_i \leftarrow \text{senseObstacles}()$ 
8      $\hat{r}_i \leftarrow \text{RC}(\mathcal{O}_i)$ 
9      $\hat{a}_i \leftarrow \text{AddVector}(\hat{p}_i, \hat{r}_i)$ 
10     $\text{setRobotWheelsSpeed}() \leftarrow \text{FDAMC}(\hat{a}_i)$ 

```

With the KNN based topological approach, we conducted a number of ARGoS based experimental simulations using a swarm of $N = \{100, 150\}$ foot-bot robots.

TABLE 6.1: Parameters and constants used in the KNN approach

| Parameter | Description | Value |
|----------------|----------------------------------|-----------------|
| b | Inter-wheels distance | 0.14 m |
| v_{max} | Maximum forward speed | 0.10 m/s |
| ω_{max} | Maximum angular speed | 180 °/s |
| d_0 | Equilibrium length of the spring | 0.3 m |
| L_0 | Obstacle influence threshold | 0.1 m |
| k_r | Obstacle scaling constant | 1.75 force unit |
| k_d | Damper gain constant | 1.25 force unit |
| k_ω | Angular speed gain | 1.5 °/s |
| k_s | Spring gain constant | 1.9 force unit |
| D_r | Maximum perception range | 1.5 m |

The area where the robots are initially randomly distributed is set to be with or without obstacles. Starting from the initial distribution of robots, a set of constants and parameters (See Table 6.1) were initiated in each experiment. Algorithm 6.1 shows its ability to emerge aggregating patterns basing on the value of K , meaning that the final emerged patterns is based on the geometrical shape to be chosen as a configuration based arrangement. The illustrations highlighted in Figure 6.4 demonstrate the variation in aggregating patterns, which emerge through giving different values to K . The figure shows different ARGoS simulations snapshots of how the algorithm yields the foot-bot robots to dynamically self-organize into various aggregating patterns throw only varying the method of how the neighbor robots should be aggregated. The snapshots are presented by three columns from left to wright; in which each column, starting from the initial positions of the N swarm robots to diversifying the values of K (from 2 to 7), illustrates the corresponding emergent final shapes achieved after running the algorithm. The right column of the figure demonstrates a situation in which obstacles can affect the bearing sensing of the robots and hence it disturbs the resulting formed shapes as a result of troubling the aggregations of the robots. The snapshots also demonstrate the ability of the K -nearest virtual viscoelastic based aggregation algorithm to self-organize into different shapes by different clusters of swarm robots.

6.3.2 Self-organized Aggregation Patterns basing on the DW-KNN approach

The pseudo code in Algorithm 6.2 is an implementation of the overall DW-KNN control model pointed in Figure 6.2. The overall detailed algorithm is reported in Appendix A.3. At every time step t and upon using the equations from the SPH density estimation model, the foot-bot is able to compute its density, which is immediately sent to its neighbors via the RAB. Since the foot-bot can exchange only 10 bytes of data, the density is scaled down by 10^3 before being sent to the neighbors.

To achieve the DW-KNN Control (DWKNNC), the robot receives the densities of the neighbors and computes its corresponding weighted distances; then, based on the

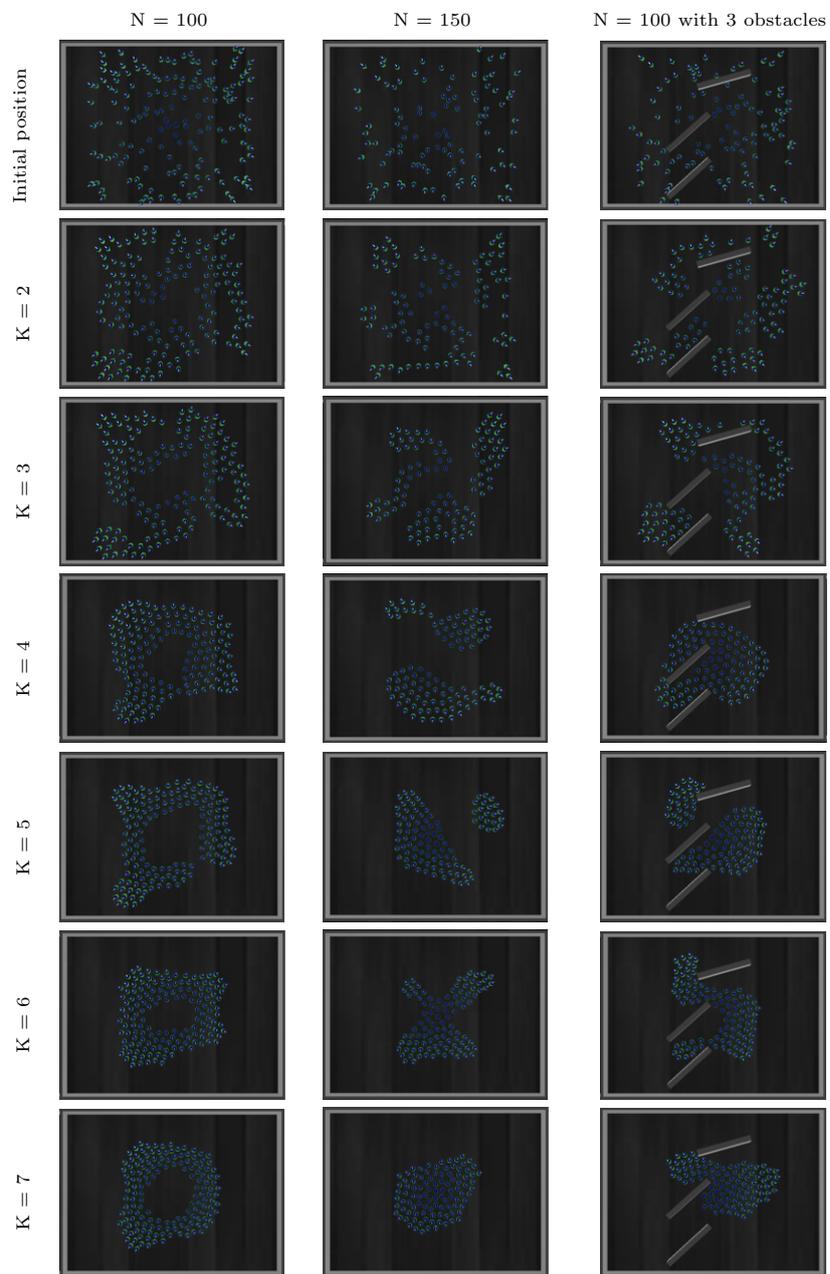


FIGURE 6.4: Self-organized Aggregation Patterns with the KNN Topological Approach.

value of K , the robot can determine the set $\mathcal{T}_k(t)$. This set is then used to compute the viscoelastic force vector as in the previous algorithms using the PC model.

Algorithm 6.2: DW-KNN based Self-Organized Aggregating Pattern Algorithm

