ANTICIPATION IN COLLECTIVE MOTION OF ROBOT SWARMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İHSAN CANER BOZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

DECEMBER 2021

Approval of the thesis:

ANTICIPATION IN COLLECTIVE MOTION OF ROBOT SWARMS

submitted by **İHSAN CANER BOZ** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Mehmet Ali Sahir Arıkan
Head of Department, **Mechanical Engineering** ______________

Assist. Prof. Dr. Ali Emre Turgut
Supervisor, **Mechanical Engineering, METU** ______________

Prof. Dr. Cristián Huepe
Co-supervisor, **Northwestern Inst. on Complex Systems, USA** ______________

Examining Committee Members:

Assoc. Prof. Dr. Erol Şahin
Computer Engineering, METU ______________

Assist. Prof. Dr. Ali Emre Turgut
Mechanical Engineering, METU ______________

Assoc. Prof. Dr. Ahmet Buğra Koku
Mechanical Engineering, METU ______________

Assoc. Prof. Dr. Ender Yıldırım
Mechanical Engineering, METU ______________

Assist. Prof. Dr. Kutluk Bilge Arıkan
Mechanical Engineering, TED University ______________

Date: 08.12.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: İhsan Caner Boz

Signature :

ABSTRACT

ANTICIPATION IN COLLECTIVE MOTION OF ROBOT SWARMS

Boz, İhsan Caner
M.S., Department of Mechanical Engineering
Supervisor: Assist. Prof. Dr. Ali Emre Turgut
Co-Supervisor: Prof. Dr. Cristián Huepe

December 2021, 85 pages

Recent technological advancements made the implementation of swarms of UAVs possible. As they face more complex situations each day, collective motion models rise in importance. In this thesis, a collective motion model for self-propelled agents with anticipative action is given. It is shown that using anticipated positions in the attraction-repulsion mechanism brings a new interaction term that depends on velocities. The noise-induced order-disorder phase transition of this model is compared with two other well-known collective motion models: Active-Elastic model and Vicsek model. The comparison shows that anticipation aligns the headings of the agents. Therefore, the anticipation horizon can be adjusted to help robot swarms accomplish certain tasks. For example; better braking performance and preventing pileup in case of a sudden stop, or squeezing through a narrow passage without losing order. Sensors used in the robot swarms are mostly position-based. Since velocity information is required for the anticipative model, a Kalman Filter that takes equations of motion into consideration is designed to improve the position measurements and supply velocity and heading information. Particle-based simulations are done for the proposed model with the designed filter. Then, the model and the filter are implemented in the Crazyswarm platform, which is used for flying a swarm of Bitcraze Crazyflie 2.x quadcopter UAVs. In the experiments, control commands of the agents are created by their own Kalman Filter, which uses noisy measurements and the motion model information. Experiments with the real UAVs show that the swarm is able to pass through a narrow passage with the proposed anticipative motion model.

Keywords: collective motion, swarm robotics, anticipatory systems, Kalman filter

ÖZ

SÜRÜ ROBOTLARIN MÜŞTEREK HAREKETİNDE BEKLENTİ

Boz, İhsan Caner
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Ali Emre Turgut
Ortak Tez Yöneticisi: Prof. Dr. Cristián Huepe

Aralık 2021 , 85 sayfa

Son teknolojik gelişmeler İHA sürülerinin uygulanabilmesini mümkün kılmıştır. Sürülerin her geçen gün daha karmaşık durumlarla karşılaşması, müşterek hareket modellerinin önemini arttırmaktadır. Bu tezde, kendinden hareketli ajanlar için beklentili eylemi olan müşterek hareket modeli verilmiştir. Çekme-itme işleyişinde beklenen konumu kullanmanın hızlara bağlı yeni bir etkileşim terimi getirdiği gösterilmiştir. Modelin gürültü ile uyarılmış düzen-düzensizlik faz geçişi, şu iki iyi bilinen müşterek hareket modeli ile karşılaştırılmıştır: Active-Elastic model ve Vicsek model. Karşılaştırma, beklentinin ajanların yönlerini hizaladığını göstermiştir. Böylece, beklenti ufku ayarlanarak robot sürülerinin bazı görevleri tamamlamasına yardım edilebilir. Örnek olarak, daha iyi fren performansı ile zincirleme kazaların önlenmesi, veya dar bir geçitten sıkışarak ve düzeni kaybetmeden geçilebilmesi verilebilir. Beklentili modelde hız bilgisi gerektiği için, pozisyon ölçümlerini iyileştiren, hız ve yön bilgisi sağlayan, ve hareket denklemlerini dikkate alan bir Kalman Filtresi tasarlanmıştır. Önerilen model ve tasarlanan filtre ile parçaçık bazlı benzetimler yapılmıştır. Daha sonra, model ve filtre, Bitcraze Crazyflie 2.x dört pervaneli helikopter İHA sürüsünü uçurmaya yarayan Crazyswarm ortamında uygulanmıştır. Deneylerde, ajanların kontrol komutları kendi Kalman Filtreleri tarafından gürültülü ölçümler ve hareket modeli bilgisi kullanılarak üretilmiştir. Gerçek İHAlar ile yapılan deneyler sürünün dar bir geçitten önerilen beklentili model ile geçebildiğini göstermiştir.

Anahtar Kelimeler: müşterek hareket, sürü robotiği, beklentili sistemler, Kalman filtresi

To Merve.