```

1 Initialize:  $d_0, L_0, b, k_\omega, k_s, k_d, k_r, v_{max}, K$ 
2 begin
3   for every time step do
4      $\mathcal{N}_i \leftarrow \text{senseNeighbors}()$ 
5      $\rho_i \leftarrow \text{SPHDEC}()$ 
6     Send  $\rho_i$  to neighbors via RAB
7      $\mathcal{T}_k \leftarrow \text{DWKNNC}(\mathcal{N}_i, K)$ 
8      $\hat{p}_i \leftarrow \text{PC}(\mathcal{T}_k)$ 
9      $\mathcal{O}_i \leftarrow \text{senseObstacles}()$ 
10     $\hat{r}_i \leftarrow \text{RC}(\mathcal{O}_i)$ 
11     $\hat{a}_i \leftarrow \text{AddVector}(\hat{p}_i, \hat{r}_i)$ 
12     $\text{setRobotWheelsSpeed}() \leftarrow \text{FDAMC}(\hat{a}_i)$ 

```

With the DW-KNN topological approach, we conducted five separate experiments with different numbers of foot-bot robots ($N = \{50, 100, 150\}$) for a duration of 2000 time steps (ts) each (1 ts = 0.1 sec). Starting from the initial distribution of robots, the same set of constants and parameters used in the KNN study, and which are reported in Table 6.1, were initiated in each experiment. In addition, the smoothing length, h , is set to be $0.5m$ in all the studies realized with the DW-KNN approach.

We demonstrate in Figure 6.5 the self-organized aggregations that developed from the execution of the overall control model by a swarm of $N = \{50, 100, 150\}$ foot-bots robot at diverse time steps ($t = 400, t = 1200$, and $t = 2000$) when starting from the initial positions ($t = 0$). Starting from the same initial position, the robot swarm achieved different self-organized aggregations as only the neighbourhood topology was varied. Figure 6.5a highlights the results obtained from a DW-2NN topology, Figure 6.5b presents the results achieved from a DW-3NN topology, whereas results with DW-4NN and DW-5NN topologies are mentioned in Figure 6.5c and Figure 6.5d, respectively.

Also, we investigated the DW-KNN approach in presence of obstacles while keeping the same parameters setup of the previous scenario. To do so, three obstacles are randomly placed in the arena. Figure 6.6 shows the evolution of the swarm to achieve accurate aggregations while smoothly avoiding obstacles.

It can be seen that in both absence and presence of obstacles, the robots swarm is able to achieve self-organized aggregating patterns via the proposed approach. Moreover, the approach can be useful in scenarios such as driving a large scale of robots from one area to another, while maintaining a connectedness between the robots, and avoiding collisions. Here, the connectivity between the neighbours are modelled using virtual viscoelastic links between the DW-KNN, and the distances toward those neighbours are weighted using an SPH density estimation technic where M4 cubic spline functions are applied. This fact could smoothly drive the swarm to emerge

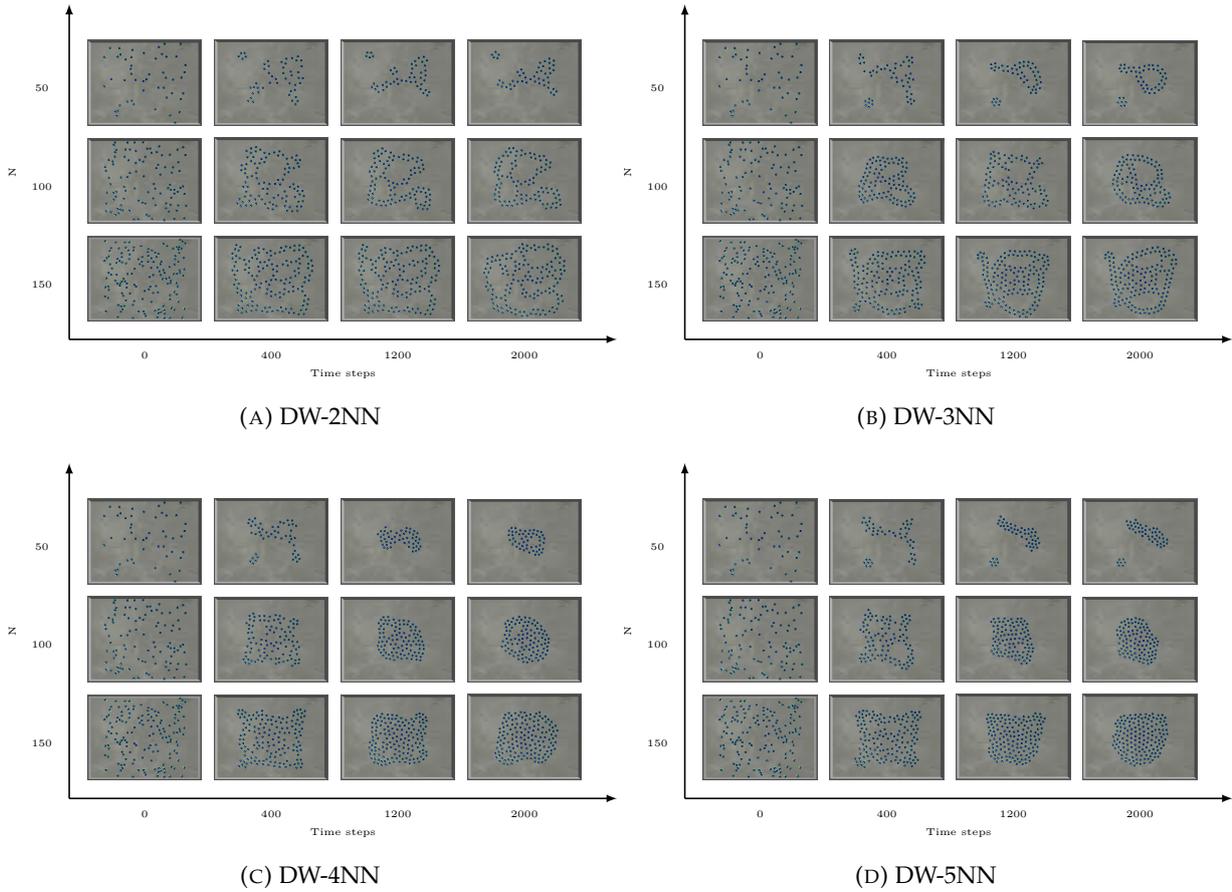


FIGURE 6.5: Self-organized aggregation of a swarm of N foot-bots running the DW-KNN topology

cubic based self-organized aggregating patterns as illustrated in snapshots of Figure 6.5 and Figure 6.6.

It can also be noticed that in presence of obstacles, more clusters could emerge compared to a situation without obstacles in the arena. See, for example, snapshots in Figure 6.6a, Figure 6.6c, and Figure 6.6d for the case when $N = \{50, 100\}$, and snapshots in Figure 6.6b for the case of $N = 50$. This is mainly due to the fact that the robot' field vision will be much influenced in existence of more obstacles, and therefore its range and bearing sensing capabilities will be effectively affected. This means that an obstacle located in a robot' RAB range will blind the vision of a robot to sense and communicate with neighbours. As consequence of an increase of both the size of robots in the swarm and the value of K in the DW-KNN approach results in a decrease of the total number of the clusters that could emerge.

6.4 Performance Analysis

In this section we are interested in studying the performance of the above proposed topological approaches. Specifically, we evaluate how robots, relying only on the KNN and the DW-KNN topologies, achieve self-organized aggregating patterns

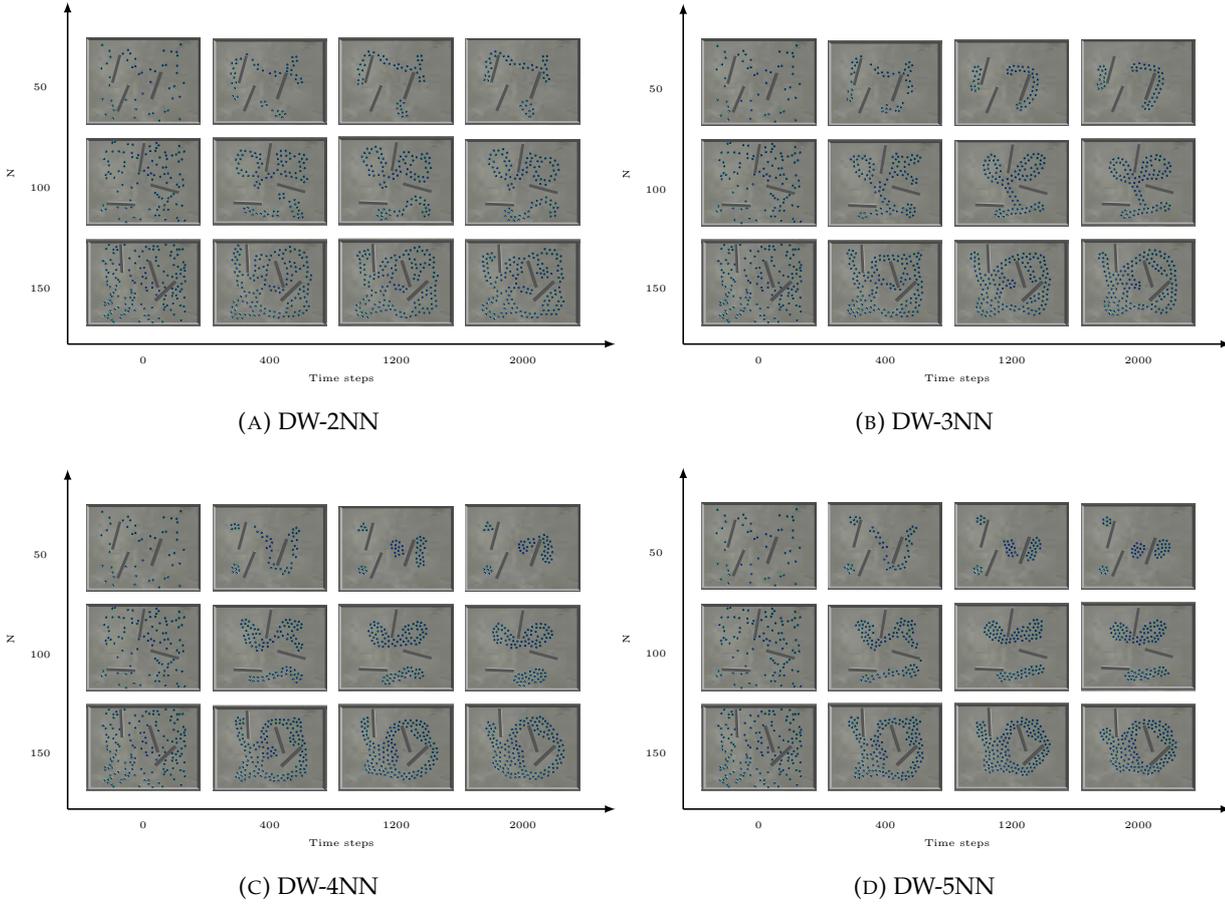


FIGURE 6.6: Self-organized aggregation of a swarm of N foot-bots running the DW-KNN topology with existing obstacles

while maintaining a certain distance between each robot. We also analyze the effect of noise on the models. For that, we use the following metrics to ascertain the attainment of this objective.

6.4.1 Performance Metrics

Distance-Weighted Distribution Quality

We define a new metric to measure the quality of the evolution of the overall weighted distances of the entire swarm. First, the weighted distances averaged over the different robots and neighbours $AWD(t)$ are calculated as follows:

$$AWD(t) = \frac{1}{N \cdot K} \sum_{i=1}^N \left(\sum_{j=1}^K w_{ij}(t) \right). \quad (6.6)$$

Then the distance-weighted quality metric, $F_w(t)$, is gotten by the following equation:

$$F_w(t) = 1 - \frac{1}{\sqrt{AWD(t) + 1}}. \quad (6.7)$$

This metric is only used to assess the performance of the DW-KNN approach.

Aggregation Quality

The aggregation quality [44], $F_{ag}(t)$, is related to the average distance of the robots from their center of mass. To measure this metric, first the distance $c_i(t)$ of each robot R_i from the center of mass of the group at simulation cycle t is computed as

$$c_i(t) = \left\| X_i(t) - \frac{1}{N} \sum_{j=1}^N X_j(t) \right\|, \quad (6.8)$$

where $X_i(t)$ is the position vector of the i^{th} robot at time step t . This value is used to compute the aggregation quality $Ag_i(t)$ of the i^{th} robot as follows:

$$Ag_i(t) = \begin{cases} 0, & c_i(t) < \tilde{r}(n) \\ \frac{R(n)-c_i(t)}{R(n)-\tilde{r}(n)}, & \tilde{r}(n) \leq c_i(t) \leq R(n) \\ 1, & c_i(t) > R(n) \end{cases} \quad (6.9)$$

where $R(n) = \tilde{r}(n) + 100$, $\tilde{r}(n) = r_s \cdot (\sqrt{n} - 1)$ is defined as an approximation of the radius of the smallest circle that has n robots positioned on its perimeter, and r_s is the radius of a robot [44].

Then the average aggregation quality, $F_{ag}(t)$, is defined as follows:

$$F_{ag}(t) = \frac{1}{N \cdot K} \sum_{i=1}^N \left(\sum_{j=1}^K Ag_i(t) \right). \quad (6.10)$$

Dispersion Quality

this metric was adopted by Gauci et al [171] to study aggregation task in a swarm of e-puck robots with the assumption of using minimal resources. Note that, contrary to the aggregation task that seeks to gather the robots together in an area of an environment, the dispersion metric is generally used to show how good a swarm robots system is well dispersed in the environment. Hence as stated in the study of [171], lower bounds of the metric are analyzed as a bad dispersion quality, meaning that the robots are most close to their centroid, and therefore this can be considered as a good aggregation sign.

We adopt this metric to measure the quality of dispersion of the entire swarm. To define this metric, the dispersion quality, $Disp_i(t)$, of a given robot is averaged over its different K neighbours as follows [171]:

$$Disp_i(t) = \frac{1}{4 \cdot r_s^2} \sum_{i=1}^K c_i(t)^2. \quad (6.11)$$

Then the dispersion quality, $F_{disp}(t)$, of the swarm is averaged over the number of the robots,

$$F_{disp}(t) = \frac{1}{N} \sum_{i=1}^N Disp_i(t). \quad (6.12)$$

Where r_s represents the radius of the robot and $c_i(t)$ represents the distance of the robot from the center of mass of the group, and it is computed as in Equation 6.8. The $4r_s^2$ in the denominator serves to normalize $F_{disp}(t)$ such that it becomes independent of r_s for geometrically similar configurations.

Averaged Mean Distance Error

The Averaged Mean Distance Error metric, $AMDE(t)$, is defined as the inter-robots distance error averaged over the different robots and neighbors. It is used mainly to measure how well the swarm maintains a certain desired distance between the individual robots as they move together. The metric differs from the Mean Distance Error metric used in the previous chapter (See sub-section 5.4.1) by omitting the condition δ , where the neighbors that satisfy it were the only ones taken into account. This can be explained by the fact that since a topological approach is used in this study, the number of topological neighbors is previously identified. Therefore the proposed metric is computed as follows:

$$AMDE(t) = \frac{1}{N \cdot K} \sum_{i=1}^N \left(\sum_{j=1}^K (d_{ij}(t) - d_0) \right). \quad (6.13)$$

6.4.2 Analysis in Normal Circumstances

In this sub-section, we assess the performance of the two topological methods when the robots are supposed to perform in normal circumstances. In other hand, the sensors of the robots, in particular the RAB sensors are set to be performing in a perfect manner, with very small tolerant noises. In such circumstances and in order to evaluate the proposed topological approaches more accurately, we investigated analytical studies, within the KNN and the DW-KNN topologies. We performed 25 runs of each experiment for each case K study in the conforming topology, and the experiment duration of each run was set again to 2000 time steps.

With the KNN case study, We analyzed the results collected from simulating a swarm of foot-bot robots composed of $N = \{100, 150\}$ members, while performing the 2NN, 3NN, 4NN and the 5NN topologies in absence of obstacles. In Figure.6.7, we depicted the evolution in time of $F_{ag}(t)$, and $AMDE(t)$ for the four KNN case studies. The curves in the plots show the median values of the 25 runs. It is noticed from that figure, that the entire swarm system successfully converged into a constant value of $F_{ag}(t)$, and tended to a 0 value of $AMDE(t)$ for all N foot-bots and for all the

KNN case studies. This means that the swarm robot system achieved the final self-organized aggregating patterns, when the robots had almost the same aggregation quality with maintaining the same desired inter-robot distance among each other.

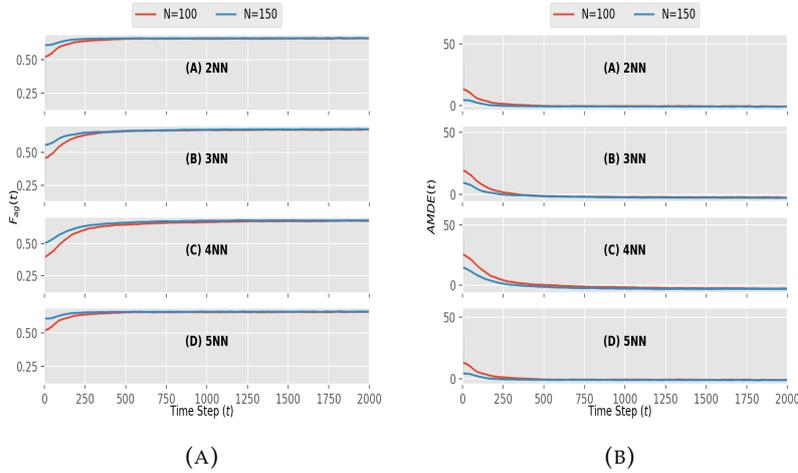


FIGURE 6.7: Performance metrics results in absence of obstacles: (A) $F_{ag}(t)$ and (B) $AMDE(t)$ obtained from the overall controller implemented on $N = \{100, 150\}$ robots when taking into consideration different KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 25 runs with different initial configurations of robots.

With the DW-KNN case study, we plotted the results obtained from the simulation of $N = \{50, 100, 150\}$ foot-bot robots within the DW-2NN, DW-3NN, DW-4NN and the DW-5NN topologies in absence and presence of obstacles. Figure.6.8 and Figure.6.9 show respectively the evolution in time of $F_w(t)$, $F_{ag}(t)$, and $AMDE(t)$ for the four DW-KNN case studies in both absence and presence of obstacles. In all of the plots, the curves show the median values of the 25 runs. As stated in the figures, the swarm robot system successfully converged into a stable value of $F_w(t)$, $F_{ag}(t)$ and $AMDE(t)$ for all N robots and for all topologies, meaning that the final self-organized aggregations are achieved where all the robots had nearly the same density while maintaining the desired inter-robot distance.

6.4.3 The Effect of Sensory Noise on the Performance of the DW-KNN Approach

In this section, we investigate how the aggregation performance of the DW-KNN approach was affected when the readings of the robots range and bearing sensors were corrupted by noise. We modelled noise as in the normal experimental simulation, i.e., as a Gaussian distribution with zero mean and a standard deviation σ , but this time we considered different values of $\sigma = \{0.1, 0.2, 0.3, 0.4, 0.5\}$. The other experimental settings (given in Table 7.2) remained fixed. We set an experiment for each value of σ and performed 25 runs of each experiment. In all of the simulations, we

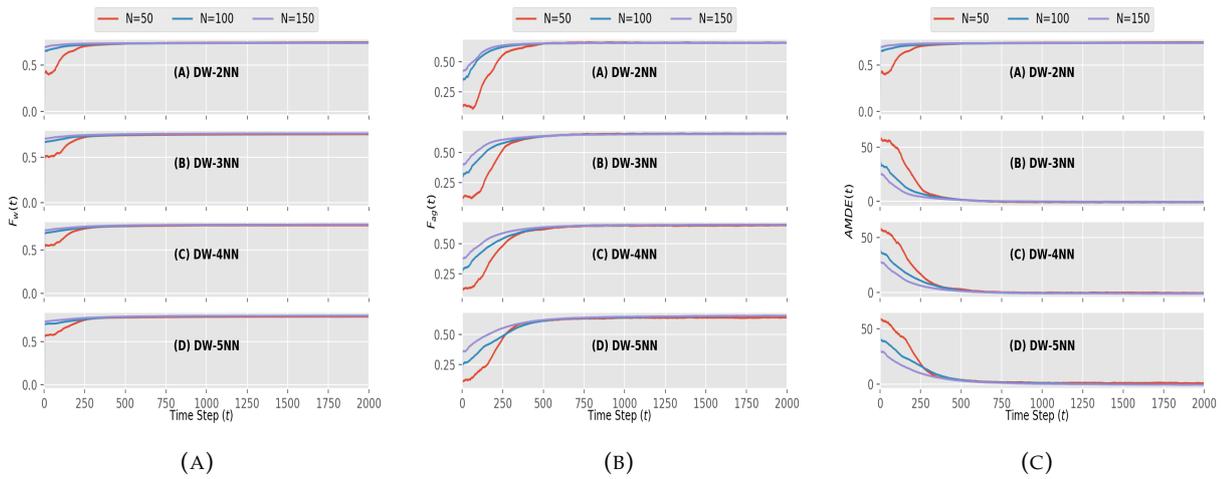


FIGURE 6.8: Performance metrics results in absence of obstacles: (A) $F_w(t)$, (B) $F_{ag}(t)$ and (C) $AMDE(t)$ obtained from the overall controller implemented on $N = \{50, 100, 150\}$ robots when taking into consideration different DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 25 runs with different initial configurations of robots.

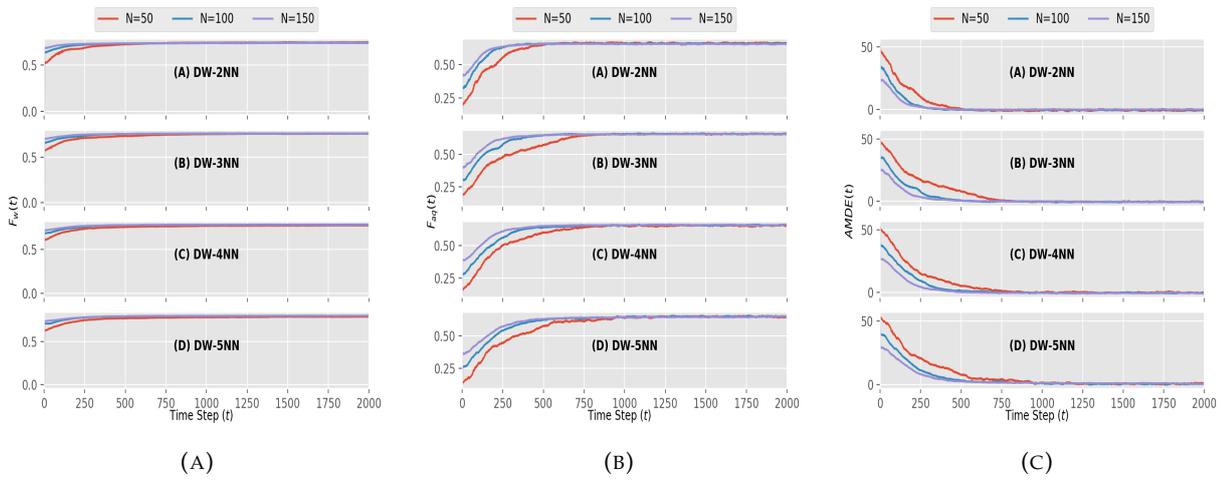
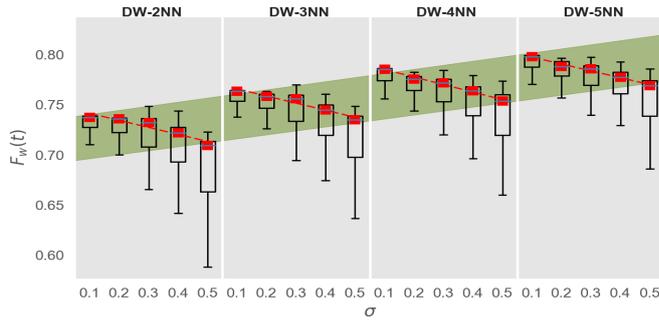


FIGURE 6.9: Performance metrics results in presence of obstacles: (A) $F_w(t)$, (B) $F_{ag}(t)$ and (C) $AMDE(t)$ obtained from the overall controller implemented on $N = \{50, 100, 150\}$ robots when taking into consideration different DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 25 runs with different initial configurations of robots.

fixed $N = 100$ robots, and we used the following DW-KNN topologies in each run: DW-2NN, DW-3NN, DW-4NN, and DW-5NN.

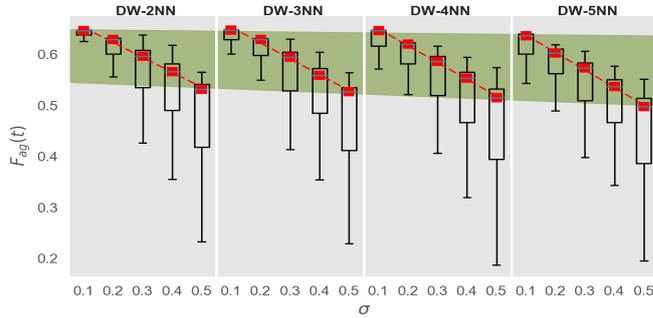
Figure 6.10 plots $F_w(t)$ (Figure 6.10a), $F_{ag}(t)$ (Figure 6.10b), and $AMDE(t)$ (Figure 6.10c) with respect to the different values of σ . In all of the plots, each box represents the metric values obtained from 25 simulations with different initial configurations of 100 robots. The red squares show the mean values, and the dashed red line shows



(A)

Table 6.a: Coefficients of $f(x)$ for $F_w(t)$

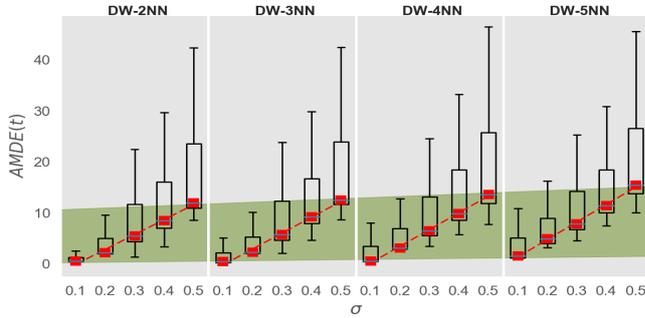
| | c_1 | c_0 |
|------------|-------|--------|
| DW-2NN | 0.748 | -0.007 |
| DW-3NN | 0.773 | -0.007 |
| DW-4NN | 0.793 | -0.007 |
| DW-5NN | 0.804 | -0.007 |
| Olive band | 0.004 | 0.700 |



(B)

Table 6.b: Coefficients of $f(x)$ for $F_{ag}(t)$

| | c_1 | c_0 |
|------------|--------|--------|
| DW-2NN | 0.682 | -0.029 |
| DW-3NN | 0.685 | -0.031 |
| DW-4NN | 0.683 | -0.032 |
| DW-5NN | 0.672 | -0.034 |
| Olive band | -0.002 | 0.550 |



(C)

Table 6.c: Coefficients of $f(x)$ for $AMDE(t)$

| | c_1 | c_0 |
|------------|--------|-------|
| DW-2NN | -2.972 | 2.922 |
| DW-3NN | -3.272 | 3.114 |
| DW-4NN | -3.077 | 3.283 |
| DW-5NN | -2.034 | 3.430 |
| Olive band | 0.062 | 0.268 |

FIGURE 6.10: Performance metrics results: (a) $F_w(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on 100 robots when taking into consideration different DW-KNN topologies with different σ . In all plots, the x-axis represents the standard deviation of noise σ . Each box represents values obtained from 25 runs with different initial configurations of robots. The red squares show the median values, and the dashed red lines shows a linear least squares regression fit to these squares; the olive band shows a linear least squares regression fit to the mean of the 5 red squares for each topology. The linear least squares regression functions generated to fit all of these points are of the form $f(x) = c_1x + c_0$; the corresponding coefficients can be found in the tables to the right of each figure.

a least squares regression fit to the five points of each topology. The olive-coloured band indicates the evolution of the metric with regards to the DW-KNN topology; it represents another linear least squares regression fit to the points of the mean values

of each five red squares. All of the least squares regression fitting functions have the form $f(x) = c_1x + c_0$. The coefficients generated for these functions are highlighted in the tables beside each plot in Figure 6.10.

We can see in Figure 6.10a that, for a given DW-KNN topology, an increase in σ decreases $F_w(t)$. Table 6.10a illustrates the corresponding least squares regression functions generated for each topology. It suggests that for a given topology, $F_w(t)$ decreases slightly and sublinearly in function of σ . Moreover, the olive band in Figure 6.10a shows that $F_w(t)$ is sublinearly increasing with regard to the number K in the DW-KNN topology.

In Figure 6.10b, the aggregation quality $F_{ag}(t)$ in a given topology is sublinearly affected by an increase in σ . We can see that an increasing σ yields a decreasing $F_{ag}(t)$. The related least squares regression functions are depicted in Table 6.10b. However, the olive band in Figure 6.10b indicates that $F_{ag}(t)$ shows a small, decreasing deviation with regard to the number K in the DW-KNN topology. Therefore, it remains almost stable.

The metric $AMDE(t)$ was also affected by noise. Figure 6.10c shows that as more noise was introduced into the RAB, $AMDE(t)$ deviated further from zero. In each topology, the red dashed lines that represent linear least square regression fitting functions indicates that $AMDE(t)$ sublinearly increases with regards to the value of σ . On the other hand, an analysis of the olive band in Figure 6.10c shows that, whatever the value of K is, $AMDE(t)$ has a constant slight, sublinear increase that seems stable.

These observations are as expected. More noise in the RAB yields more mismeasurements of the distances and bearings to neighbouring robots, which immediately affects the quality of all the metrics. While attracting more robots by increasing the number K in the DW-KNN topology does not seem to have a great impact on the quality of the aggregation or the average mean distance error, it does increase the quality of $F_w(t)$.

6.5 Analysis of the Two Approaches within Different Noise Models

we evaluated the performance of the DW-KNN approach as well as the KNN one under the presence of complex noise model. Specifically, we investigated different noise models while comparing the DW-KNN aggregation method to the KNN one. More precisely, we assessed the performance of the two approaches using the following noise models while capturing the measures of the range and bearing sensors:

6.5.1 Effect of Uniform Noise on the Performance Proposed Approach

In this noise model, range and bearing sensors noise are generated from a uniform distribution of $[-3, 3cm]$ for a range measure and of $[-5^\circ, 5^\circ]$ for a bearing measure.

Figure.6.11 illustrates the results of the metrics for both the KNN and the DW-KNN based topological approaches. Both the KNN and the DW-KNN aggregation approaches converges almost to the same quality of aggregation, $F_{Ag}(t)$ and to the same mean distance error, $AMDE(t)$ (see Fig.6.11b and Fig.6.11c). But the DW-KNN approach gives lower dispersion quality than the KNN approach in the case of $K = \{3, 4, 5\}$ see Fig.6.11c), which mean that the robots are better close to their center of masse, and hence it is a good sign of aggregation performance.

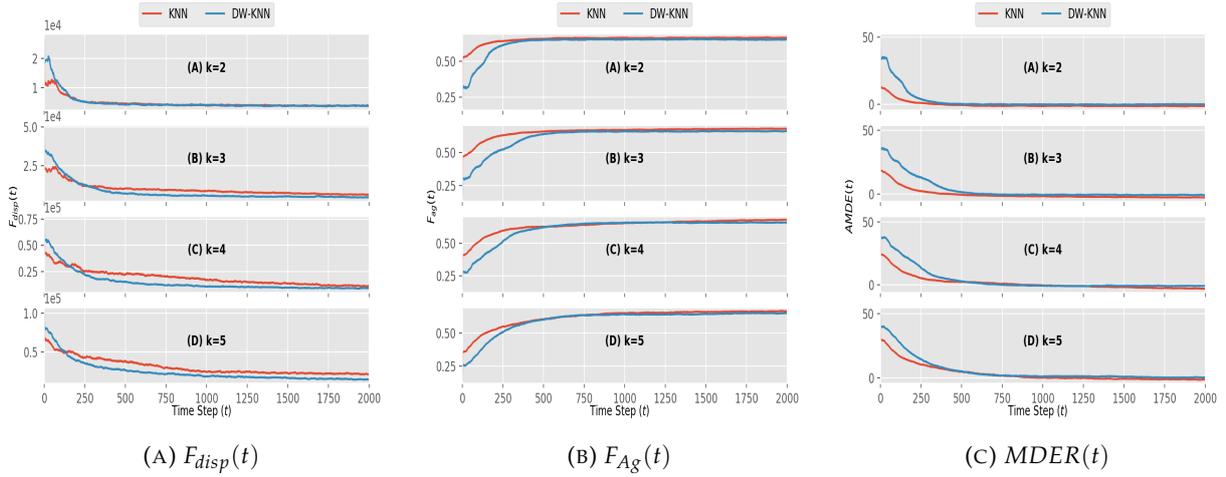


FIGURE 6.11: Performance metrics results for the KNN and the DW-KNN topological Aggregation Approaches with the uniform noise model: (a) $F_{disp}(t)$, (b) $F_{Ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 5 runs with different initial configurations of robots.

6.5.2 Effect of Gaussian Noise on the Performance Proposed Approach

In this noisy model, we add a Gaussian distribution to the true measure of the range and bearing sensor measures of the form $\mathcal{N}(\mu, \sigma^2)$, where $\mu = 0.05$ and $\sigma = 0.1$.

In Figure 6.12, we illustrate the results of the metrics for both studies with the KNN and the DW-KNN based topological approaches. Similar to the uniform noise model, the KNN and the DW-KNN aggregation approaches converges almost to the same quality of aggregation, $F_{Ag}(t)$ and to the same mean distance error, $AMDE(t)$ (see Figure 6.12b and Figure 6.12c). However, the overall aggregation performance of the DW-KNN approach is better than the KNN due the fact that it gives lower dispersion quality than the KNN approach in the case of $K = \{3, 4, 5\}$ (see Figure 6.12a).

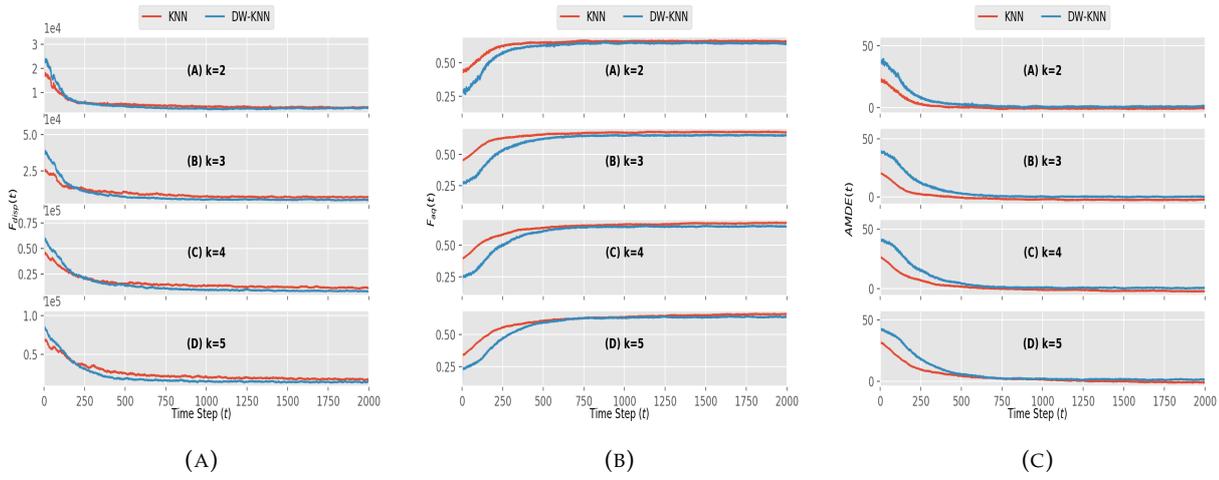


FIGURE 6.12: Performance metrics results for the KNN and the DW-KNN topological Aggregation Approaches with the Gaussian noise model: (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 5 runs with different initial configurations of robots.

6.5.3 Effect of Mean Shift Noise on the Performance Proposed Approach

We add a mean shift noise model to both the range and bearing measurements as follows:

$$Z'_i(t) = Z_i(t) + \zeta_i(t), \quad (6.14)$$

$$\zeta_i(t) = b_i(t) + \mathcal{N}(0, 1), \quad (6.15)$$

Where $Z'_i(t)$ is the noisy measure, $Z_i(t)$ is the true measure, $b_i(t)$ is a bias constant, and $\mathcal{N}(0, 1)$ is a Gaussian distribution of a standard deviation $\sigma = 1$. Figure 6.13 illustrates the results of the metrics for both the KNN and the DW-KNN based topological approaches. As stated in the figure, even though both aggregation approaches converge to the same $F_{Ag}(t)$ and $AMDE(t)$ values (see Figure 6.13b and Figure 6.13c), however the dispersion quality $F_{disp}(t)$ of the DW-KNN is lower than the KNN method meaning that the robots are very close to their group centre, and therefore the overall aggregation performance of the DW-KNN is much better than the KNN aggregation method (see Figure 6.13a).

6.5.4 Effect of Autocorrelated Noise on the performance proposed approach

In this subsection, the performance of the DW-KNN and KNN-based neighbourhood topology to study self-organization in an aggregating robot swarm will be investigated in the presence of autocorrelated noise. Towards this end, the performance of DW-KNN and KNN approaches have been studied when the measurement noise is

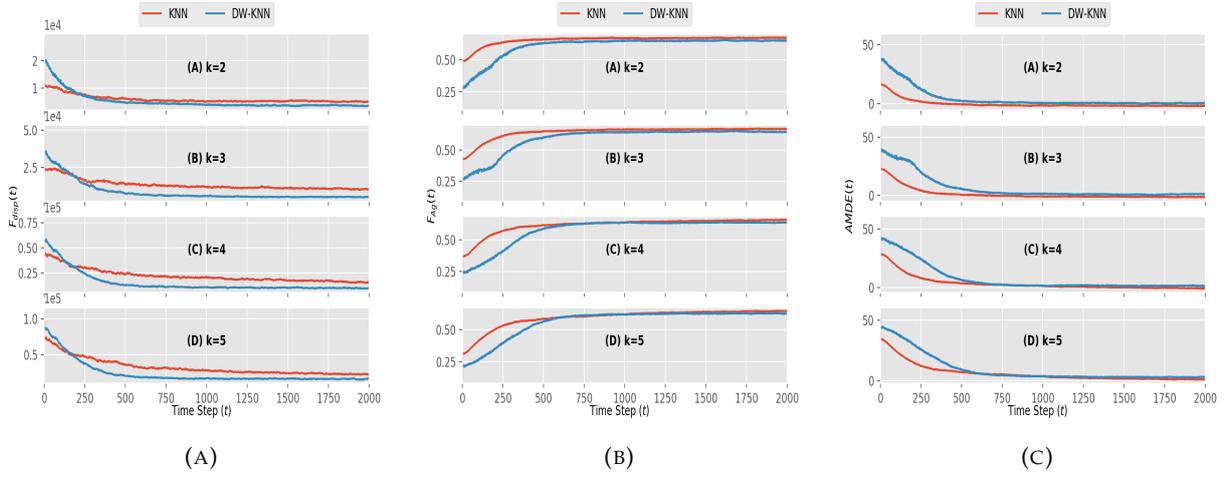


FIGURE 6.13: Performance metrics results for the KNN and the DW-KNN topological Aggregation Approaches with the Mean Shift noise model: (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 5 runs with different initial configurations of robots.

generated from a first-order autoregressive process, or AR(1). Specifically, the measurement noise of the range and bearing sensors are generated using an AR(1) model as follow:

$$Z'_i(t) = Z_i(t) + \zeta_i(t), \quad (6.16)$$

$$\zeta_i(t) = a\zeta_i(t-1) + \varepsilon_i(t), \quad (6.17)$$

Where $Z'_i(t)$ is the noisy measure, $Z_i(t)$ is the true measure, a is the autocorrelation coefficient with lag 1, and $\varepsilon_i(t)$ is a Gaussian distribution of 0 means and a standard deviation 1 ($\mathcal{N}(0, 1)$).

With this noise model, we investigated the performance of the two approaches by conducting simulations for a value of the AR-parameter $a = 0.5$. Figure 6.14 plots successively the results of the two based topological approaches for the value of a . The analysis results are close to the analysis depicted in the previous noise models where the two approaches converge nearly to the same $F_{Ag}(t)$ and the same $AMDE(t)$ in a hand. In another hand, the DW-KNN approach gives more dispersion quality than the KNN approach (a lower value is better). The only difference is that the overall aggregation metrics take much time steps to be converged while compared to the previous noise models.

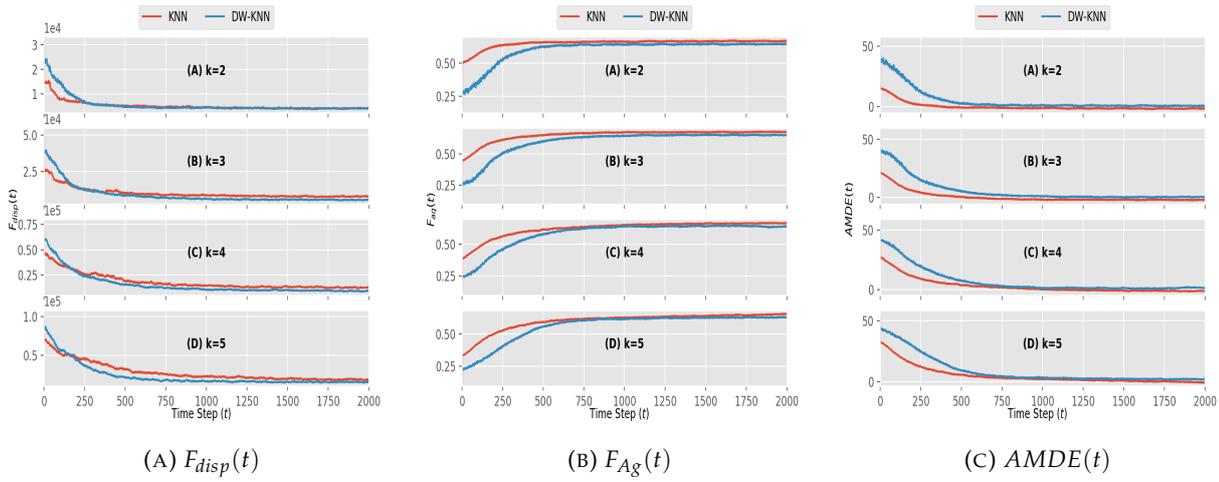


FIGURE 6.14: Performance metrics results for the KNN and the DW-KNN topological Aggregation Approaches with the Auto-Correlated noise model where $a = 0.5$: (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t . Each curve represents the median values obtained from 5 runs with different initial configurations of robots.

6.6 Conclusions

We have addressed the problem of controlling a team of swarm robots to dynamically achieve self-organizing aggregating patterns basing on intra virtual physical connectivity among neighbours. The intra virtual physical connectivity is based on our overall virtual viscoelastic based interaction model proposed in the previous chapter. We defined a neighbouring relationship within only a K robots among the neighbours. Thus by varying this neighbourhood relationship, virtual viscoelastic links are dynamically created and destroyed between the robots and their K sensed neighbours. Each robot of the swarm identified its K nearby mates basing on two topological approaches: a KNN and a DW-KNN approach. In the KNN topological method, the robots interacted only with their K -nearby robots, meaning that the distance toward neighbors were the major factor in selecting this K -neighbors. As a result, the proposed KNN approach is able to achieve different unplanned aggregating patterns that depend on the value of K to be chosen as a configuration based arrangement.

When studying the collective behaviours of a large number of individual robots, the inter-robot distance can be of key importance, but additional properties such as the density of robots could have a greater impact on the collective behaviour of the whole swarm. In the DW-KNN approach, we relied on both the distance and the density as equal key factors to study self-organized patterns in an aggregating robot swarm. This topology, was achieved by defining a distance-weighted function based on an SPH interpolation technic to estimate the density of robots in the swarm.

Regardless if obstacles are either existed or not in the arena, the proposed topological approaches might achieve accurate aggregating patterns. In particular, the DW-KNN approach could smoothly drive the robots swarm to achieve cubic based self-organized aggregations, which could be beneficial in situation when attracting a large scale of robots from one area to another while maintaining a connectedness between the robots and avoiding collision.

With the two approaches, various self-organized based aggregations patterns are achieved using the ARGoS simulator in both absence and presence of obstacles, and performance analysis within four metrics shows the efficacy of the proposed approaches. The effect of noise in the robot range and bearing sensing capabilities is also addressed in this study showing how the proposed models are behaving in such circumstance.

We further compared the proposed aggregation approaches in presence of different noise models (uniform, Gaussian, Mean-shift, and Auto-correlated noise models). Results within the performance metrics (Quality of Aggregation, Mean Distance Error, and Dispersion Quality) shown the efficiency of the DW-KNN approach compared to the KNN one.

Chapter 7

Detecting Faulty Robots in Aggregating robots Swarms

7.1 Overview

Using swarm robotics system, with one or more faulty robots, to accomplish specific tasks may lead to degradation in performances complying with the target requirements. In such circumstances, robot swarms require continuous monitoring to detect abnormal events and to sustain normal operations. In this chapter, an innovative exogenous fault detection method for monitoring robots swarm is presented. The method merges the flexibility of principal component analysis (PCA) models and the greater sensitivity of the exponentially-weighted moving average (EWMA) and cumulative sum (CUSUM) control charts to insidious changes. The method is tested and evaluated on a swarm of simulated foot-bot robots performing a circle formation task, via the viscoelastic control model. We illustrate through simulated data collected from the ARGoS simulator that a significant improvement in fault detection can be obtained by using the proposed method where compared to the conventional PCA-based methods (i.e., T^2 and Q).

7.2 the task and the main objective

While several fault detection techniques have been proposed for robotic swarm systems, MSPC charts have not been used for monitoring in swarm robotics until recently. This chapter focus on monitoring robot swarms using PCA-based fault detection approaches. Principal component analysis (PCA) is a basic method of multivariate analysis and is a powerful tool for monitoring multivariate processes with highly correlated process data. PCA is one of the most commonly used techniques for dimension reduction. Using the PCA method, the covariance structure in data can be explained in a reduced dimensional space through an orthogonal set of principal components (PCs), i.e, a set of linear combinations of the original variables. Faults in the monitored swarm can be detected by extracting useful data from the original dataset through PCA modeling, and then monitoring against those indices. However,

conventional PCA-based monitoring indices such as T^2 and Q charts lose the ability to detect small changes in the mean of process data [172, 173].

The overarching goal of this chapter is to tackle multivariate challenges in process monitoring by merging the advantages of traditional univariate and multivariate techniques to enhance their performance and widen their practical applicability. Exponentially-weighted moving average (EWMA) and cumulative sum (CUSUM) control charts are widely used univariate control charts. The key idea is to apply PCA dimension reduction techniques to the features of a process, and use control charts to monitor only the more informative variables, or principal components. Specifically, we extend the abilities of the univariate monitoring techniques such as EWMA and CUSUM to deal with multivariate processes by developing linear PCA-based EWMA and CUSUM monitoring methods to monitor robotic swarm systems. Note that the main advantage of the PCA-based EWMA and CUSUM fault detection approaches is that the testing step is performed online, which is not the case in a classifier (the classifier algorithms are performed offline rather than online). A decision can be made for each new sample by comparing the value of the EWMA or CUSUM decision statistic with the value of the threshold. An anomaly is declared if the EWMA or CUSUM statistic exceeds the threshold.

The proposed monitoring approach is applied to detect faults in a swarm of foot-bot robots while they are forming a circle. We refer to our virtual viscoelastic control (VVC) model proposed in Chapter 5 for robot swarm circle formation; this model was previously implemented on simulated e-puck robots using the ARGoS simulator. Here we implement the model again on simulated foot-bot robots. During the simulation, we collect various inputs and outputs of data for each robot of the swarm; these data are later used in the PCA model for monitoring.

7.3 PCA-based monitoring approaches

The goal of PCA is to explain the variance/covariance structure through an orthogonal set of linear combination of original variables in the reduced dimensional space. Due to dependency and collinearity, much of the variation can be accounted for by only small number of principal components (PCs).

7.3.1 Feature extraction using PCA

Consider a properly scaled data matrix or measurement matrix $\mathbf{X} = [\mathbf{x}_1^T, \dots, \mathbf{x}_n^T]^T \in \mathbb{R}^{n \times m}$, with n measurements and m process variables. In the following discussion, it is assumed that the scaled data is zero-mean centered with unit variance. Usually, due to redundancy and noise in the data, l , principal components ($l \ll m$) can capture much of the variability in \mathbf{X} . The data matrix \mathbf{X} can be expressed by PCA as two complementary orthogonal parts: a modeled data $\hat{\mathbf{X}}$ which contains the most significant

variations present in the data and a residual data \mathbf{E} which represents noises, i.e.,

$$\mathbf{X} = \mathbf{TP}^T = \sum_{i=1}^l t_i p_i^T + \sum_{i=l+1}^m t_i p_i^T = \widehat{\mathbf{X}} + \mathbf{E} \quad (7.1)$$

where $\mathbf{T} = [t_1 \ t_2 \ \dots \ t_m] \in R^{n \times m}$ represents a matrix of the transformed uncorrelated variables, $t_i \in R^n$ termed principal components (PCs), which are defined as uncorrelated, linear combinations of the original variables that successively maximize the total variance of data projection. l is the number of PCs retained in the PCA model. The column vectors $p_i \in R^m$, termed the loading vectors, arranged in the matrix $\mathbf{P} \in R^{m \times m}$ are obtained by the eigenvectors related to the covariance matrix of \mathbf{X} , i.e., Σ . The loading vectors are the eigenvectors of the covariance matrix, Σ . Through singular value decomposition, Σ can be decomposed as:

$$\Sigma = \frac{1}{n-1} \mathbf{X}^T \mathbf{X} = \mathbf{P} \Lambda \mathbf{P}^T \quad \text{with} \quad \mathbf{P} \mathbf{P}^T = \mathbf{P}^T \mathbf{P} = \mathbf{I}_n. \quad (7.2)$$

Here, $\Lambda = \text{diag}(\sigma_1^2, \dots, \sigma_m^2)$ is a diagonal matrix containing the eigenvalues of Σ in decreasing magnitude, and \mathbf{I}_n is the identity matrix [174]. In PCA, it is very important to select the optimal number of PCs to be retained in the model [175]. There are many techniques for selecting the dimension l , such as cross-validation, cumulative percent variance (CPV), and variance of reconstruction error. In this paper, the CPV technique is employed to determine the number of retained PCs, l : $CPV(l) = \frac{\sum_{i=1}^l \lambda_i}{\sum_{i=1}^m \lambda_i} \times 100$.

7.3.2 PCA-based fault detection

Once a PCA model based on past normal operation is obtained, it can be used to monitor future deviation from normality. Two monitoring statistics, the T^2 and Q statistics, are usually utilized for fault detection purposes [176]. The T^2 statistic based on the number of retained PCs, l , is defined as [176]:

$$T^2 = \sum_{i=1}^l \frac{t_i^2}{\lambda_i}, \quad (7.3)$$

where λ_i is eigenvalue of the covariance matrix of X . The T^2 statistic measures the variation in the PCs only. A large change in the PC subspace is observed if some points exceed the confidence limit of the T^2 chart, indicating a big deviation in the monitored system. Confidence limits for T^2 at level $(1 - \alpha)$ relate to the Fisher distribution, F , as follows [176]:

$$T_{l,n,\alpha}^2 = \frac{l(n-1)}{n-l} F_{l,n-l,\alpha}, \quad (7.4)$$

where $F_{l,n-l,\alpha}$ is the upper $100\alpha\%$ critical point of F with l and $n-l$ degrees of freedom.

The squared prediction error (SPE) or Q statistic, which is defined as [176]:

$$Q = \mathbf{e}^T \mathbf{e}, \quad (7.5)$$

captures the changes in the residual subspace. $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$ represents the residuals vector, which is the difference between the new observation, \mathbf{x} , and its prediction, $\hat{\mathbf{x}}$, via PCA model. Equation (7.5) provides a direct mean of the Q statistic in terms of the total sum of measured variation in the residual vector \mathbf{e} . The SPE can be considered a measure of the system-model mismatch. The confidence limits for SPE are given by [174]. This test suggests the existence of an abnormal condition when $Q > Q_\alpha$, where Q_α , is defined as:

$$Q_\alpha = \varphi_1 \left[\frac{h_0 c_\alpha \sqrt{2\varphi_2}}{\varphi_1} + 1 + \frac{\varphi_2 h_0 (h_0 - 1)}{\varphi_1^2} \right], \quad (7.6)$$

where c_α is the confidence limits for the $1 - \alpha$ percentile in a normal distribution, $\varphi_i = \sum_{j=l+1}^m \lambda_j^i$, for $i = 1, 2, 3$, and $h_0 = 1 - \frac{2\varphi_1\varphi_3}{3\varphi_2^2}$.

However, the PCA-based T^2 and Q approaches fail to detect small faults [177]. The CUSUM and EWMA charts, which are widely used univariate control charts, are proposed as improved alternatives for fault detection. The objective is to tackle PCA challenges in process monitoring by merging the advantages of the CUSUM, EWMA, and PCA approaches to enhance their performance and widen their practical applicability.

7.4 Univariate statistical control charts

Univariate statistical methods, such as CUSUM and EWMA, have been widely used to monitor industrial processes for many years. These methods are briefly reviewed here.

7.4.1 EWMA monitoring charts

EWMA is a statistic which gives less weight to old data, and more weight to new data. The EWMA charts are able to detect small shifts in the process mean, since the EWMA statistic is a time-weighted average of all previous observations. The EWMA monitoring chart is an anomaly-detection technique widely used by scientists and engineers in various disciplines [178, 179, 180, 181, 182, 183]. Assume that $\{x_1, x_2, \dots, x_n\}$ are individual observations collected from a monitored process. The expression for the EWMA is [183]:

$$z_t = \lambda x_t + (1 - \lambda) z_{t-1} \text{ if } t > 0. \quad (7.7)$$

The starting value z_0 is usually set to the mean of the fault-free data, μ_0 . z_t is the output of EWMA and x_t is the observation from the monitored process at the current time. The forgetting parameter $\lambda \in (0, 1]$ determines how fast EWMA forgets

historical data. We can see that if λ is small, then more weight is assigned to past observations. Thus the chart is tuned to have efficiency for detecting small changes in the process mean. On the other hand, if λ is large, then more weight is assigned to the current observations, and the chart is more suitable for detecting large shifts [183, 182]. As λ approaches zero, EWMA approximates the CUSUM criteria, which gives equal weights to the current and historical observations.

The upper and lower control limits of the EWMA chart for detecting a mean shift are: $UCL/LCL = \mu_0 \pm L\sigma_{z_t}$, where L is a multiplier of the EWMA standard deviation σ_{z_t} , $\sigma_{z_t} = \sigma_0 \sqrt{\frac{\lambda}{(2-\lambda)}[1 - (1-\lambda)^{2t}]}$, and σ_0 is the standard deviation of the fault-free or preliminary data set. The parameters L and λ need to be set carefully [183, 182]. In practice, L is usually set to 3, which corresponds to a false alarm rate of 0.27%. If z_t is within the interval [LCL UCL], then we conclude that the process is under control up to time point t . Otherwise, the process is considered out of control.

7.4.2 Cumulative sum (CUSUM) charts

Like the EWMA chart, CUSUM charts have also a good capacity to detect small shifts in the process mean due to an extensive memory of the process [184]. The CUSUM chart aggregates all the information from past and current samples in the decision procedure. The CUSUM statistic (S_t) is defined as the following [183]:

$$S_t = \sum_{j=1}^t (x_j - \mu_0), \quad (7.8)$$

where t denotes the current time point, S_t is the cumulative sum of all samples, including the most recent, and μ_0 is the targeted process mean. A one-sided CUSUM statistic is computed using the following equation [183]:

$$S_t = \sum_{j=1}^t \left[\bar{x}_j - (\mu_0 + k) \right], \quad (7.9)$$

where k is a parameter used as a reference to detect changes in the process mean. If S_t becomes negative, then the CUSUM statistic is set to zero. An out-of-control process is defined by S_t exceeding the decision interval, which is another parameter needed for the CUSUM charts to function. The parameters k and h are defined as $k = \frac{\Delta}{2}$, and $h = \frac{d\Delta}{2}$, respectively, where $d = \left(\frac{2}{\delta^2}\right) \ln\left(\frac{1-\beta}{\alpha}\right)$, $\delta = \frac{\Delta}{\sigma_x}$, σ_x is the standard deviation of the average of the process variable (x) being monitored, α and β are probabilities, and Δ is the size of the shift in the mean that needs to be detected. In practice, Montgomery recommends using a value of 4σ or 5σ for h [183]. This choice would provide a reasonable detection for a shift of 1σ in the process mean. Numerous variations of the CUSUM exist; for more details see [183].

7.4.3 Combining PCA with CUSUM and EWMA charts

Once a PCA model based on historical, normal data is obtained, it can be utilized to monitor future deviation of the process. In this paper, we combine the advantages of PCA modeling with those of the univariate monitoring charts, CUSUM and EWMA, which results in an improved fault detection system, especially for detecting small faults in highly correlated, multivariate data. Towards this end, we applied CUSUM and EWMA charts to the "minor" components obtained from PCA model. As we know, the principal components (PCs) explain most of the variation in the data; minor components refer to the unimportant or residual information that is not retained in a PCA model. The minor components, which capture the variability that arises from noise, represent the residuals of the process, and may contain redundancies that exist between variables. Thus, the loading vectors related to the minor components actually describe the correlations between variables. Indeed, under normal operation with little noise and few errors, the minor components are close to zero, while they significantly deviate from zero in the presence of abnormal events. In this work, the minor components are used as fault indicator. Few studies have taken the minor components into account when doing PCA analysis.

The implementation of the developed monitoring methods is comprised of two stages: offline modeling and online monitoring. In the offline modeling phase, PCA is performed on the normal operating data (training data) enabling us to obtain a reference model. Then, the fault detection procedure is executed by using the reference PCA model with EWMA and CUSUM charts in the online monitoring phase. The PCA-based CUSUM and EWMA fault detection algorithms are schematically summarized as shown in Table 7.1, which is schematically represented in Figure 7.1.

The methodology of using PCA for statistical process monitoring is illustrated through a simulated robot swarm in the next section.

7.5 Results and discussion

In this study, we perform ARGoS-based experimental simulations on a swarm of foot-bots; the robots are programmed to perform the VVC model to self-organize into a uniform circle from a randomly dispersed distribution. ARGoS comes with a configuration file in which we can set the arena, the robots, their sensors, and their actuators devices. In our simulation setup, we activate the RAB equipment within a range $D_r = 3m$, the arena is set to a closed room of $10 * 6m^2$, the number of the foot-bots is set to $n = 6$, the foot-bots are randomly distributed in the arena, and their orientations are set to be a Gaussian distribution of zero means and a standard deviation of 360° . In ARGoS, the simulation time step is set to 0.1s, with five iterations each experiment, for a total of 1500 time steps. During the experimental simulations, we collect data that are further used as inputs and outputs for the PCA-based monitoring approach; we summarize these data in Table 7.2. Figure (7.2) plots the average of the five running simulations for both the group speed, GS , and the average mean distance error,

TABLE 7.1: PCA-based EWMA and CUSUM fault detection procedures.

| Step | Action |
|------|---|
| 1. | <p>Given:</p> <ul style="list-style-type: none"> • A training fault-free data set that represents the normal process operations and a testing data set (possibly faulty data), • The parameters of the EWMA control scheme: smoothing parameter λ and the control limit width L, |
| 2. | <p>Data preprocessing</p> <ul style="list-style-type: none"> • Scale the data to zero mean and unit variance, |
| 3. | <p>Build the PCA model using the training fault-free data</p> <ul style="list-style-type: none"> • Express the data matrix as a sum of approximate and residual matrices as shown in equation (7.1), • Compute the ignored principal components using PCA. • Compute the control limits of the EWMA and CUSUM control schemes |
| 4. | <p>Test the new data</p> <ul style="list-style-type: none"> • Scale the new data, • Compute the ignored principal components using the builded PCA model, • Compute the EWMA and CUSUM decision statistics, |
| 5. | <p>Check for faults</p> <ul style="list-style-type: none"> • Declare a fault when the EWMA or CUSUM decision function exceeds the control limits previously computed using the training data. |

$AMDE$, of the entire swarm. The plots show that from time step $t = 500$, the robotic swarm system becomes stable and converges to a constant $AMDE$ and a tiny variable GS . Figure (7.3) shows ARGoS-based snapshots in step times ($t = 0, t = 250$, and $t = 500$) during the VVC model simulation with a swarm of six foot-bots.

TABLE 7.2: Data collected from the ARGoS simulation

| Parameter | Description |
|--------------------|-----------------------------------|
| $AMDE$ | Average mean distance error |
| GS | Group speed |
| v_{r_i} | Right wheel forward speed |
| v_{l_i} | Left wheel forward speed |
| \hat{p}_i | Virtual viscoelastic force length |
| $\angle \hat{p}_i$ | Virtual viscoelastic angle |

7.5.1 PCA modeling

In this study, a swarm of six robots is considered. The data matrix X used to build a PCA model contains 3000 observations and 12 variables (i.e., viscoelastic force length

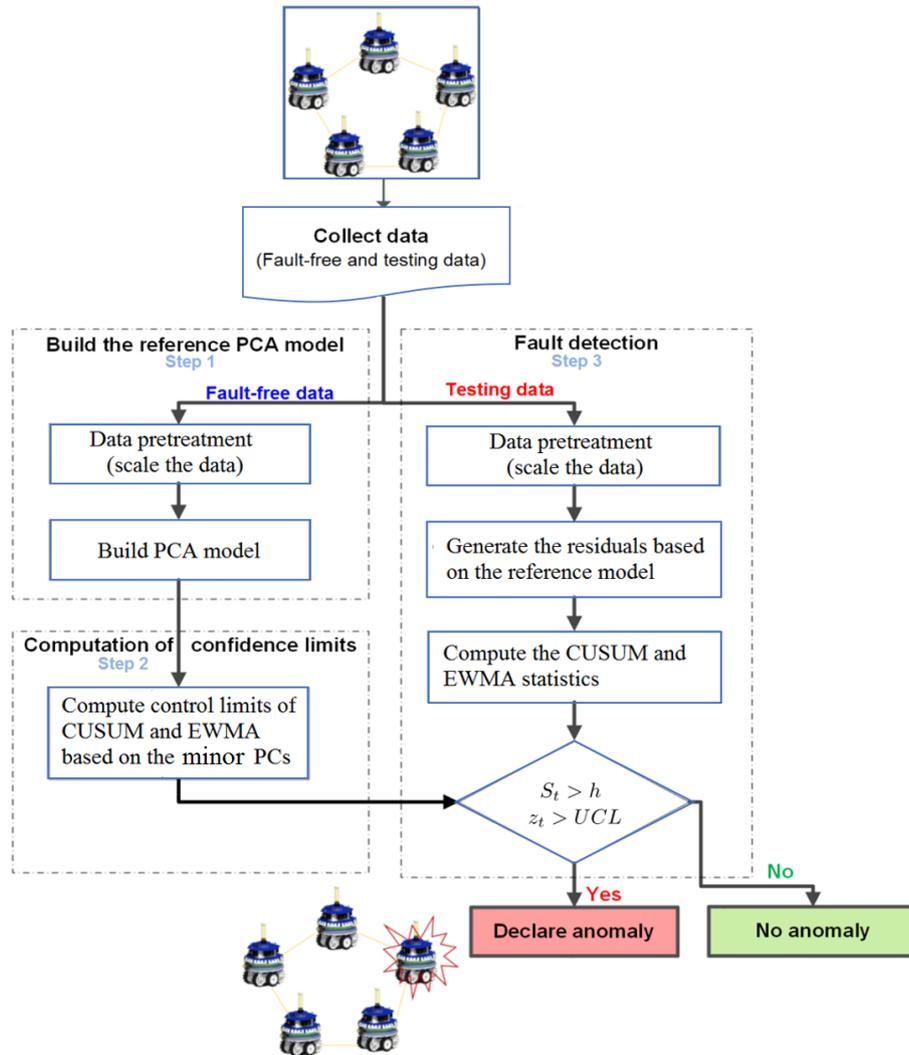
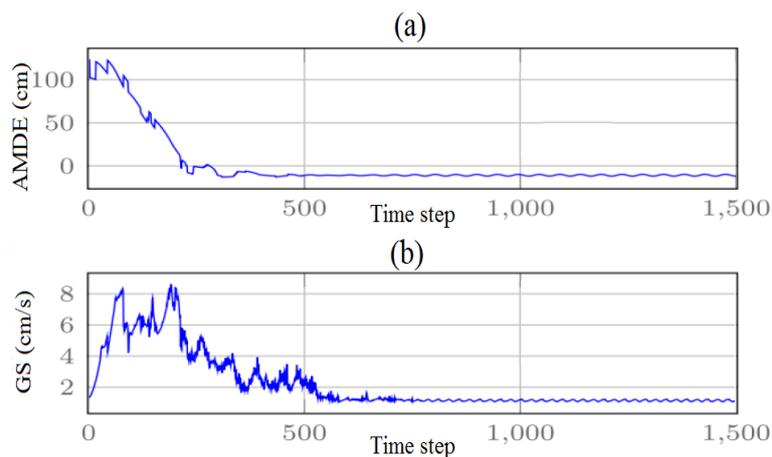


FIGURE 7.1: A flowchart of a PCA-based fault detection schemes.

FIGURE 7.2: Evolution in time of (a) *AMDE*, and (b) *GS*.

and viscoelastic force angle collected from each robot). These twelve signals measured when the swarm system is operating normally. Moreover, all the measured observations are collected during the stabilization phase of the swarm system (from

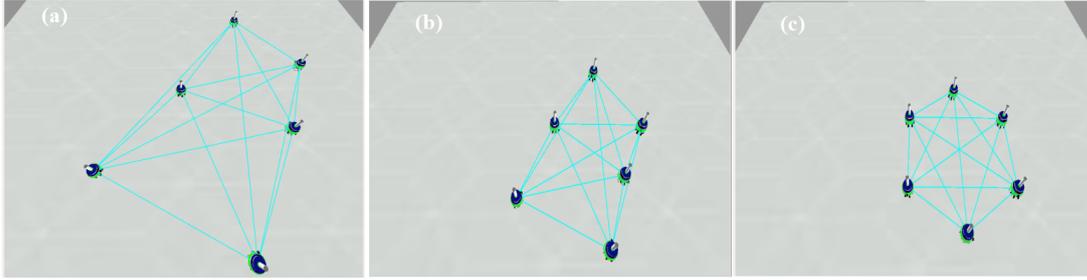


FIGURE 7.3: Snapshots during ARGoS simulation of 6 foot-bots performing the VVC model at: (a) $t=0$, (b) $t=250$ and (c) $t=500$.

starting point of a time window ($t=500$) to the end of the simulation). First, these training data are scaled to zero mean and variance one, then used to build the PCA model. The number of PCs retained in the PCA model are determined using the CPV method with a threshold of 95%. The first PC explains 56% of the total variance; the second PC explains 37% of the total variance, and the third PC explains 3% of the total variance. Together, three PCs can capture 96% of the useful information in the monitored robotic swarm system (see Figure 7.4). Thus, only three PCs need to be retained in the PCA model.

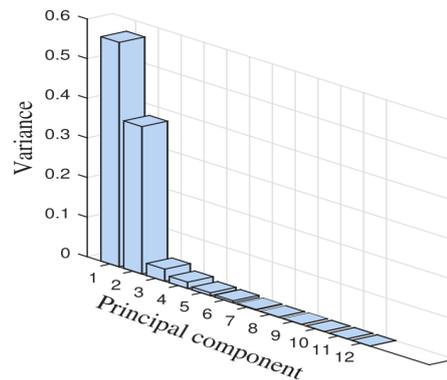


FIGURE 7.4: Three PCs capture 96% of information in the system.

Monitoring results of the PCA-based T^2 , Q , and EWMA charts for the normal operating data are shown in Figure 7.5(a-c). Since the Q plot shown in Figure 7.5(b) is based on normal operating data, one should expect that almost all the data will lie within the 95% confidence interval. Similarly, the data points in the PCA-EWMA and CUSUM charts are also within the 95% confidence limits (see Figure 7.5(c-d)). However, the T^2 plot given in Figure 7.5(a) shows a few false alarms. We can conclude that the PCA model describes the data well when no faults are present.

7.5.2 Detection results

After a system model has been successfully identified, we can proceed with fault detection. Five types of faults in robotic swarm systems will be considered here: abrupt, intermittent, random walk, complete stop, and gradual faults.

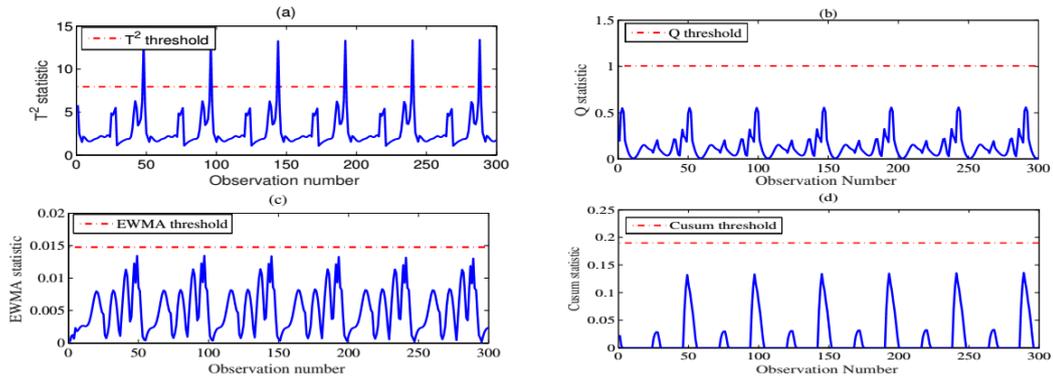


FIGURE 7.5: Monitoring results of PCA- T^2 (a), PCA-Q (b), and PCA-EWMA charts(c) for the normal operation data.

Case (A): Abrupt fault detection

In this case study, an abrupt change is simulated by adding a small, constant deviation to the viscoelastic force length of the first robot, x_1 , between sample times 150 and 200. Since the viscoelastic force is largely related to the RAB device, this could represent a misperception of the range of neighbors or noisy data (velocities) received from neighbours. The two examples below show the performance of the fault detection techniques in detecting an abrupt fault.

Case (A₁): In the first example, the magnitude of the deviation is equal to 40% of the total variation in x_1 . Monitoring results are shown in Figure 7.6(a-d). The T^2 chart, as expected, has no ability to whatsoever to detect this moderate fault (see Figure 7.6(a)). This fact is due to the PCs subspace sometimes being insensitive to moderate and small faults, because each PC is a combination of all process variables. The monitoring results of the PCA-Q, PCA-EWMA, and PCA-CUSUM charts are demonstrated in Figure 7.6(b-d). All the charts show signs of a fault because the bias shift in this case is quite large.

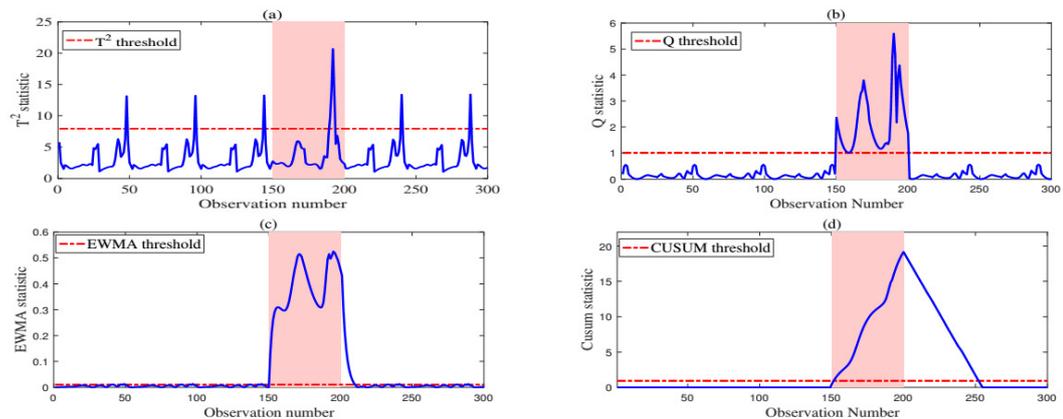


FIGURE 7.6: Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $h = 0.19$) (c) charts in the presence of an abrupt fault in x_1 from sample 150 to 200 (Case (A₁)).

Case (A₂): In the second example, a bias fault of 10% of the total variation is introduced in x_1 between sample times 150 and 200. This could represent a total sensor offset or noisy sensing in the RAB device; this means a possible misperception of both the range and the bearing measurements of neighbors, in addition to possible miscommunications received from neighbors. The four monitoring charts are shown in Figure 7.7(a-d). The T^2 and Q charts are demonstrated in Figure 7.7(a-b), from which we can see that they cannot give any sign of an anomaly. The major reason for this oversight of the conventional PCA-based monitoring methods (i.e., T^2 and Q) is that they use current observation data alone to evaluate system performance ignoring the historical data. We then apply the CUSUM chart with $k = 0.25$ and $h = 0.19$ and the EWMA chart with $\lambda = 0.3$ to the testing dataset. Both statistics clearly exceed the control limits, indicating the occurrence of some abnormal condition. However, the CUSUM chart gave several false alarms, an error rate of 26.4%. Indeed, after conditions return to normal, the CUSUM chart continues to show abnormality for some time, resulting in a large number of false alarms. This case study clearly shows the superiority of the EWMA chart over all other charts.

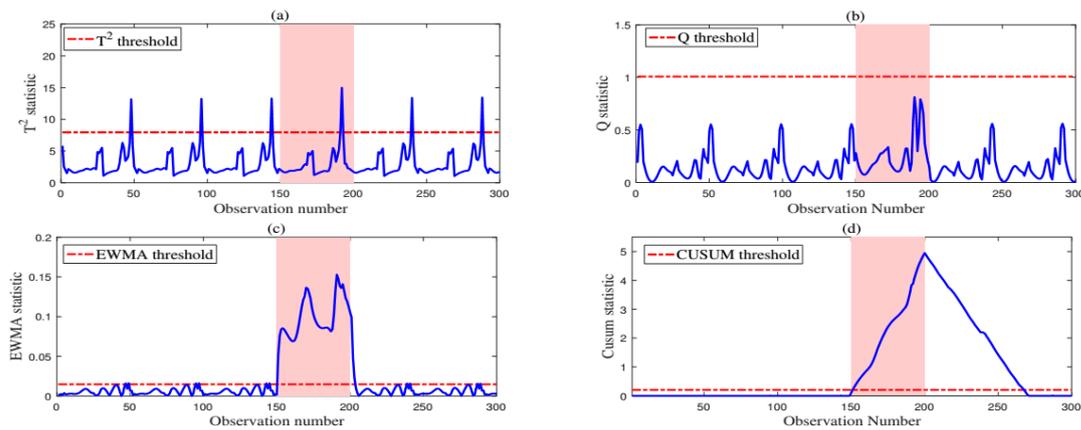


FIGURE 7.7: Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $h = 0.19$) (c) charts in the presence of an abrupt fault in x_1 from sample 150 to 200 (Case (A₂)).

Case (B): Intermittent fault

In this case study, we introduce into the testing data a bias of amplitude 40% of the total variation in x_1 of between samples 50 and 100, and a bias of 10% between samples 150 to 200. This again could be due to a repeated misperception of the range and the bearing measurements for nearby robots or noisy received data (a RAB sensor fault). Figure 7.8(a-d) shows the monitoring results of the PCA-based T^2 , Q , EWMA, and CUSUM charts. Figure 7.8(a) shows that the PCA-based T^2 chart has no power to detect this fault. From Figure 7.8(b), it can be seen that the PCA-EWMA chart can detect the intermittent faults but with several missed detections. It can be seen from Figure 7.8(d) that the PCA-CUSUM chart can indeed detect this fault, but with some missed detections. On the other hand, the PCA-EWMA chart with $\lambda = 0.3$

correctly detects this intermittent fault (see Figure 7.6(c)). In this case study, we can see that detection performance is much enhanced when using the PCA-EWMA chart compared to the others.

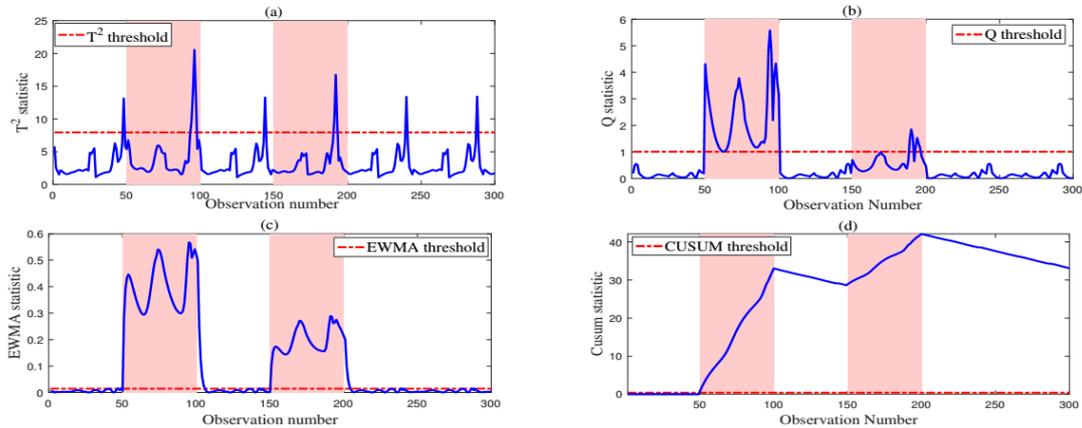


FIGURE 7.8: Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $h = 0.19$) (c) charts in the presence of intermittent faults in x_1 between sample times $[50\ 100]$ and $[150\ 200]$ (Case (B)).

Case (C): Random walk fault

As the movement pattern of swarming robots is highly cross-correlated, we investigate the ability of the proposed approaches to detect a random walk fault in a robot swarm. In this case study, the first robot is performing a random walk and not following the other robots. Such an event could occur when there are noises in the RAB device of the robot. To generate the data with a random walk fault, the viscoelastic force length of the first robot, x_1 , is contaminated with random Gaussian noise with a variance of $\sigma = 0.5$ from sample number 200 until the end of the test data. The four monitoring charts are shown in Figure 7.9(a-d). The PCA- T^2 chart fails to detect this fault, as shown in Figure 7.9(a). Figure 7.9(a) shows that the PCA- Q is able to detect the fault, but with several missed detections. On the other hand, the PCA-based CUSUM and EWMA charts perform reasonably well (see Figure 7.9).