ACKNOWLEDGMENTS

I would like to thank my supervisor Ali Emre Turgut and my co-supervisor Cristián Huepe for their support and for being always available whenever I needed their invaluable guidance. Experiments in this thesis are conducted in the METU ROMER UAV Laboratory. Many thanks to Roketsan Missiles Inc. for supporting my graduate study. I would like to express my sincerest gratitude to my wife Merve and my family for their unconditional love and support.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
AE	Active-Elastic
AEA	Active-Elastic-Anticipation
API	Application Programming Interface
EKF	Extended Kalman Filter
ER	Erdős–Rényi
GPS	Global Positioning System
LED	Light Emitting Diode
LIDAR	Laser Imaging Detection and Ranging
NN	Nearest Neighbor
SF	Scale-Free
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
UWB	Ultra-Wide-Band
VN	Vicsek-Network

CHAPTER 1

INTRODUCTION

Complex systems are composed of many interacting systems, and their behavior may not be predicted by analyzing only the microscopic interactions. Cellular automata are an example of complex systems. Initial states and interaction rules are defined at the start, and there may not be a closed-form solution for finding the future states of the system. Collective motion is another example of complex system behavior, which is widely observed in living organisms. Members of the group act upon the limited information available to them, yet the outcome is an organized group that moves collectively and reacts to changes in the environment.

Collective motion is a popular research topic since it both helps to explain the phenomenon seen in nature, and gives a way to move physical agents in an orderly fashion in the real world. Most of the collective motion models use attraction-repulsion or heading alignment mechanisms, or a mixture of both. Mainly, these mechanisms depend on the current state of the swarm. However, anticipation is an important concept in decision-making. Following Rosen's definition in [2], anticipatory systems can estimate their future states with the use of a predictive model, and uses these estimates in their decisions in the current time. For instance, [3] shows how the anticipated position of a moving object is created at the retina level, so that the delays associated with the slow transduction of information to the brain do not affect the animal's ability to capture moving objects.

Arguably, traffic is a form of collective motion, and many facts about the importance of anticipation can be derived from it. For instance, brake lights are activated when a car decelerates, and the driver at the back anticipates that distance with the front car will decrease and presses the brake pedal. Thus, with the help of the brake light, braking action of the rear car will start before the cars get close. In [4], 343 taxis were equipped with a third brake light that pulses more frequently as the deceleration amount increases. Data gathered from the 10-month long experiment show that taxis with third brake light had a 60.6% reduction in the rear-end collision rate. It can be argued that taxi drivers were sensitive to the new arrangements and drove cautiously because of the Hawthorne effect [5], which refers to the tendency of the participants in an experiment to behave differently due to being observed. However, it is claimed that this effect does not apply here because collisions are due to the cars that follow the taxis, which are not the participants in the experiment. Also, the frontal collision rate of the taxis with the third brake light was the same as the control group, which means that the control and the experimental groups behave similarly. In [6], drivers with varying degrees of experience participated in driving simulations with different scenarios, such as chain-breaking due to a slow tractor or a vehicle cutting in front. It is reported that experienced drivers make more anticipatory actions, which help to prevent an accident.

With regard to these examples, anticipation is an interesting concept to study in the scope of collective motion. Active-Elastic model [7][8] is a well-studied collective motion model that uses only relative positions in the attraction-repulsion mechanism. Anticipation can be implemented into this model by using anticipated positions, instead of the current positions. In this thesis, this novel model is proposed as Active-Elastic-Anticipation (AEA) model. This implementation brings a new interaction term that depends on the relative velocities. Thus, there is a new parameter to adjust, in order to achieve better performance in certain scenarios. For example, similar to the brake lights, the relative velocity dependent term brings better braking performance and prevents multi-robot collisions. Moreover, robots can adjust their orientation depending on the anticipated positions and squeeze through a narrow passage without losing order.

Anticipative interaction is introduced to the AEA model with a linear damper. Since dampers are used in the model, relative velocity information of the agents are required. Acquiring this information may not be an easy task, since most measurement systems for UAVs are position-based; such as Ultra-wideband positioning systems, motion capture systems, or LIDARs. Acquiring relative velocity information from the position measurements by differentiation is also not feasible unless the measurement noise is exceptionally low. Therefore, using a method to improve the information at hand is essential. Extended Kalman Filter (EKF) is used for fusing sensor measurements and system model to get better estimates of the relative velocity, relative position and orientation of the agents. There are many studies on collective localization, in which agents communicate with each other for improved state estimation. However, in this study, it is assumed that no communication exists between agents and only on-board sensors are utilized. This assumption might be more suitable to explain the behavior of simpler organisms.

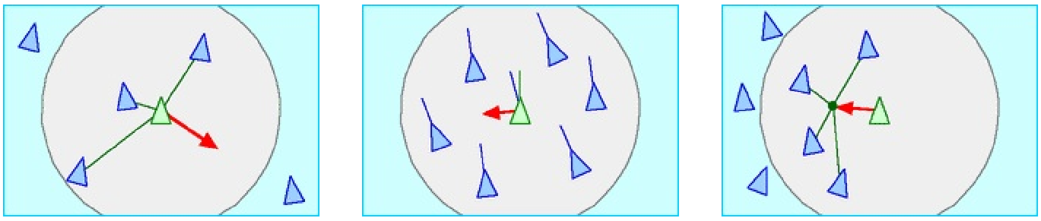
Noise and order characteristics of the proposed model are analyzed and compared with other well-known collective motion models. An EKF design that takes motion equations into consideration is given. The designed filter uses self-position, self-orientation, and relative positions of the neighbors, which are readily available in UAV systems. Performance of the model with the designed filter is evaluated with particle-based simulations. Then, model and the filter is implemented in the Crazyswarm platform, which is used to simulate and fly Bitcraze Crazyflie quadcopter UAVs. Implementation is verified with the simulation mode and then real flight tests are conducted with Crazyflie 2.x UAVs.