Case (D): Complete stop fault

In this case study, the detection of a complete stop fault in a robot swarm is investigated. In this case study, we consider a complete stop error, which is when a robot has completely stopped working, becoming invisible to neighboring robots. For this purpose, the value of the viscoelastic force of the first robot is zeroed from sampling time 200 until the end of the test data. This means that both the RAB device and the motor actuator of the faulty robot have completely stopped working (the robot can move nor send or receive messages). Here the T^2 chart can detect the fault but with several missed detections (see Figure 7.10). The other three charts, PCA-based Q , CUSUM,

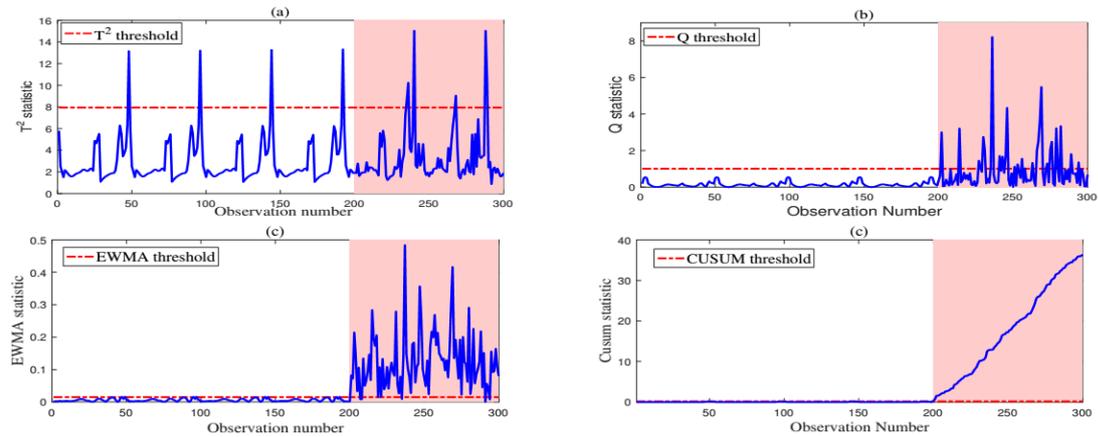


FIGURE 7.9: Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $H = 0.19$) (c) charts when the first robot performs a random walk from sample number 200 through the end of the testing data, Case (C).

and EWMA, all perform reasonably well because the anomaly in this case is relatively large.

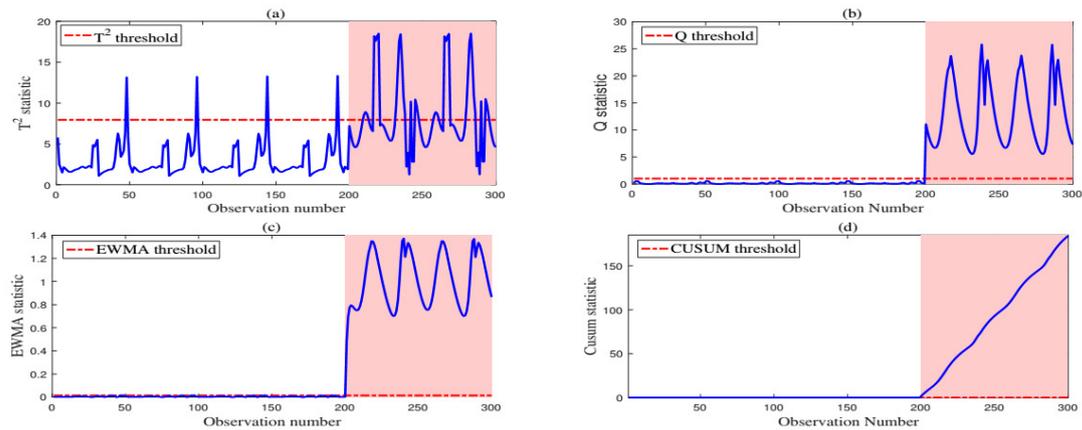


FIGURE 7.10: Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $h = 0.19$) (c) charts when the first robot has completely stopped working between sample times 200-300, Case (D).

To quantify the efficiency of the proposed strategies, we use two metrics: the false detection rate (FAR) and the miss detection rate (MDR) [185]. The FAR is the number of normal observations that are wrongly judged as faulty (false alarms) over the total number of fault-free data. The MDR is the number of faults that are wrongly classified as normal (missed detections) over the total number of faults. The FAR and MDR of the above examples are summarized in Table 7.3. The smaller the FAR and MDR are, the better the detection rate is. From Table 7.3 it can be seen that the developed PCA-EWMA chart provides better detection performances compared to the other charts when detecting small and persistent faults.

TABLE 7.3: False and miss detection rates for all monitoring charts.

| Chart | Case (A ₁) | | Case A ₂) | | Case (B) | | Case (C) | | Case (D) | |
|-------|------------------------|-----|-----------------------|-----|----------|-----|----------|-----|----------|-----|
| | FAR | MDR | FAR | MDR | FAR | MDR | FAR | MDR | FAR | MDR |
| T^2 | 2 | 92 | 2 | 98 | 2 | 93 | 2 | 94 | 2 | 66 |
| Q | 0 | 0 | 0 | 100 | 0 | 44 | 0 | 74 | 0 | 0 |
| CUSUM | 20 | 2 | 26.4 | 2 | 75 | 0 | 0 | 2 | 0 | 0 |
| EWMA | 7.6 | 0 | 4 | 0 | 5.5 | 0 | 3.5 | 5 | 0 | 0 |

Case (E): Drift failure detection

A ramp type, or slow drift, fault is simulated by adding a ramp change to the normal measurements of x_1 from sample 150 through the end of the testing data. This means that either a gradual decrease of the viscoelastic force has occurred due to degradation of a battery, or a sudden increase of robot speed has happened due to problems in the robot's motor. Figure 7.11(a) shows that the PCA- T^2 is not sensitive to this drift fault. The PCA- Q chart is shown in Figure 7.11(b), which first flags the fault at sample 181. Figure 7.11(c) shows that the PCA-EWMA chart first detects the fault at the 157th observation. Therefore, fewer observations are needed for the PCA-EWMA chart to detect a fault compared to the other charts. This case study testifies again to the superiority of the proposed approaches compared to conventional PCA-based fault detection. Of course, this paper also demonstrates through simulated data that significant improvement in fault detection can be obtained by using the PCA model when combined with well established statistical techniques such as the EWMA and CUSUM charts.

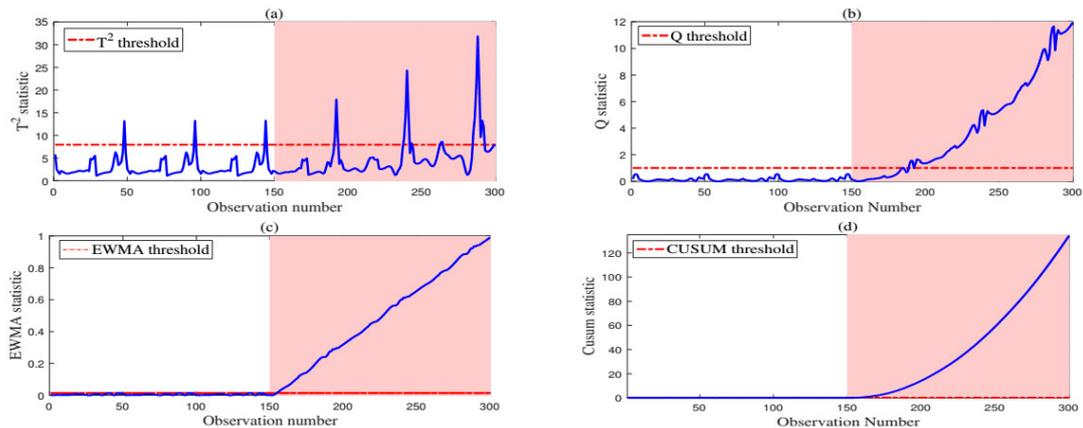


FIGURE 7.11: Monitoring results of the T^2 (a), Q (b), EWMA with $\lambda = 0.3$ (d), and CUSUM (with $k = 0.25$ and $h = 0.19$) (c) charts in the presence of a drift fault with slope 0.01 in x_1 from sample 150, Case (E).

7.6 Conclusion

This chapter focuses on an improved data-based fault detection strategy and its application to fault detection in a swarm of foot-bot robots. Towards this end, the VVC model is used for the circle formation of the robot swarm. Different kind of faults have been tested in this study including abrupt faults, drift faults, random walks and complete stop faults. The swarm data, simulated via the ARGoS simulator, show that significant improvement in fault detection can be obtained by using the EWMA chart instead of the Q or T^2 charts, which are conventionally used with PCA-based techniques. Because the PCA-based T^2 and Q charts evaluate monitored system performance based on the current data alone, they are suitable for detecting relatively large faults. They are less capable of detecting relatively small and persistent shifts, compared to the CUSUM and EWMA charts.

Conventional PCA models are most suitable for dealing with a steady state system. However, in practice systems are usually dynamic and time-varied. Directly applying the PCA method to monitor or model such a process often results in false alarms and model-process mismatch. To adapt to a process drift or change of operating point, we plan in future work to develop a recursive model by updating an online PCA model.

Chapter 8

Conclusion and Future Works

Swarm robotics is a very interesting and challenging sub-domain research, which mainly seeks to bring the theory behind swarms in nature to multi-robotic systems (Chapter 2). Models in swarm robotics are basically inspired from the collective behaviors of social animals such as birds, ants and bees. They are generally applied in studies that address collective tasks problems. As instance, patterns formations and specifically self-organized aggregating patterns are among the challenging problematic that are being a matter of interest in the literature of swarm robotics (Chapter 3).

This problematic was addressed in this thesis by designing synthesized controllers for meaningful swarm robotics collective behaviors. In particular, the controllers were successfully applied to study swarm robotic self-organizing pattern formation tasks. The controllers are mainly based on an artificial viscoelastic model that is inspired from the bio-mechanics properties - such as viscosity and elasticity - of the sub-cellular components. It is shown that the base controller is able to achieve simple geometric formations within two kinds of swarm robotics platforms: foot-bots and e-pucks (Chapter 4). With the foot-bots robots (Chapter 5), we showed the ability of our proposed control model to achieve various regular geometric configurations such as triangles, squares, and pentagons. Later, we adapted the model to address the circle formation task within a swarm of e-puck robots.

Further, in spot of the topological metric strategy revealed in communication studies in birds flocking and other social animals, we extended the basic overall control model to investigate self-organized aggregating patterns using two topological neighborhood approaches: a KNN based aggregation topological approach and a DW-KNN based aggregation topological approach (Chapter 6). In both topological methods, robots do not interact with all the neighbors in their field of visions. Instead, they are interacting only with a restricted number, K , of neighbors. In the KNN approach, the key factor in selecting these K neighbors is the distance toward nearby robots. In this case, robots aggregated basing on their K -nearest teammates, leading to emerge self-organized patterns that can be useful in area coverage and environmental exploration. While in certain swarm applications, where inter-robot distance is not the basic factor on the collective behavior of the swarm, additional properties such as density could play a crucial role on the overall swarm behavior. In the DW-KNN

topological approach, we addressed this issue by introducing a SPH density based technique to estimate densities of the robots in the swarm. Then based on that densities, distances toward neighbors are weighted. The robots then aggregated basing on the nearest K weighted distances, meaning that both the distance and the density are the two key factors in selecting the neighbors. It was noticed that the fact of a SPH density estimation technique is based on a cubic based kernel functions, the DW-KNN topological approach lead in self-organized cubic based aggregating patterns, which could be beneficial in driving a large scale of robots swarm into specific areas with preserving connectivity between the robots and avoiding collisions. Additionally, the two topological methods were tested in absence and presence of obstacles showing the variations in self-organized aggregating patterns that emerge from the proposed topological based control models. Illustrations within, for example, the DW-KNN approach showed that even in presence of obstacles, the swarm could smoothly avoid them while maintaining forming self-organized cubic aggregating patterns.

The performance of the overall controllers presented in this dissertation were all assessed and evaluated using different performance metrics. Results within these metrics showed the efficiency and the scalability of the proposed controllers in studying self-organized aggregating patterns. Results also showed that even in presence of noise in the sensors of the robots, the controllers always achieve accurate performances. It is shown that the DW-KNN method performed better than the conventional KNN approach in presence of different noise models (i.e., uniform, Gaussian noise with mean-shift, and auto-correlated noise models).

Furthermore, while the overall behavior of a swarm robotic system is susceptible to be affected by partial failures mode, a fault detection approach to monitor such failures was proposed to enhance our self-organized control models (Chapter 7). To this end, the proposed fault detection method was investigated and tested on a swarm of foot-bot robots while performing a circle formation task. It is shown that the method merges the flexibility of PCA models and the greater sensitivity of EWMA and CUSUM control charts to induces changes. Results within the swarm data, simulated via the ARGoS simulator, showed that the EWMA chart combined with PCA-based technique leads to a significant improvement in the proposed fault detection approach while compared to the Q or T^2 charts. This is due to the fact that the PCA-based T^2 and Q charts are suitable for detecting relatively large faults and are less capable of detecting relatively small and persistent shifts, compared to the CUSUM and EWMA charts.

We believe that the obtained results within these controllers are meaningful within a broader context, and will contribute in paving the way to the implementation of massively-distributed robotic systems to cope with large scale pattern formation tasks. We think that these controllers could be a starting point to address our motivations in applying robotics swarm systems in the entertainment and art domains. Specifically, when involving swarm robotics systems in artistic self-organized pattern formations.

8.1 Future Works

In spite of the results obtained in this work are very encouraging. However, the controllers presented in this thesis, at least in their current form, are not claimed to be successfully applicable in real swarm robotic systems. For instance, the computation part of the developed controllers might be a matter of interest when dealing with real robots. This is due to the fact that real swarm robotics platforms are generally limited in terms of their physical resources, such as the storage ability of their memory and the capacity of their microcontrollers in processing operations. Despite that the overall controllers presented in this thesis were highly implemented in a simulated version of real robots, using one of the most successful swarm robotics simulators. Yet, these controllers have not been tested in real robots due to the lack of such platforms in most of research laboratories in Algeria. Therefore, we look further to study the overall performance of our control models when re-implementing them in real robotics platforms.

In Chapter 6, we showed that the DW-KNN topological aggregation approach leads to smoothly emerge self-organized based cubic spline patterns, even in presence of obstacle. This was explained to the fact that the connectivity between the DW-KNN members are modelled using virtual viscoelastic links, and the distances toward those members are weighted using an SPH density estimation technic, which is based on a summation of cubic spline kernel functions. Therefore, with this approach, the choice of the density estimation method to be used to weight the distances could play a crucial role in emerging self-organized aggregation patterns. This is let to the scenario in which the approach will be applied, For example, we believe that with the proposed aggregation approach, various B-spline based self-organized aggregations could easily emerge by adopting other kernel functions in the SPH density estimation methods such as those based on the Schoenberg B-spline functions [186] (i.e., the M5 quartic functions and the M6 quantic functions [167]). Moreover, different density estimation methods that differ from the SPH approach (i.e., Gaussian kernel density estimation method) could be used as an alternative to achieve different self-organized aggregating patterns, or to drive the swarm of robots to a desired density distribution. Therefore, we open a door for further studies on how to select the density estimation method for a particular scenario.

We highlighted also, that the obtained self-organized aggregating patterns that might emerge from performing the KNN and the DW-KNN topological approaches, depends strongly to the number K of the conforming topology. Here, the number, K , represents how many neighbors should a robot take into consideration while building a mesh of virtual viscoelastic links. Note that this number was predefined ($K = 2, 3, 4, 5$) in all the corresponding experimental simulations studies of this thesis. Therefore, as a possible direction for future work, we plan to study decision making within the same topological approaches. Specifically, we first seek for each robot to

make a decision on how to dynamically switch its neighboring relationship via an automatic adaptation of the value of K . This could be beneficial in studying the dynamic transformation from one pattern to another within the same framework proposed in this thesis. Second, we would like also to study maintaining cohesively between clusters of robots. Specifically, we plane to study dynamic adaptation of the equilibrium length of the spring both inside the same cluster of the robots and between different clusters of robots. We think these two points can enhance our model to be applied in real applications such as area coverage or environmental exploration.

In Chapter 7, promising results have been obtained within the exogenous fault detection approach proposed to monitor a robot swarm, while performing a circle formation task. Future works in this context seeks to use experimental data, collected during training the model, to test and validate the performance of the proposed approach in detecting faults at real time. Experimental data could be recorded using external tracking systems, or via using on-board sensors. An external tracking system is generally an external infrastructure, with the required captors, that should be installed to record the desired measurements. For example, the Vicon tracking system [187] built at Bristol Robotics Lab (BRL) implements virtual sensors, to allow online evolution of collective behaviors within a swarm of e-puck robots. The OptiTrack system [152] installed at the York Robotics Lab (YRL) provides high precision real-time position tracking, to perform a comparison between the expected and the observed behavior in an e-puck robot augmented with a Linux Extension Board (LEB). However due the height cost of such tracking infrastructures, an alternative approach to be used in our future works is the use of the robot on-board sensors such as the range and bearing (RAB) equipment [188]. The RAB can be used to broadcast the observed data (i.e \hat{p}_i and $\angle \hat{p}_i$) computed by each foot-bot robot to one or more robots that act as observers. The observer(s) will then perform the PCA-based fault detection approach, to independently monitor the behavior of the other robots that are within their range of perception.

Appendix A

Algorithms and Listings

A.1 An Example of an XML based ARGoS Configuration File

LISTING A.1: Config-sample.argos

```

1 <argos-configuration>
2 <!-- ***** -->
3 <!-- * General configuration * -->
4 <!-- ***** -->
5   <framework>
6     <experiment length="0" ticks_per_second="10"/>
7   </framework>
8 <!-- ***** -->
9 <!-- * Controllers * -->
10 <!-- ***** -->
11   <controllers>
12 <!--
13 *****
14 -->
15 <!--
16 * This is the Lua controller to associate to robots *
17 -->
18 <!--
19 *****
20 -->
21   <lua_controller id="lua">
22 <!-- Normal actuator/sensor configuration follows -->
23     <actuators>
24       <differential_steering implementation="default"/>
25       <range_and_bearing implementation="default"/>
26     </actuators>
27     <sensors>
28       <range_and_bearing implementation="medium" medium="rab" show_rays="true"/>
29       <footbot_proximity implementation="default" show_rays="true"/>
30     </sensors>
31 <!-- No required configuration -->
32     <params/>
33   </lua_controller>
34 </controllers>
35 <!-- ***** -->
36 <!-- * Arena configuration * -->

```

```

37 <!-- ***** -->
38 <arena size="3,3,1" center="0,0,0.5">
39
40 <!-- Place four boxes in a square to delimit the arena -->
41 <box id="wall_north" size="2,0.1,0.5" movable="false">
42   <body position="0,1,0" orientation="0,0,0" />
43 </box>
44 <box id="wall_south" size="2,0.1,0.5" movable="false">
45   <body position="0,-1,0" orientation="0,0,0" />
46 </box>
47 <box id="wall_east" size="0.1,2,0.5" movable="false">
48   <body position="1,0,0" orientation="0,0,0" />
49 </box>
50 <box id="wall_west" size="0.1,2,0.5" movable="false">
51   <body position="-1,0,0" orientation="0,0,0" />
52 </box>
53
54 <!-- Place a foot-bot in the origin and bind it to the controller -->
55 <foot-bot id="fb_0">
56   <body position="0,0,0" orientation="0,0,0" />
57   <controller config="lua"/>
58 </foot-bot>
59
60 </arena><!-- ***** -->
61 <!-- * Physics engines * -->
62 <!-- ***** -->
63 <physics_engines>
64   <dynamics2d id="dyn2d"/>
65 </physics_engines>
66 <!-- ***** -->
67 <!-- * Media * -->
68 <!-- ***** -->
69 <media>
70   <range_and_bearing id="rab" index="grid" grid_size="1,1,1"/>
71 </media>
72 <!-- ***** -->
73 <!-- * Visualization * -->
74 <!-- ***** -->
75 <visualization>
76 <!--
77 *****
78 -->
79 <!--
80 * To activate the Lua editor, just use the 'lua_editor' flag attribute *
81 -->
82 <!--
83 *****
84 -->
85 <qt-opengl lua_editor="true">
86   <camera>
87     <placement idx="0" position="-0.00569879,0.01,4.86243" look_at="-0
      .00569879,0.01,3.86243" lens_focal_length="20"/>

```

```
88     <placement idx="1" position="-2,0,2" look_at="0,0,0" lens_focal_length="20"
      />
89     <placement idx="2" position="0.884183,0.359128,0.490269" look_at="0.924486,
      -0.486744,-0.0415919" lens_focal_length="20"/>
90   </camera>
91 </qt-opengl>
92 </visualization>
93 </argos-configuration>
```

A.2 The Overall KNN Control Algorithm Implemented in a Foot-Bot Robot

Algorithm A.1: Part I

Global Parameters: *Define the global parameters in Table.6.1*

```

1 struct {
  |   float: distance, angle, speed
2 } Vect
3 Function init()
4   |   Initialize Global Parameters
5   |   return
  // function executed every time step
6 Function step()
  |   // Send  $v_i$  to neighbor via RAB
7   |   robot.range_and_bearing.set_data(1,  $v_i$ )
  |   // Build the KNN mesh
8   |    $\mathcal{M}_i \leftarrow \text{KNNC}()$ 
  |   // Call the Proximal Control Function
9   |    $\hat{p}_i(x, y) \leftarrow \text{PC}(\mathcal{M}_i)$ 
  |   // Call the Repulsive Control Function
10  |    $\hat{r}_i(x, y) \leftarrow \text{RC}()$ 
  |   // Compute vector  $\hat{a}_i$ 
11  |    $\hat{a}_i(x, y) \leftarrow (\hat{p}_i.x + \hat{r}_i.x, \hat{p}_i.y + \hat{r}_i.y)$ 
  |   // Call the Motion Control Function
12  |    $(v_{l_i}, v_{r_i}) \leftarrow \text{FDAMC}(\hat{a}_i)$ 
  |   // actuate the robot' s wheel speeds
13  |   robot.wheels.set_velocity ( $v_{l_i}, v_{r_i}$ )
14  |   return
15 Function PC()
  |   Input:  $v_i, \mathcal{M}_i$ 
16  |    $result(x, y) \leftarrow (0, 0)$ 
  |   // Collect data from the robot local RAB sensors
17  |   for  $j = 1..|\mathcal{M}_i|$  do
  |   |   // compute virtual voigt force generated by neighbor j using
  |   |   equation (5.3)
18  |   |    $f(x, y) \leftarrow \text{ComputeVoigtForce}(\mathcal{M}_i[j], v_i)$ 
19  |   |    $result(x, y) \leftarrow result(result.x + f.x, result.y + f.y)$ 
20  |   end
21  |   return  $result(x, y)$ 

```

Algorithm A.2: Part II**22 Function RC()****Using:** *Proximity Sensors*23 $result(x, y) \leftarrow (0, 0);$ 24 $\mathcal{F}_i(x, y) \leftarrow (0, 0);$ 25 **for** $i = 1..24$ **do**26 $x_i \leftarrow robot.proximity[i].value * \cos(robot.proximity[i].angle);$ 27 $y_i \leftarrow robot.proximity[i].value * \sin(robot.proximity[i].angle);$ 28 $\mathcal{F}_i(x, y) \leftarrow \mathcal{F}_i(\mathcal{F}_i.x + x_i, \mathcal{F}_i.y + y_i);$ // Now \mathcal{F}_i is the vector that points to the direction to theclosest obstacle where $\|\mathcal{F}_i\| = \sqrt{\mathcal{F}_i.x^2 + \mathcal{F}_i.y^2}/24$ and $\angle \mathcal{F}_i = \text{atan2}(\mathcal{F}_i.y, \mathcal{F}_i.x)$

// Compute the repulsive force vector using eq.5.9

29 $result(x, y) \leftarrow computeRepulsiveForce(\mathcal{F}_i(x, y))$ 30 **return** $result(x, y);$ **31 Function KNNC()****Using:** *Range and Bearing Sensors (RAB)***Input:** K 32 $\mathcal{S} \leftarrow \emptyset;$

// Collect data from the robot RAB sensors

33 **for** $i = 1..|robot.range_and_bearing|$ **do**34 $\mathcal{V} \leftarrow new(Vect);$ 35 $\mathcal{V}.distance \leftarrow robot.range_and_bearing[i].range;$ 36 $\mathcal{V}.angle \leftarrow robot.range_and_bearing[i].horizontal_bearing;$ 37 $\mathcal{V}.speed \leftarrow robot.range_and_bearing[i].data[1];$ 38 $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{V};$ 39 $SortByNearestDistances(\mathcal{S});$ 40 $\mathcal{R} \leftarrow getTheFirstKElements(\mathcal{S}, K);$ 41 **return** $\mathcal{R};$ **42 Function FDAMC(force)****Input:** *force*// Compute the angular and the forward speed (ω_i, v_i) using equations (5.10 and 5.11)43 $(\omega_i, v_i) \leftarrow computeAngularAndForwardSpeed(force);$ // Compute the velocities (v_{li}, v_{ri}) of the left and the right wheels of the robot using equation (4.4)44 $(v_{li}, v_{ri}) \leftarrow computeLeftAndRightSpeed(\omega_i, v_i);$ 45 **return** $(v_{li}, v_{ri});$

A.3 The Overall DW-KNN Control Algorithm Implemented in a Foot-Bot Robot

Algorithm A.3: Part I

Global Parameters: Define the global parameters in Table.6.1

```

1 struct {
  | float: distance, angle, speed, weight
2 } Vect
3 Function init()
4   Initialize Global Parameters
5   return
  // function executed every time step
6 Function step()
  // Send  $v_i$  to neighbor via RAB
7   robot.range_and_bearing.set_data(1,  $v_i$ )
  // Call the SPH Density Estimation Control Function
8    $\rho_i \leftarrow$  SPHDEC()
  // Send  $\rho_i$  to neighbor via RAB
9   robot.range_and_bearing.set_data(2,  $\rho_i$ )
  // Build the Distance-Weighted KNN mesh
10   $\mathcal{T}_i \leftarrow$  DWKNNC()
  // Call the Proximal Control Function
11   $\hat{p}_i(x, y) \leftarrow$  PC( $\mathcal{T}_i$ )
  // Call the Repulsive Control Function
12   $\hat{r}_i(x, y) \leftarrow$  RC()
  // Compute vector  $\hat{a}_i$ 
13   $\hat{a}_i(x, y) \leftarrow$  ( $\hat{p}_i.x + \hat{r}_i.x, \hat{p}_i.y + \hat{r}_i.y$ )
  // Call the Motion Control Function
14  ( $v_{l_i}, v_{r_i}$ )  $\leftarrow$  FDAMC( $\hat{a}_i$ )
  // actuate the robot' s wheel speeds
15  robot.wheels.set_velocity ( $v_{l_i}, v_{r_i}$ )
16  return
17 Function SPHDEC()
  Using: Range and Bearing Sensors (RAB)
  Input:  $h$ 
18   $\rho_i \leftarrow 0$ 
  // Collect data from the robot local RAB sensors
19  for  $j = 1..|robot.range\_and\_bearing|$  do
20     $d_{ij} \leftarrow$  robot.range_and_bearing[ $i$ ].range
    // Compute  $W(d_{ij}, h)$  and update  $\rho_i$  using eq.(6.4)
21     $W \leftarrow$  computeW( $d_{ij}, h$ )
22     $\rho_i \leftarrow \rho_i + W$ 
23  end
24  return  $\rho_i$ 

```

Algorithm A.4: Part II

```

25 Function PC()
    Input:  $v_i, \mathcal{T}_i$ 
26    $result(x, y) \leftarrow (0, 0);$ 
    // Collect data from the robot local RAB sensors
27   for  $j = 1..|\mathcal{T}_i|$  do
    // compute virtual voigt force generated by neighbor j using
    // equation (5.3)
28      $f(x, y) \leftarrow ComputeVoigtForce(\mathcal{T}_i[j], v_i);$ 
29      $result(x, y) \leftarrow result(result.x + f.x, result.y + f.y);$ 
30   return  $result(x, y);$ 