This thesis has six chapters, including this first Introduction chapter. In Chapter 2, literature survey of collective motion models, anticipation in collective motion models and state estimation in multi-agent systems are given. In Chapter 3, the definition of the Active-Elastic-Anticipation model is given. Also, the design of the Extended Kalman Filter for this model is explained there. In Chapter 4, details about the experimental setups and results of the conducted analyses are given. In Chapter 5, the methodology and results are discussed; possible improvements and future work are also given. Lastly, in Chapter 6, all of the work given in the thesis is concluded.

CHAPTER 2

LITERATURE SURVEY

Collective motion is observed widely in living organisms, such as birds [9], fish [10] and bacteria [11]. It is also observed in inanimate objects when external excitation is applied [12] [13]. Creating artificial systems that can achieve collective motion based on simple interactions is a popular research area. One of the oldest and well-known collective motion model is Reynolds Boid model [14]. Reynolds model has 3 rules: collision avoidance, velocity matching, and flock centering. The model successfully simulates flocks of bird-like objects called boids. The rules of the model are illustrated in the Figure 2.1.



(a) Separation

(b) Alignment

(c) Cohesion

Figure 2.1: Visual depiction of the Reynolds model rules. The gray area shows the local interaction region of the focal agent. (a) Separation: Boids avoid collision with each other. (b) Alignment: Boids match their velocity to the neighboring boids. (c) Cohesion: Boids try to move towards the center of the boids in their locality. Reprinted from [1]

Another well-known collective motion model that influenced many works in the field is the Vicsek model [15]. Vicsek model is an orientation-based model, each agent adjusts its heading to the average heading of the agents in its locality. Agents are self-propelled and have the same speed. In a similar manner, the Cucker-Smale model [16] [17] shows that agents can reach a consensus in the direction of movement by aligning their velocity vectors with their neighbors. Unlike the Vicsek model, strength of the interactions in the Cucker-Smale model depends on the proximity of the agents. There are many modified versions of both the Vicsek model and the Cucker-Smale model. In the original Vicsek model, the distance between the agents will grow if they are placed in an infinite arena, especially under noise. In [18], an attraction-repulsion term is added to aid this problem. [19] added a repulsion-only term to the Cucker-Smale model, to avoid collision between the agents. There is also a model based solely on pairwise attraction-repulsion mechanism, called the Active-Elastic (AE) model [7][8]. In the AE model, agents know only the position of their neighbors relative to themselves.

Following Rosen's definition in [2], anticipatory systems can estimate their future states with the use of a predictive model and use these estimates in their decisions in the current time. Anticipation is an important concept for correct decision-making and task execution. For instance, in [3] authors showed how the anticipated position of a moving object is created at the retina level, so that delays associated with slow transduction of information to the brain do not affect the animal's ability to capture moving objects. Most of the well-known collective motion models use only the current state of the group but there are studies on anticipation in collective motion.

As described in the previous chapter, anticipation is a useful concept and there are studies on anticipation in collective motion. In [20], an angular velocity dependent term is added to the Vicsek model. Hence, this new model uses the expected angular positions in a later time, rather than the current angular positions. Depending on the coefficient of the angular velocity term, swarming or spinning states are observed. Therefore, the swarm can be ordered to move collectively or stay around a point without stopping altogether by changing the coefficient of the anticipative term. [21] is a distinct paper that proposes a discrete, lattice-based model with asynchronous position update rules. Even though the model is not based on an explicit alignment rule, position update of the agents depends on the anticipated positions of their neighbors. The model can successfully achieve collective motion, and results are in line with the empirical study conducted on soldier crabs. In [22], it was demonstrated that inclusion of anticipation converts the simple harmonic oscillator to a damped harmonic oscillator. Thus, it was shown that anticipation in an attraction-repulsion based model can be modeled by adding a linear damper in parallel to the spring. Also, energy dissipation brought by the damping term is analyzed. A thorough mathematical analysis of anticipation in systems reacting to radial potentials can be found in [23]. The importance of alignment brought by the anticipation is demonstrated. [24] proposes a model called the Active-Elastic-Alignment model, which is an extension of the AE model. In the Active-Elastic-Alignment model, springs are connected to the arms that extend from the center of the agents. It was shown that as the length of the arms increase, interaction of the agents becomes dominantly alignment-based. This paper has great importance for this thesis, since the model proposed in this thesis is also based on the AE model, and extending arms are homologous to using anticipated positions.

In a real-world implementation of a collective motion model, agents rely on noisy sensor measurements or communication to acquire required information. In general, obtaining the heading information of other agents is not an easy task. [25] is a study on a communication strategy for heading control of a swarm of physical robots. Robots used in the experiments are the Foot-bot robots that were developed within the Swarmanoid project [26]. In this study, robots rely on a communication system to obtain the heading and relative positions of the neighboring robots. [27] is an extensive study on disruptive effects which arise in the realization of collective motion models. Real quadcopter UAVs equipped with Global Positioning System (GPS) are used in the study. UAVs communicate their position and velocity measurements to be able to calculate the control outputs. [28] presents a computer vision algorithm that provides range and bearing for robot swarms. A marker with specific features is designed to be placed on each robot in the swarm. The appearance of this marker makes range and bearing calculations possible. In these studies, either communication or a specific system is used to find the heading of the neighboring robots.

There are studies for improving state estimation in multi-agent systems by fusing the information coming from multiple agents. In [29], a Kalman Filter in decentralized form is used to fuse the measurements from different agents to improve the localization accuracy. This method requires the communication of sensor measurements. In

[30], also a Kalman Filter that improves the localization accuracy is used but only state estimates of the agents are broadcasted. In [31], 2 different methods for relative localization using only relative distances between agents are proposed but again these methods also require communication between agents.

In this thesis, a novel collective motion model with anticipative action is proposed and the model is named as Active-Elastic-Anticipation model. An Extended Kalman Filter (EKF) is designed to estimate the states of the focal agent and its neighbors, which are used in the equations of motion of the proposed model. The contribution of the thesis is two-fold. First, anticipation is added to the well-known AE model by simply adding a linear damper in parallel to the linear spring that exists in the model. The noise-induced phase transition of the model is analyzed and compared with the AE model and the Vicsek model. Second, an EKF design that estimates the heading information of the neighbors without communication or any specific arrangement is given. Estimation performance is improved by using motion equations of the model in the filter design.

CHAPTER 3

METHODOLOGY

3.1 Background

In this section, two well-known Collective Motion models that are important in the development of the Active-Elastic-Anticipation model are introduced: Active-Elastic model and Vicsek model. Active-Elastic model interactions depend on positions, whereas Vicsek model interactions depend on headings. This fundamental distinction is best visualized with the phase transition diagram, which is described in the following subsections.

3.1.1 Interaction Networks

In the study of complex systems, the dynamics of multiple interacting components are typically analyzed using one of two different modeling approaches: agent-based or network-based. If the components can be characterized as particles moving in a physical or abstract space and interactions depend on their positions, the system is best described using agent-based approaches. Alternatively, if the components can be characterized as nodes in a network, with internal states that evolve by interacting through their connections, it is best described using network-based approaches. Both types of models have been widely used to analyze the self-organized collective dynamics observed in a broad range of complex systems.

In this thesis, interactions between the agents are described with static interaction networks. The main interaction topologies used in this thesis are Nearest-neighbor (NN) networks, random Erdős–Rényi (ER) networks, and random Scale-Free (SF) networks. Each type has a distinct degree distribution, as shown in the plots of the number of nodes with a given number of connections presented in Figure 3.1. These networks are later combined to study the dynamics of the model when interacting through different superpositions of their topologies.

In order to compare the dynamics resulting from various interaction topologies under equivalent conditions, the total number of agents (nodes) N and the total number of links K should be kept constant over all connectivity structures. This also ensures that the average number of connections per node K/N is constant, although other statistical quantities may change as a result of the different degree distributions of the networks.

3.1.1.1 Nearest-neighbor network

In the NN network, agents are connected to their first neighbors in their starting position. K value depends on the given N and starting positions. Agents on the corners and sides may have a lower number of connections, but most of the agents have the same number of connections.

3.1.1.2 Erdős-Rényi network

Starting with N nodes with no connection, a pair of nodes (i, j) is randomly selected and connected, unless $i = j$ or they are already connected. This process is continued until there are K links. Then, a breadth-first search is made to check network connectivity. If all agents are visited in the search, then the network is connected; there are no disconnected sub-networks. If the network is not connected, it is discarded and the process starts over.