31 Function RC()
    Using: Proximity Sensors
32    $result(x, y) \leftarrow (0, 0);$ 
33    $\mathcal{F}_i(x, y) \leftarrow (0, 0);$ 
34   for  $i = 1..24$  do
35      $x_i \leftarrow robot.proximity[i].value * \cos(robot.proximity[i].angle);$ 
36      $y_i \leftarrow robot.proximity[i].value * \sin(robot.proximity[i].angle);$ 
37      $\mathcal{F}_i(x, y) \leftarrow \mathcal{F}_i(\mathcal{F}_i.x + x_i, \mathcal{F}_i.y + y_i);$ 
    // Now  $\mathcal{F}_i$  is the vector that points to the direction to the
    // closest obstacle where  $\|\mathcal{F}_i\| = \sqrt{\mathcal{F}_i.x^2 + \mathcal{F}_i.y^2}/24$  and
    //  $\angle \mathcal{F}_i = atan2(\mathcal{F}_i.y, \mathcal{F}_i.x)$ 
    // Compute the repulsive force vector using eq. 5.9
38    $result(x, y) \leftarrow computeRepulsiveForce(\mathcal{F}_i(x, y))$ 
39   return  $result(x, y);$ 

40 Function DWKNNC()
    Using: Range and Bearing Sensors (RAB)
    Input: K
41    $\mathcal{S} \leftarrow \emptyset;$ 
    // Collect data from the robot RAB sensors
42   for  $i = 1..|robot.range\_and\_bearing|$  do
43      $\mathcal{V} \leftarrow new(Vect);$ 
44      $\mathcal{V}.distance \leftarrow robot.range\_and\_bearing[i].range;$ 
45      $\mathcal{V}.angle \leftarrow robot.range\_and\_bearing[i].horizontal\_bearing;$ 
46      $\mathcal{V}.speed \leftarrow robot.range\_and\_bearing[i].data[1];$ 
47      $\mathcal{V}.weight \leftarrow$ 
    //  $robot.range\_and\_bearing[i].data[2] * robot.range\_and\_bearing[i].range;$ 
48      $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{V};$ 
49    $SortByNearestWeightedDistances(\mathcal{S});$ 
50    $\mathcal{R} \leftarrow getTheFirstKElements(\mathcal{S}, K);$ 
51   return  $\mathcal{R};$ 