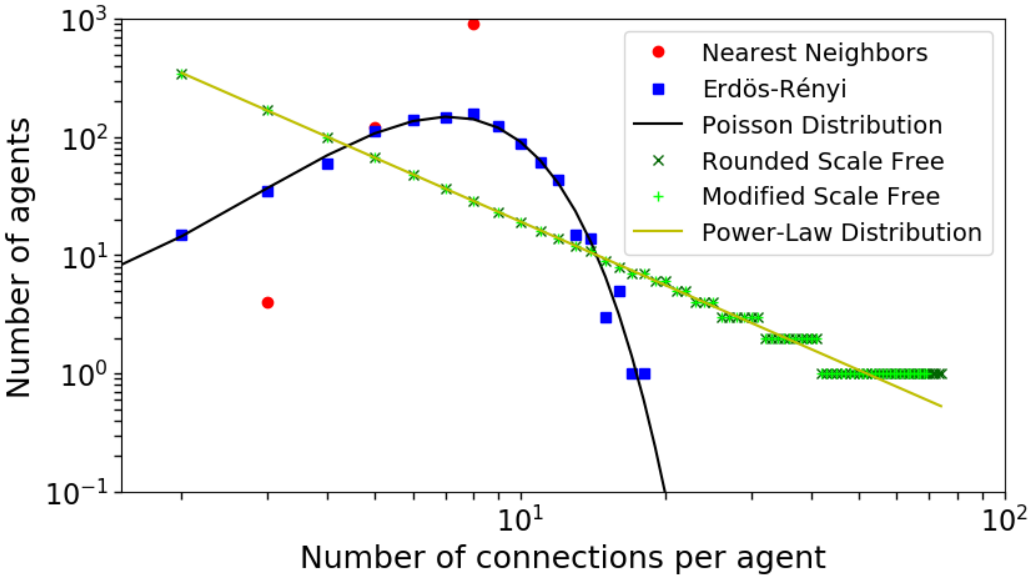


Figure 3.1: Degree distributions of the three types of interaction networks implemented in this thesis (Nearest Neighbour, Erdős-Rényi, and Scale-Free) for a regular square lattice system with $9 \times 9 = 81$ agents and 272 connections. Each plot shows the number of agents that have a given number of interactions with other agents. In the Nearest Neighbour network (red dots), connections are determined by the number of immediate neighbors in a square lattice, here 3, 5, or 8 for agents in the corners, sides, or bulk, respectively. In the Erdős-Rényi network (blue squares), the degree distribution must be Poissonian, as approximated by the implemented case displayed. Finally, in Scale-Free networks ($\times$ and $+$ signs) the degree distribution must follow a power-law, here well approximated by the Modified Scale Free connectivity case (which was generated by manually correcting the Rounded Scale Free case), despite the finite and discrete nature of the system.

3.1.1.3 Scale-Free network

In the SF topology, linked agents are selected at random, as in the ER topology, but the resulting degree distribution must follow a power law of the form

$$n_k = C\,k^b. \tag{3.1}$$

Here, n_k is the number of agents with k connections, C is a constant prefactor, and b is the law's exponent, which is always negative [32, 33].

In a finite and discrete system, k can only have integer values within a certain range, which is defined to be from $k_{\min}$ to $k_{\max}$. The minimum number of connections per agent, $k_{\min}$, is set to be 2 in all created networks. At the other end of the distribution, another restriction is $k_{\max} \leq N-1$, because self-connections or more than one connection between two nodes are not allowed. Finally, in order to generate SF networks with predetermined N and K values, the following two expressions must be satisfied:

$$\sum_{k=k_{\min}}^{k_{\max}} n_k \;=\; N \tag{3.2}$$

$$\sum_{k=k_{\min}}^{k_{\max}} n_k\,k \;=\; K. \tag{3.3}$$

In principle, for any given exponent b a solution can be found for C and $k_{\max}$ that satisfies both expressions. In practice, however, the problem has additional constraints that make it slightly more complicated. Indeed, the n_k values resulting from equation (3.1) must be rounded to the nearest integer and they cannot be smaller than 1. Thus, both constraints can be imposed by modifying the degree distribution, redefining it as

$$n_k = \begin{cases} \mathrm{round}(C\,k^b) & \text{if } \; C\,k^b \geq 1 \\ 1 & \text{if } \; C\,k^b < 1, \end{cases} \tag{3.4}$$

where $\mathrm{round}(\cdot)$ is the function that rounds to the nearest integer.

The problem is now reduced to finding values for C and $k_{\max}$ that satisfy equations (3.2) and (3.3), using the n_k function defined in (3.4). Given that $k_{\max}$ and all n_k values must be integers, however, there is typically no exact solution for this system. Thus, it is solved as an optimization problem, finding the C and $k_{\max}$ values that minimize the difference between the left- and right-hand sides of equations (3.2) and (3.3). The corresponding objective function J is defined as the weighted sum of the squared error with respect to the target N and K values, which is defined as N_{tgt} and K_{tgt}. Thus, the objective function is defined as

$$J = W_N(N_{\text{opt}} - N_{\text{tgt}})^2 + W_K(K_{\text{opt}} - K_{\text{tgt}})^2 \tag{3.5}$$

Here, N_{opt} and K_{opt} are the optimization variables while W_N and W_K are the manually tuned weights. To favor the convergence to the correct number of agents, W_N is chosen as 10 and W_K as 1. Results do not depend significantly to the exact choice of these values. Figure 3.1 presents an example of the resulting n_k values, labeled Rounded Scale Free. Finally, $k_{\max}$ is increased or decreased as needed to exactly match the required total number of nodes N and then a few connections may be added or removed by hand to have exact K in total, so that all constraints are met. An example of the final n_k values is displayed in the curve labeled Modified Scale Free in Figure 3.1. It confirms that the final degree distribution still follows an approximate power law. It is observed that Modified Scale Free is almost identical to the Rounded Scale Free distribution, showing that the required final manual adjustments are minimal. It is also noted that, for the $b = -2$ case displayed here, $k_{\max}$ is low enough for the distribution not to include a long flat region with $n_k = 1$ at high k values. This is one of the reasons why b is set as -2 in most simulations below.

Once the exact degree distribution is computed, each node is associated with the corresponding number of links at random, starting from the nodes with the highest degree. As in the ER case, the network is checked for full connectivity.

3.1.1.4 Network superposition method

In this thesis, combinations of the network topologies defined above are considered. To this end, a superposition protocol that interpolates between a NN network and either an ER or an SF random network is devised, as a function of a topological control parameter $p \in [0, 1]$. For $p = 0$, the exact NN structure is recovered; and for $p = 1$, one of the two random networks (ER or SF) with the same number of nodes and links is obtained. In addition, by setting $0 < p < 1$, it is possible to interpolate between NN and ER topologies or between NN and SF topologies. The proposed superposition protocol is defined as follows. First, a NN network of a given size is set up, which determines the values of the total number of nodes N and the total number of connections K. Then, a realization of one of the two types of random networks (ER or SF) with N nodes and K links is generated. Finally, these networks are combined by first deleting pK links from the NN network and $(1-p)K$ links from the ER or SF network at random, and then superimposing the two resulting structures. Resulting network is checked for full connectivity.

3.1.2 Active-Elastic Model

Active-Elastic model is a position-based collective motion model introduced in [7][8]. Agents move in 2D plane and they are connected to each other with linear springs. Equations of motion for the Active-Elastic model are given in two following equations:

$$\dot{\vec{x}}_i(t) = v_0\,\hat{n}_i(t) + \alpha\left[\vec{F}_i(t)\cdot\hat{n}_i(t)\right]\hat{n}_i(t) \tag{3.6}$$

$$\dot{\theta}_i(t) = \beta\left[\vec{F}_i(t)\cdot\hat{n}_i^{\perp}(t)\right] + \eta\,\xi_i(t). \tag{3.7}$$

Here, $\hat{n}_i(t)$ and $\hat{n}_i^{\perp}(t)$ are the unit vectors that point parallel and perpendicular to the heading of the agent i at time t, respectively. α and β are coefficients that relate force to linear and angular velocities, respectively. v_0 is the self-propulsion speed. ξ is a uniform distributed random variable whose boundaries are $-1/2$ and $1/2$. η is the noise intensity coefficient and is used to control the intensity of applied noise. Force vector calculation is given in Equation 3.8 as:

$$\vec{F}_i(t) = \sum_{j\in S_i} -\kappa(|\vec{r}_{ij}| - l_{ij})\frac{\vec{r}_{ij}}{|\vec{r}_{ij}|} \tag{3.8}$$

where $\vec{r}_{ij} = \vec{x}_j - \vec{x}_i$ and κ is the spring constant. The set S_i contains all agents that agent i interacts with. Usually, these differential equations are integrated with Euler's method with a time step Δt.

3.1.3 Vicsek Model

In the Vicsek model, agents move at a constant speed. The focal agent adjusts its heading to the average headings of the agents it is interacting with including its own heading. Agents that are in a given range interact with each other, thus the interaction network changes dynamically. However, in this thesis, only static networks are used, thus the interaction links between agents do not change. Therefore, there is no need to simulate the change in the positions of the agents. This version of the Vicsek model is referred to as Vicsek-Network (VN) model in [34]. Also, the same implementation was introduced in [35] as Vectorial Network model.

Orientation update equation of the VN model is given in Equation 3.9,

$$\theta_i[k+1] = \text{angle}\left(\sum_{j \in S_i} \hat{n}_j[k]\right) + \eta \xi_i[k] \tag{3.9}$$

where $\hat{n}_j[k]$ is the unit vector of agent j at time step k that points towards its heading direction. The set S_i contains all agents that agent i interacts with. Therefore, angle$(.)$ function with the sum of unit vectors in the argument gives the average orientation in the S_i. The noise term, ξ is a uniform distributed random variable, with the boundaries of $-1/2$ and $1/2$. η can be adjusted to obtain desired noise intensity. $\eta = 0$ corresponds to no noise, whereas $\eta = 2\pi$ corresponds to fully random motion.

3.1.4 Phase Transition

Phase transition diagrams are usually linked with an order parameter and they depict different phases of a system. In a simplified way, if the transition has a discontinuity in the first derivative, it is called first-order transition. Water to ice transition is an example of the first-order phase transition. Likewise, if the transition has a discontinuity in the second derivative, it is called second-order transition. Paramagnetic to ferromagnetic phase transition is an example of the second-order phase transition. The type of the transition can be inferred from the phase transition diagram and the noise-induced phase transition behavior of a collective motion model can be used for classification.

3.2 Active-Elastic-Anticipation Model

Active-Elastic-Anticipation (AEA) model is the novel model and the main contribution of this thesis. This model presents a way to add anticipation to the AE model, by using the expected positions instead of the current positions.

Taylor Series expansion can be used to find an approximate value of the position in a future time. Expansion of the anticipated position is given in Equation 3.10.

$$x(t+\tau) = x(t) + \tau\dot{x}(t) + \tau^2\frac{\ddot{x}(t)}{2!} + \tau^3\frac{\dddot{x}(t)}{3!} + \dots \tag{3.10}$$

The first-order extrapolation of the current position can be obtained by neglecting the higher-order terms. Approximate position after time τ, or anticipated position by the agent, is given in Equation 3.11.

$$\begin{aligned} x(t+\tau) &\approx x(t) + \tau\dot{x}(t), \\ &\approx x(t) + \tau v(t) \end{aligned} \tag{3.11}$$

It can be seen that using the anticipated position brings a new term that is proportional to velocity. By choosing anticipation time amount as τ, the interaction force of the AE model transforms into Equation 3.12. The AE model force equation was given in Equation 3.8.

$$\begin{aligned} F_{ij} &= -\kappa(|\vec{r}_{ij}| + \tau|\vec{v}_{ij}| - l_{ij}) \\ &= -\kappa(|\vec{r}_{ij}| - l_{ij}) - \kappa\tau|\vec{v}_{ij}| \end{aligned} \tag{3.12}$$

This new force term that depends on velocity can be modeled with a linear damper with a damping coefficient of $\kappa\tau$. Interaction force applied on agent i for the AEA model is given in equation 3.13, where b equals to $\kappa\tau$.