52 Function FDAMC(force)
    Input: force
    // Compute the angular and the forward speed  $(\omega_i, v_i)$  using
    // equations (5.10 and 5.11)
53    $(\omega_i, v_i) \leftarrow computeAngularAndForwardSpeed(force);$ 
    // Compute the velocities  $(v_{li}, v_{ri})$  of the left and the right
    // wheels of the robot using equation (4.4)
54    $(v_{li}, v_{ri}) \leftarrow computeLeftAndRightSpeed(\omega_i, v_i);$ 
55   return  $(v_{li}, v_{ri});$ 

```

Bibliography

- [1] Gregory Dudek and Michael Jenkin. *Computational principles of mobile robotics*. Cambridge university press, 2010.
- [2] Zhiguo Shi et al. "A survey of swarm robotics system". In: *Advances in Swarm Intelligence* (2012), pp. 564–572.
- [3] Rolf Pfeifer, Max Lungarella, and Fumiya Iida. "Self-organization, embodiment, and biologically inspired robotics". In: *science* 318.5853 (2007), pp. 1088–1093.
- [4] Minoru Asada et al. "Cognitive developmental robotics: A survey". In: *IEEE Transactions on Autonomous Mental Development* 1.1 (2009), pp. 12–34.
- [5] Rolf Pfeifer, Max Lungarella, and Fumiya Iida. "The challenges ahead for bio-inspired 'soft' robotics". In: *Communications of the ACM* 55.11 (2012), pp. 76–87.
- [6] Marc Raibert et al. "Bigdog, the rough-terrain quadruped robot". In: *IFAC Proceedings Volumes* 41.2 (2008), pp. 10822–10825.
- [7] Hui Keng Lau. "Error Detection in Swarm Robotics: A Focus on Adaptivity to Dynamic Environments". PhD thesis. University of York, 2012.
- [8] Manuele Brambilla et al. "Swarm robotics: a review from the swarm engineering perspective". In: *Swarm Intelligence* 7.1 (2013), pp. 1–41.
- [9] Ying Tan and Zhong-yang Zheng. "Research advance in swarm robotics". In: *Defence Technology* 9.1 (2013), pp. 18–39.
- [10] Iñaki Navarro and Fernando Matía. "An introduction to swarm robotics". In: *ISRN Robotics 2013* (2012).
- [11] Amanda JC Sharkey. "Swarm robotics and minimalism". In: *Connection Science* 19.3 (2007), pp. 245–260.
- [12] Nicholas R Hoff et al. "Two foraging algorithms for robot swarms using only local communication". In: *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*. IEEE. 2010, pp. 123–130.
- [13] Alexandre Campo et al. "Artificial pheromone for path selection by a foraging swarm of robots". In: *Biological cybernetics* 103.5 (2010), pp. 339–352.
- [14] Eliseo Ferrante et al. "Self-organized flocking with a mobile robot swarm: a novel motion control method". In: *Adaptive Behavior* (2012), p. 1059712312462248.

- [15] Michael Rubenstein et al. "Collective transport of complex objects by simple robots: theory and experiments". In: *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2013, pp. 47–54.
- [16] Brian Shucker and John K Bennett. "Scalable control of distributed robotic macrosensors". In: *Distributed Autonomous Robotic Systems 6* (2007), pp. 379–388.
- [17] Belkacem Khaldi and Foudil Cherif. "An Overview of Swarm Robotics: Swarm Intelligence Applied to Multi-robotics". In: *International Journal of Computer Applications* 126.2 (2015), pp. 31–37.
- [18] Belkacem Khaldi and Foudil Cherif. "Swarm robots circle formation via a virtual viscoelastic control model". In: *Modelling, Identification and Control (ICMIC), 2016 8th International Conference on*. IEEE. 2016, pp. 725–730.
- [19] Belkacem Khaldi and Foudil Cherif. "A Virtual Viscoelastic Based Aggregation Model for Self-organization of Swarm Robots System". In: *Conference Towards Autonomous Robotic Systems*. Springer. 2016, pp. 202–213.
- [20] Belkacem Khaldi et al. "Self-Organization in Aggregating Robot Swarms: A DW-KNN Topological Approach". In: *Biosystems* 165 (2018), pp. 106–121.
- [21] Belkacem Khaldi et al. "A distance weighted-based approach for self-organized aggregation in robot swarms". In: *2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)*. IEEE. 2017, pp. 1–6.
- [22] Belkacem Khaldi et al. "An Efficient Aggregation Topological Strategy for Self-Organized Patterns within Robots". In: *2017 3rd International Conference on Electrical Engineering and Control Applications (ICEECA)*. IEEE. 2017, pp. xx–xx.
- [23] Belkacem Khaldi et al. "A measurement-based fault detection approach applied to monitor robots swarm". In: *Systems and Control (ICSC), 2017 6th International Conference on*. IEEE. 2017, pp. 21–26.
- [24] Belkacem Khaldi et al. "Monitoring a robot swarm using a data-driven fault detection approach". In: *Robotics and Autonomous Systems* 97 (2017), pp. 193–203.
- [25] A William. "Modeling Artificial, Mobile Swarm Systems". PhD thesis. Doctoral Thesis. Institute of Technology, California, 2003.
- [26] Jon Timmis, Paul Andrews, and Emma Hart. "On artificial immune systems and swarm intelligence". In: *Swarm Intelligence* 4.4 (2010), pp. 247–273.
- [27] Michael G Hinchey, Roy Sterritt, and Chris Rouff. "Swarms and swarm intelligence". In: *Computer* 40.4 (2007).
- [28] Amelia Ritahani Ismail. "Immune-inspired self-healing swarm robotic systems". PhD thesis. University of York, 2011.

- [29] Alexandre Campo. "On the Design of Self-Organized Decision Making in Robot Swarms". PhD thesis. Université Libre de Bruxelles, 2011.
- [30] Hazem Ahmed and Janice Glasgow. "Swarm intelligence: concepts, models and applications". In: *School Of Computing, Queens University Technical Report* (2012).
- [31] Peter Miller. *Swarm theory*. 2007 (accessed 2018). URL: <http://ngm.nationalgeographic.com/2007/07/swarms/miller-text.html>.
- [32] Thomas D Seeley and P Kirk Visscher. "Group decision making in nest-site selection by honey bees". In: *Apidologie* 35.2 (2004), pp. 101–116.
- [33] Simon Garnier, Jacques Gautrais, and Guy Theraulaz. "The biological principles of swarm intelligence". In: *Swarm Intelligence* 1.1 (2007), pp. 3–31.
- [34] Gustaf Olson. *Emergent Schooling Behavior in Fish*. Available at <https://pdfs.semanticscholar.org/1f73/012456e79c5158f1c4d316d2fa676776d67e.pdf>. 2008.
- [35] Gerardo Beni and Jing Wang. "Swarm intelligence in cellular robotic systems". In: *Robots and Biological Systems: Towards a New Bionics?* Springer, 1993, pp. 703–712.
- [36] N Bighnaraj et al. "Cooperative Swarm based Evolutionary Approach to find optimal cluster centroids in Cluster Analysis". In: *IJCSI International Journal of Computer Science* 9 (2012).
- [37] Marco Dorigo, Mauro Birattari, et al. "Swarm intelligence". In: *Scholarpedia* 2.9 (2007), p. 1462.
- [38] F Eliseo. "A Control Architecture for a Heterogeneous Swarm of Robots". In: *Rapport d'avancement de recherché (PhD), Université Libre De Bruxelles* (2009).
- [39] Hiroaki Kitano. "Biological robustness". In: *Nature Reviews Genetics* 5.11 (2004), pp. 826–837.
- [40] Xin-She Yang. "Swarm-Based Metaheuristic Algorithms and No-Free-Lunch Theorems". In: *Theory and New Applications of Swarm Intelligence*. InTech, March 2012. Chap. 1, pp. 1–16.
- [41] Mehdi Neshat et al. "Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications". In: *Artificial Intelligence Review* (2014), pp. 1–33.
- [42] Amira Hamdi et al. *Bee-based algorithms: a review*. Available at https://www.researchgate.net/profile/Amira_Hamdi/publication/237844606_META10_Bee_based_algorithms_a_review_META_2010/links/0046351be9feb2e247000000/META10-Bee-based-algorithms-a-review-META-2010.pdf. 2011.
- [43] Marco Dorigo et al. "The swarm-bots project". In: *International Workshop on Swarm Robotics*. Springer. 2004, pp. 31–44.

- [44] Marco Dorigo et al. "Evolving self-organizing behaviors for a swarm-bot". In: *Autonomous Robots* 17.2 (2004), pp. 223–245.
- [45] Heiko Hamann. *Space-time continuous models of swarm robotic systems: Supporting global-to-local programming*. Vol. 9. Springer Science & Business Media, 2010.
- [46] Aleksandar Jevtić and Diego Andina. "Swarm Intelligence and Its Applications in Swarm Robotics". In: *Proceedings of the 6th WSEAS International Conference on Computational Intelligence, Man-machine Systems and Cybernetics. CIMMACS'07*. Tenerife, Canary Islands, Spain: World Scientific, Engineering Academy, and Society (WSEAS), 2007, pp. 41–46. URL: <http://dl.acm.org/citation.cfm?id=1984502.1984510>.
- [47] Erol Şahin. "Swarm robotics: From sources of inspiration to domains of application". In: *International workshop on swarm robotics*. Springer. 2004, pp. 10–20.
- [48] Y Tan. "Swarm robotics: collective behavior inspired by nature". In: *J Comput Sci Syst Biol* 6 (2013), e106.
- [49] Simon Garnier et al. "Aggregation behaviour as a source of collective decision in a group of cockroach-like-robots". In: *ECAL*. Vol. 3630. Springer. 2005, pp. 169–178.
- [50] Onur Soysal, Erkin Bahçeci, and Erol Şahin. "Aggregation in swarm robotic systems: Evolution and probabilistic control". In: *Turkish Journal of Electrical Engineering & Computer Sciences* 15.2 (2007), pp. 199–225.
- [51] Shervin Nouyan, Alexandre Campo, and Marco Dorigo. "Path formation in a robot swarm". In: *Swarm Intelligence* 2.1 (2008), pp. 1–23.
- [52] Paul M Maxim, William M Spears, and Diana F Spears. "Robotic chain formations". In: *IFAC Proceedings Volumes* 42.22 (2009), pp. 19–24.
- [53] Rehan O'Grady, Anders Lyhne Christensen, and Marco Dorigo. "SWARMORPH: multirobot morphogenesis using directional self-assembly". In: *IEEE Transactions on Robotics* 25.3 (2009), pp. 738–743.
- [54] Timothy Stirling and Dario Floreano. "Energy efficient swarm deployment for search in unknown environments". In: *International Conference on Swarm Intelligence*. Springer. 2010, pp. 562–563.
- [55] Frederick Ducatelle et al. "Self-organized cooperation between robotic swarms". In: *Swarm Intelligence* 5.2 (2011), pp. 73–96.
- [56] Ali E Turgut et al. "Self-organized flocking in mobile robot swarms". In: *Swarm Intelligence* 2.2-4 (2008), pp. 97–120.
- [57] Roderich Groß and Marco Dorigo. "Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling". In: *Adaptive Behavior* 16.5 (2008), pp. 285–305.

- [58] Alvaro Gutiérrez et al. "Collective decision-making based on social odometry". In: *Neural Computing and Applications* 19.6 (2010), pp. 807–823.
- [59] Giovanni Pini et al. "Task partitioning in swarms of robots: An adaptive method for strategy selection". In: *Swarm Intelligence* 5.3 (2011), pp. 283–304.
- [60] S Camazine et al. "E. Bonabeau. 2003. Self-Organization in Biological Systems". In: *Princeton University PressCamazineSelf-organization in biological systems2001* (2001).
- [61] Zhibin Xue et al. "Flocking Motion, Obstacle Avoidance and Formation Control of Range Limit Perceived Groups Based on Swarm Intelligence Strategy". In: *Journal of Software* 6.8 (2011), pp. 1594–1602.
- [62] Pierre Broly et al. "Aggregation in woodlice: social interaction and density effects". In: *ZooKeys* 176 (2012), p. 133.
- [63] Pierre Broly, J-L Deneubourg, and Cédric Devigne. "Benefits of aggregation in woodlice: a factor in the terrestrialization process?" In: *Insectes sociaux* 60.4 (2013), pp. 419–435.
- [64] Serge Kernbach et al. "Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system". In: *Adaptive Behavior* 17.3 (2009), pp. 237–259.
- [65] Farshad Arvin et al. "Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method". In: *Adaptive Behavior* 22.3 (2014), pp. 189–206.
- [66] Charlotte K Hemelrijk and Hanno Hildenbrandt. "Schools of fish and flocks of birds: their shape and internal structure by self-organization". In: *Interface focus* (2012), rsfs20120025.
- [67] Yilun Shang and Roland Bouffanais. "Influence of the number of topologically interacting neighbors on swarm dynamics". In: *Scientific reports* 4 (2014).
- [68] Cheng-Ming Chuong and Michael K Richardson. "Pattern formation today". In: *The International journal of developmental biology* 53.5-6 (2009), p. 653.
- [69] L Wolpert. "Diffusible gradients are out-an interview with Lewis Wolpert. Interviewed by Richardson, Michael K." In: *The International journal of developmental biology* 53.5-6 (2008), pp. 659–662.
- [70] Tamás Vicsek and Anna Zafeiris. "Collective motion". In: *Physics Reports* 517.3 (2012), pp. 71–140.
- [71] George F Young et al. "Starling flock networks manage uncertainty in consensus at low cost". In: *PLoS computational biology* 9.1 (2013), e1002894.
- [72] Charlotte K Hemelrijk and Hanno Hildenbrandt. "Some causes of the variable shape of flocks of birds". In: *PloS one* 6.8 (2011), e22479.
- [73] Charlotte K Hemelrijk and Hanno Hildenbrandt. "Self-Organized Shape and Frontal Density of Fish Schools". In: *Ethology* 114.3 (2008), pp. 245–254.

- [74] Craig W Reynolds. "Flocks, herds and schools: A distributed behavioral model". In: *ACM SIGGRAPH computer graphics*. Vol. 21. 4. ACM. 1987, pp. 25–34.
- [75] Zhao Cheng et al. "Aggregation pattern transitions by slightly varying the attractive/repulsive function". In: *PLoS One* 6.7 (2011), e22123.
- [76] Tamás Vicsek et al. "Novel type of phase transition in a system of self-driven particles". In: *Physical review letters* 75.6 (1995), p. 1226.
- [77] Veysel Gazi and Kevin M Passino. "Stability analysis of swarms". In: *IEEE Transactions on Automatic Control* 48.4 (2003), pp. 692–697.
- [78] Iain D Couzin et al. "Collective memory and spatial sorting in animal groups". In: *Journal of theoretical biology* 218.1 (2002), pp. 1–11.
- [79] Hanno Hildenbrandt, Cladio Carere, and Charlotte K Hemelrijk. "Self-organized aerial displays of thousands of starlings: a model". In: *Behavioral Ecology* 21.6 (2010), pp. 1349–1359.
- [80] Michele Ballerini et al. "Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study". In: *Proceedings of the national academy of sciences* 105.4 (2008), pp. 1232–1237.
- [81] Michele Ballerini et al. "Empirical investigation of starling flocks: a benchmark study in collective animal behaviour". In: *Animal behaviour* 76.1 (2008), pp. 201–215.
- [82] Takayuki Niizato, Hisashi Murakami, and Yukio-Pegio Gunji. "Emergence of the scale-invariant proportion in a flock from the metric-topological interaction". In: *Biosystems* 119 (2014), pp. 62–68.
- [83] Veysel Gazi et al. "Aggregation, foraging, and formation control of swarms with non-holonomic agents using potential functions and sliding mode techniques". In: *Turkish Journal of Electrical Engineering & Computer Sciences* 15.2 (2007), pp. 149–168.
- [84] Swaminathan Karthikeyan and Minai A Ali. "A General Approach to Swarm Coordination using Circle Formation". In: *Stigmergic Optimization*. Springer, 2006, pp. 65–84.
- [85] Donghwa Jeong and Kiju Lee. "Dispersion and Line Formation in Artificial Swarm Intelligence". In: *arXiv preprint arXiv:1407.0014* (2014).
- [86] Blesson Varghese and Gerard McKee. "Swarm Patterns: Trends & Transformation Tools". In: *Multi-Robot Systems, Trends and Development*. InTech, 2011.
- [87] Veysel Gazi and Barış Fidan. "Coordination and control of multi-agent dynamic systems: Models and approaches". In: *International Workshop on Swarm Robotics*. Springer. 2006, pp. 71–102.

- [88] Erkin Bahceci, Onur Soysal, and Erol Sahin. "A review: Pattern formation and adaptation in multi-robot systems". In: *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-43* (2003).
- [89] Levent Bayindir and Erol Şahin. "A review of studies in swarm robotics". In: *Turkish Journal of Electrical Engineering & Computer Sciences* 15.2 (2007), pp. 115–147.
- [90] Erol Sahin et al. "Swarm Robotics." In: *Swarm intelligence* 1 (2008), pp. 87–100.
- [91] C Ronald Kube and Hong Zhang. "Collective robotics: From social insects to robots". In: *Adaptive behavior* 2.2 (1993), pp. 189–218.
- [92] Owen Holland and Chris Melhuish. "An interactive method for controlling group size in multiple mobile robot systems". In: *Advanced Robotics, 1997. ICAR'97. Proceedings., 8th International Conference on. IEEE. 1997*, pp. 201–206.
- [93] Grégory Mermoud et al. "Aggregation-mediated collective perception and action in a group of miniature robots". In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 2-Volume 2. International Foundation for Autonomous Agents and Multiagent Systems. 2010*, pp. 599–606.
- [94] Michael Bodi et al. "Interaction of robot swarms using the honeybee-inspired control algorithm BEECLUST". In: *Mathematical and Computer Modelling of Dynamical Systems* 18.1 (2012), pp. 87–100.
- [95] Michael Bodi et al. "BEECLUST used for exploration tasks in Autonomous Underwater Vehicles". In: *IFAC-PapersOnLine* 48.1 (2015), pp. 819–824.
- [96] Farshad Arvin et al. "Comparison of different cue-based swarm aggregation strategies". In: *International Conference in Swarm Intelligence. Springer. 2014*, pp. 1–8.
- [97] Farshad Arvin et al. "Investigation of cue-based aggregation in static and dynamic environments with a mobile robot swarm". In: *Adaptive Behavior* 24.2 (2016), pp. 102–118.
- [98] Farshad Arvin et al. "Imitation of honeybee aggregation with collective behavior of swarm robots". In: *International Journal of Computational Intelligence Systems* 4.4 (2011), pp. 739–748.
- [99] Mostafa Wahby, Alexander Weinhold, and Heiko Hamann. "Revisiting BEECLUST: Aggregation of swarm robots with adaptiveness to different light settings". In: *proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2016*, pp. 272–279.
- [100] Andrew Vardy. "Aggregation in robot swarms using odometry". In: *Artificial Life and Robotics* 21.4 (2016), pp. 443–450.

- [101] Simon Garnier et al. "The embodiment of cockroach aggregation behavior in a group of micro-robots". In: *Artificial life* 14.4 (2008), pp. 387–408.
- [102] Raphael Jeanson et al. "Self-organized aggregation in cockroaches". In: *Animal behaviour* 69.1 (2005), pp. 169–180.
- [103] Nikolaus Correll and Alcherio Martinoli. "Modeling and designing self-organized aggregation in a swarm of miniature robots". In: *The International Journal of Robotics Research* 30.5 (2011), pp. 615–626.
- [104] Onur Soysal and Erol Sahin. "Probabilistic aggregation strategies in swarm robotic systems". In: *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*. IEEE. 2005, pp. 325–332.
- [105] Hideki Ando et al. "Distributed memoryless point convergence algorithm for mobile robots with limited visibility". In: *IEEE Transactions on Robotics and Automation* 15.5 (1999), pp. 818–828.
- [106] Jorge Cortés, Sonia Martínez, and Francesco Bullo. "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions". In: *IEEE Transactions on Automatic Control* 51.8 (2006), pp. 1289–1298.
- [107] Noam Gordon, Israel A Wagner, and Alfred M Bruckstein. "Gathering multiple robotic agents with limited sensing capabilities". In: *International Workshop on Ant Colony Optimization and Swarm Intelligence*. Springer. 2004, pp. 142–153.
- [108] Noam Gordon, Yotam Elor, and Alfred M Bruckstein. "Gathering multiple robotic agents with crude distance sensing capabilities". In: *International Conference on Ant Colony Optimization and Swarm Intelligence*. Springer. 2008, pp. 72–83.
- [109] Maria Carmela De Gennaro and Ali Jadbabaie. "Decentralized control of connectivity for multi-agent systems". In: *Decision and Control, 2006 45th IEEE Conference on*. IEEE. 2006, pp. 3628–3633.
- [110] Vito Trianni et al. "Evolving aggregation behaviors in a swarm of robots". In: *European Conference on Artificial Life*. Springer. 2003, pp. 865–874.
- [111] Melvin Gauci et al. "Evolving aggregation behaviors in multi-robot systems with binary sensors". In: *Distributed autonomous robotic systems*. Springer, 2014, pp. 355–367.
- [112] Yaochu Jin and Yan Meng. "Morphogenetic robotics: An emerging new field in developmental robotics". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 41.2 (2011), pp. 145–160.
- [113] Hongliang Guo, Yan Meng, and Yaochu Jin. "Swarm robot pattern formation using a morphogenetic multi-cellular based self-organizing algorithm". In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE. 2011, pp. 3205–3210.

- [114] Yan Meng, Hongliang Guo, and Yaochu Jin. "A morphogenetic approach to flexible and robust shape formation for swarm robotic systems". In: *Robotics and Autonomous Systems* 61.1 (2013), pp. 25–38.
- [115] Yaochu Jin, Hongliang Guo, and Yan Meng. "A hierarchical gene regulatory network for adaptive multirobot pattern formation". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.3 (2012), pp. 805–816.
- [116] William M Spears et al. *An overview of physicomimetics*. Springer, 2004.
- [117] William M Spears et al. "Distributed, physics-based control of swarms of vehicles". In: *Autonomous Robots* 17.2-3 (2004), pp. 137–162.
- [118] Suranga Hettiarachchi and William M Spears. "Moving Swarm Formations through Obstacle Fields." In: *IC-AI*. 2005, pp. 97–103.
- [119] Suranga Hettiarachchi et al. "Distributed adaptive swarm for obstacle avoidance". In: *International Journal of Intelligent Computing and Cybernetics* 2.4 (2009), pp. 644–671.
- [120] Andrew Howard, Maja J Matarić, and Gaurav S Sukhatme. "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem". In: *Distributed Autonomous Robotic Systems* 5. Springer, 2002, pp. 299–308.
- [121] Christoph Moeslinger, Thomas Schmickl, and Karl Crailsheim. "A minimalist flocking algorithm for swarm robots". In: *Advances in Artificial Life. Darwin Meets von Neumann*. Springer, 2009, pp. 375–382.
- [122] Andrea Gasparri, Attilio Priolo, and Giovanni Ulivi. "A swarm aggregation algorithm for multi-robot systems based on local interaction". In: *Control Applications (CCA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 1497–1502.
- [123] Andrea Gasparri et al. "A swarm aggregation algorithm based on local interaction for multi-robot systems with actuator saturations". In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE. 2012, pp. 539–544.
- [124] Antonio Leccese et al. "A swarm aggregation algorithm based on local interaction with actuator saturations and integrated obstacle avoidance". In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE. 2013, pp. 1865–1870.
- [125] Hiroshi Hashimoto et al. "Stability of swarm robot based on local forces of local swarms". In: *SICE Annual Conference, 2008*. IEEE. 2008, pp. 1254–1257.
- [126] Brian Shucker and John K Bennett. "Virtual spring mesh algorithms for control of distributed robotic macrosensors". In: *University of Colorado at Boulder, Technical Report CU-CS-996-05* (2005).

- [127] Nicola Bezzo and Rafael Fierro. "Decentralized connectivity and user localization via wireless robotic networks". In: *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*. IEEE. 2011, pp. 1285–1290.
- [128] Nicola Bezzo and Rafael Fierro. "Swarming of mobile router networks". In: *American Control Conference (ACC), 2011*. IEEE. 2011, pp. 4685–4690.
- [129] T Dewi, P Risma, and Y Oktarina. "Wedge Formation Control of Swarm Robots". In: *14th Industrial Electronics Seminar IES(2012), Electronic Engineering Polytechnic Institute of Surabaya (EEPIS)*. Indonesia, 2012, pp. 294–298.
- [130] Pablo Urcola et al. "Cooperative navigation using environment compliant robot formations". In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE. 2008, pp. 2789–2794.
- [131] Rolf Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.
- [132] Jon Timmis et al. "An immune-inspired swarm aggregation algorithm for self-healing swarm robotic systems". In: *Biosystems* 146 (2016), pp. 60–76.
- [133] Jan Dyre Bjerknes and Alan FT Winfield. "On fault tolerance and scalability of swarm robotic systems". In: *Distributed autonomous robotic systems*. Springer, 2013, pp. 431–444.
- [134] Alan FT Winfield and Julien Nembrini. "Safety in numbers: fault-tolerance in robot swarms". In: *International Journal of Modelling, Identification and Control* 1.1 (2006), pp. 30–37.
- [135] James O’Keeffe et al. "Towards Fault Diagnosis in Robot Swarms: An Online Behaviour Characterisation Approach". In: *Conference Towards Autonomous Robotic Systems*. Springer. 2017, pp. 393–407.
- [136] Cesare Fantuzzi, Cristian Secchi, and Antonio Visioli. "On the fault detection and isolation of industrial robot manipulators". In: *IFAC Proceedings Volumes* 36.17 (2003), pp. 399–404.
- [137] TESI PER IL CONSEGUIMENTO DEL TITOLO and DI RICERCA DI DOTTORE. "Model-based fault diagnosis in dynamic systems using identification techniques". In: (2002).
- [138] Steven Ding. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media, 2008.
- [139] Anders Lyhne Christensen, Rehan OGrady, and Marco Dorigo. "From fireflies to fault-tolerant swarms of robots". In: *IEEE Transactions on Evolutionary Computation* 13.4 (2009), pp. 754–766.
- [140] Alan G Millard, Jon Timmis, and Alan FT Winfield. "Towards exogenous fault detection in swarm robotic systems". In: *Conference Towards Autonomous Robotic Systems*. Springer. 2013, pp. 429–430.

- [141] Elias N Skoundrianos and Spyros G Tzafestas. "Finding fault-fault diagnosis on the wheels of a mobile robot using local model neural networks". In: *IEEE Robotics & Automation Magazine* 11.3 (2004), pp. 83–90.
- [142] X. Yuan et al. "A novel Mittag-Leffler kernel based hybrid fault diagnosis method for wheeled robot driving system". In: *Computational intelligence and neuroscience* 2015 (2015), p. 65.
- [143] Anders Lyhne Christensen et al. "Fault detection in autonomous robots based on fault injection and learning". In: *Autonomous Robots* 24.1 (2008), pp. 49–67.
- [144] Anders Lyhne Christensen et al. "Automatic Synthesis of Fault Detection Modules for Mobile Robots." In: *AHS. 2007*, pp. 693–700.
- [145] Richard Canham, Alexander H Jackson, and Andy Tyrrell. "Robot error detection using an artificial immune system". In: *Evolvable Hardware, 2003. Proceedings. NASA/DoD Conference on*. IEEE. 2003, pp. 199–207.
- [146] Maizura Mokhtar et al. "A modified dendritic cell algorithm for on-line error detection in robotic systems". In: *2009 IEEE Congress on Evolutionary Computation*. IEEE. 2009, pp. 2055–2062.
- [147] HuiKeng Lau et al. "Adaptive data-driven error detection in swarm robotics with statistical classifiers". In: *Robotics and Autonomous Systems* 59.12 (2011), pp. 1021–1035.
- [148] Nick DL Owens et al. "T cell receptor signalling inspired kernel density estimation and anomaly detection". In: *International Conference on Artificial Immune Systems*. Springer. 2009, pp. 122–135.
- [149] Bojan Jakimovski and Erik Maehle. "Artificial immune system based robot anomaly detection engine for fault tolerant robots". In: *International Conference on Autonomic and Trusted Computing*. Springer. 2008, pp. 177–190.
- [150] Danesh Tarapore et al. "To err is robotic, to tolerate immunological: fault detection in multirobot systems". In: *Bioinspiration & biomimetics* 10.1 (2015), p. 016014.
- [151] Adil Khadidos, Richard M Crowder, and Paul H Chappell. "Exogenous Fault Detection and Recovery for Swarm Robotics". In: *IFAC-PapersOnLine* 48.3 (2015), pp. 2405–2410.
- [152] Alan G Millard, Jon Timmis, and Alan FT Winfield. "Run-time detection of faults in autonomous mobile robots based on the comparison of simulated and real robot behaviour". In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE. 2014, pp. 3720–3725.
- [153] Carlo Pinciroli et al. "ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems". In: *Swarm intelligence* 6.4 (2012), pp. 271–295.

- [154] Carlo Pinciroli et al. "ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics". In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE. 2011, pp. 5027–5034.
- [155] Carlo Pinciroli, Marco Dorigo, and Mauro Birattari. "On the design and implementation of an accurate, efficient, and flexible simulator for heterogeneous swarm robotics systems". In: (2014).
- [156] Francesco Mondada et al. "The e-puck, a robot designed for education in engineering". In: *Proceedings of the 9th conference on autonomous robot systems and competitions*. Vol. 1. LIS-CONF-2009-004. IPCB: Instituto Politécnico de Castelo Branco. 2009, pp. 59–65.
- [157] Álvaro Gutiérrez et al. "Open e-puck range & bearing miniaturized board for local communication in swarm robotics". In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE. 2009, pp. 3111–3116.
- [158] James F Roberts et al. "2.5 D infrared range and bearing system for collective robotics". In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE. 2009, pp. 3659–3664.
- [159] Alvaro Gutiérrez et al. "An open localization and local communication embodied sensor". In: *Sensors* 8.11 (2008), pp. 7545–7563.
- [160] Sandeep Kumar Malu and Jharna Majumdar. "Kinematics, localization and control of differential drive mobile robot". In: *Global Journal of Research In Engineering* (2014).
- [161] Shiladitya Banerjee. "Cell Mechanics From cytoskeletal dynamics to tissue-scale mechanical phenomena". PhD thesis. Syracuse University, 2013.
- [162] Yousef Jamali, Mohammad Azimi, and Mohammad RK Mofrad. "A sub-cellular viscoelastic model for cell population mechanics". In: *PLoS One* 5.8 (2010), e12097.
- [163] Qifeng Chen et al. "Virtual Spring-Damper Mesh-Based Formation Control for Spacecraft Swarms in Potential Fields". In: *Journal of Guidance, Control, and Dynamics* 38.3 (2015), pp. 539–546.
- [164] Mark AC Gill and Albert Y Zomaya. *Obstacle avoidance in multi-robot systems: experiments in parallel genetic algorithms*. Vol. 20. World Scientific, 1998.
- [165] Inaki Navarro and F Matia. "A proposal of a set of metrics for collective movement of robots". In: *Proc. Workshop on Good Experimental Methodology in Robotics*. Seattle, WA, USA., 2009.
- [166] Damien Violeau and Benedict D Rogers. "Smoothed particle hydrodynamics (SPH) for free-surface flows: past, present and future". In: *Journal of Hydraulic Research* 54.1 (2016), pp. 1–26.
- [167] Daniel J Price. "Smoothed particle hydrodynamics and magnetohydrodynamics". In: *Journal of Computational Physics* 231.3 (2012), pp. 759–794.

- [168] Luciano CA Pimenta et al. "Swarm coordination based on smoothed particle hydrodynamics technique". In: *IEEE Transactions on Robotics* 29.2 (2013), pp. 383–399.
- [169] Sheng Zhao, Subramanian Ramakrishnan, and Manish Kumar. "Density-based control of multiple robots". In: *Proceedings of the 2011 American Control Conference*. IEEE. 2011, pp. 481–486.
- [170] Jianping Gou et al. "A new distance-weighted k-nearest neighbor classifier". In: *J. Inf. Comput. Sci* 9.6 (2012), pp. 1429–1436.
- [171] Melvin Gauci et al. "Self-organized aggregation without computation". In: *The International Journal of Robotics Research* 33.8 (2014), pp. 1145–1161.
- [172] Fouzi Harrou et al. "Improved detection of incipient anomalies via multivariate memory monitoring charts: Application to an air flow heating system". In: *Applied Thermal Engineering* 109 (2016), pp. 65–74.
- [173] Fouzi Harrou et al. "Ozone measurements monitoring using data-based approach". In: *Process Safety and Environmental Protection* 100 (2016), pp. 220–231.
- [174] J.E. Jackson and G. Mudholkar. "Control procedures for residuals associated with principal component analysis". In: *Technometrics* 21 (1979), pp. 341–349.
- [175] M. Zhu and A. Ghodsi. "Automatic dimensionality selection from the scree plot via the use of profile likelihood". In: *Computational Statistics & Data Analysis* 51 (2006), pp. 918–930.
- [176] S.J. Qin. "Statistical process monitoring: Basics and beyond". In: *Journal of Chemometrics* 17.8/9 (2003), pp. 480–502.
- [177] F. Harrou et al. "PLS-based EWMA fault detection strategy for process monitoring". In: *Journal of Loss Prevention in the Process Industries* 36 (2015), pp. 108–119.
- [178] J.M. Lucas and M.S. Saccucci. "Exponentially weighted moving average control schemes: properties and enhancements". In: *Technometrics* 32.1 (1990), pp. 1–12.
- [179] F. Harrou and M.N. Nounou. "Monitoring linear antenna arrays using an exponentially weighted moving average-based fault detection scheme". In: *Systems Science & Control Engineering: An Open Access Journal* 2.1 (2014), pp. 433–443.
- [180] P.A. Morton et al. "The application of statistical process control charts to the detection and monitoring of hospital-acquired infections". In: *Journal of quality in clinical practice* 21.4 (2001), pp. 112–117.
- [181] F. Harrou, M.N. Nounou, and H.N. Nounou. "A statistical fault detection strategy using PCA based EWMA control schemes". In: *9th Asian Control Conference (ASCC)*. IEEE. 2013, pp. 1–4.

- [182] Farid Kadri et al. "Seasonal ARMA-based SPC charts for anomaly detection: Application to emergency department systems". In: *Neurocomputing* 173 (2016), pp. 2102–2114.
- [183] D. C. Montgomery. "Introduction to statistical quality control". In: *John Wiley & Sons, New York* (2005).
- [184] E.S. Page. "Continuous inspection schemes". In: *Biometrika* 41.1/2 (1954), pp. 100–115.
- [185] F. Harrou, Y. Sun, and M. Madakyaru. "Kullback-Leibler distance-based enhanced detection of incipient anomalies". In: *Journal of Loss Prevention in the Process Industries* 44 (2016), pp. 73–87.
- [186] Isaac J Schoenberg. "Contributions to the problem of approximation of equidistant data by analytic functions". In: *IJ Schoenberg Selected Papers*. Springer, 1988, pp. 3–57.
- [187] Alan FT Winfield, Christian Blum, and Wenguo Liu. "Towards an ethical robot: internal models, consequences and ethical action selection". In: *Conference Towards Autonomous Robotic Systems*. Springer. 2014, pp. 85–96.
- [188] Alan Millard. "Exogenous Fault Detection in Swarm Robotic Systems". PhD thesis. University of York, 2016.