$$\vec{F}_i(t) = \sum_{j\in S_i}(-\kappa(|\vec{r}_{ij}| - l_{ij}) - b|\vec{v}_{ij}|)\frac{\vec{r}_{ij}}{|\vec{r}_{ij}|} \tag{3.13}$$

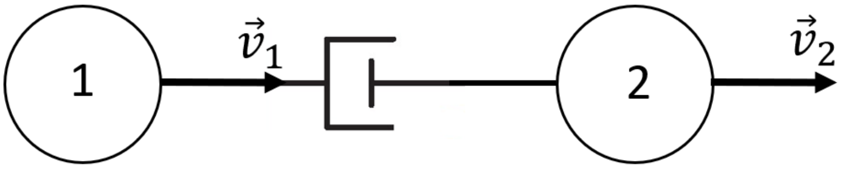


Figure 3.2: Two agents connected with a damper

This proposed model is most similar to the Active-Elastic-Alignment model in [24]. There are two main differences in the force calculation between the model proposed in this thesis and the Active-Elastic-Alignment model. First of all, in this work, the anticipated position is found by linear extrapolation of the current position with a fixed time length. In the Active-Elastic-Alignment model, the interaction point has a fixed distance from the center of the agent, which is equivalent to the anticipated position. Since the velocity of the agents can change, fixed-time anticipation may differ from fixed-length anticipation. Secondly, in the Active-Elastic-Alignment model, interaction elements are connected to the extending arms. Therefore, the direction of the force depends on the anticipated positions. In this work, direction of the force is calculated with the current positions.

In the AE model, positions and orientations of the agents follow two equations of motion that were given in Equations 3.6 and 3.7. It can be seen that the velocity equation depends on the force, and force in the AEA model is also dependent on velocity due to the damper term. An example is given to demonstrate how this can pose a problem. Suppose two agents are connected only with a damper, as shown in Figure 3.2. For ease of calculations, agents are restricted to move in one dimension, and the spring is removed. Therefore, orientation dynamics are removed and velocity update equations for each agent are given in Equation 3.14.

$$\begin{aligned} \dot{x}_1 &= v_0 - \alpha b(\dot{x}_1 - \dot{x}_2) \\ \dot{x}_2 &= v_0 - \alpha b(\dot{x}_2 - \dot{x}_1) \end{aligned} \tag{3.14}$$

This system has only one solution, where velocity of the agents are equal to the self-propulsion speed, $\dot{x}_1 = \dot{x}_2 = v_0$. Therefore, the velocity of the agents can not change and agents would not be interacting with each other through the damper as intended.

The system can be discretized with the zero-order hold method to evaluate how the velocity update equation would behave in the simulations. The discrete state-space representation of the system is given in Equation 3.15.

$$\begin{bmatrix} v_1^{k+1} \\ v_2^{k+1} \end{bmatrix} = \begin{bmatrix} -\alpha b & \alpha b \\ \alpha b & -\alpha b \end{bmatrix} \begin{bmatrix} v_1^{k} \\ v_2^{k} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} v_0 \tag{3.15}$$

For the system to be stable, magnitude of the eigenvalues of the state matrix should be smaller than one (or poles of the system should be inside the unit circle). Therefore, as seen in Equation 3.16, αb product should be smaller than 0.5. Agents change their velocity depending on the changes in the relative velocity. If the reaction is bigger than the perturbation, which means the magnitude of the eigenvalue is bigger than 1, perturbations grow.

$$\text{eig}\begin{pmatrix} -\alpha b & \alpha b \\ \alpha b & -\alpha b \end{pmatrix} = \{-2\alpha b, 0\} \tag{3.16}$$

This method works as long as b value is chosen to be smaller than the stability threshold, for a given α value. This puts a tight limit on the values that can be chosen for the damper coefficient.

In short, the AE model agents have no inertia and their velocity can change instantaneously. The velocity update equation depends on the velocities when the damper is introduced to the model, and the stability problem mentioned above occurs.

The proposed solution to this problem is to include inertia in the model. A proportional controller, given in Figure 3.3, is used to control the velocity of the agents in the AEA model. Agents can not reach their desired velocity (v_d) instantly, but an error signal is formed from the desired and the current velocities. The amount of acceleration depends on this error signal and the proportional gain. This kind of approach can be found in multi-agent motion model studies. [36] proposes a model for microscopic traffic flow, called the Optimal Velocity model. In this model, each agent accelerates proportional to the difference between the target and the current velocity, which is also a proportional control scheme.

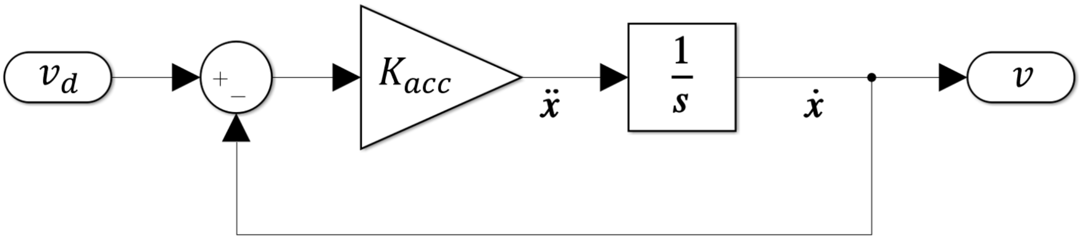


Figure 3.3: Proportional velocity controller used in the AEA model

The effect of the proportional velocity controller on the increase of the margin for stability is demonstrated with an example. The new differential equation governing the motion of the first agent in Figure 3.2 is given in Equation 3.17.

$$\begin{aligned} v_d &= v_0 - \alpha b(\dot{x}_1 - \dot{x}_2) \\ \ddot{x}_1 &= K_{acc}(v_d - \dot{x}_1) \end{aligned} \tag{3.17}$$

The discrete version of the Equation 3.17 is given in Equation 3.18.

$$v_1^{k+1} = v_1^k + K_{acc}\left(v_0 - \alpha b(v_1^k - v_2^k) - v_1^k\right)\Delta t \tag{3.18}$$

The discrete state-space representation of the system given in Figure 3.2 is given in Equation 3.19.

$$\begin{aligned} \begin{bmatrix} v_1^{k+1} \\ v_2^{k+1} \end{bmatrix} = &\begin{bmatrix} (1 - K_{acc}\alpha b\Delta t - K_{acc}\Delta t) & K_{acc}\alpha b\Delta t \\ K_{acc}\alpha b\Delta t & (1 - K_{acc}\alpha b\Delta t - K_{acc}\Delta t) \end{bmatrix} \begin{bmatrix} v_1^k \\ v_2^k \end{bmatrix} \\ &+ \begin{bmatrix} 1 \\ 1 \end{bmatrix} K_{acc} v_0 \Delta t \end{aligned} \tag{3.19}$$

The eigenvalues of the state matrix are:

$$1 - K_{acc}\Delta t$$
$$1 - 2K_{acc}\alpha b\Delta t - K_{acc}\Delta t$$

There are multiple parameters that affect the eigenvalues. If Δt is chosen as 0.1, the same as the previous studies on the AE model, then, from the first eigenvalue, it can be seen that proportional gain K_{acc} should not be larger than 20 for the system to be stable. The upper bound for αb can be extracted from the second eigenvalue. By choosing K_{acc} as 1 for instance, αb should be smaller than $19/2$. This means b value can be chosen to be 19 times higher for the same α value, compared to the system without inertia.

Building upon the previous analyses, the AEA model equations are given in Equation 3.20.

$$\begin{aligned}
\vec{F}_i(t) &= \sum_{j \in S_i} \left(-\kappa \left(|\vec{r}_{ij}| - l_{ij}\right) - b|\vec{v}_{ij}|\right) \frac{\vec{r}_{ij}}{|\vec{r}_{ij}|} \\
v_d(t) &= v_0 - \alpha \left(\vec{F}_i(t) \cdot \hat{n}_i(t)\right) \\
\ddot{\vec{x}}_i(t) &= K_{acc} \left(v_d(t) - |\dot{\vec{x}}_i(t)|\right) \hat{n}_i(t) \\
\dot{\theta}_i(t) &= \beta \left[\vec{F}_i(t) \cdot \hat{n}_i^{\perp}(t)\right] + \eta\, \xi_i(t).
\end{aligned} \tag{3.20}$$

3.3 Extended Kalman Filter Implementation

The AEA model includes linear dampers, thus agents need to know the velocity difference relative to the connected agents. Mostly, relative velocity measurements are not available in swarm systems unless a specific measurement system is available. Velocity calculation from the derivative of the position measurements is not feasible because the derivative operation would amplify the measurement noise, and obtained velocity information would be too noisy to use. A Low-Pass-Filter can be used to smooth the data, but it would bring a time delay. Instead, a Kalman Filter can be used to estimate the states of the model. Kalman Filter fuses the system model information with the measurements and creates better estimates of the states. Equations of motion for the AEA model are nonlinear, therefore Extended Kalman Filter (EKF) should be used instead of the standard Kalman Filter.

3.3.1 Formulation

There are many different sources on EKF formulation but [37] is followed in this thesis.

The model of the system is given by a set of first-order nonlinear differential equations.

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}) + \mathbf{w} \\ \mathbf{z} &= h(\mathbf{x}) + \mathbf{v} \end{aligned} \tag{3.21}$$

where $\mathbf{x}$ is the state array and $\mathbf{z}$ is the measurement array. $\mathbf{w}$ and $\mathbf{v}$ are zero-mean random processes. The process noise and the measurement noise matrices are given in Equation 3.22.

$$\begin{aligned} \mathbf{Q} &= E(\mathbf{w}\mathbf{w}^T) \\ \mathbf{R} &= E(\mathbf{v}\mathbf{v}^T) \end{aligned} \tag{3.22}$$

Discrete measurement noise matrix $\mathbf{R}_k$ is a diagonal matrix filled with the error variances of the measurement sources. $\mathbf{Q}$ matrix is a diagonal matrix filled with the spectral density of the white process noise sources. Since system dynamics and measurement functions may be nonlinear, their Jacobians are used in the EKF equations. Also, the fundamental matrix is found with first-order approximation. First-order approximation is sufficient because it is used in covariance and Kalman gain calculations, not in state propagation.

$$\begin{aligned} \mathbf{F} &= \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{x=\hat{\mathbf{x}}} \\ \mathbf{H} &= \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{x=\hat{\mathbf{x}}} \\ \Phi_k &\approx I + \mathbf{F}T_s \end{aligned} \tag{3.23}$$

The discrete process noise matrix can be found using the following integration.

$$\mathbf{Q}_k = \int_0^{T_s} \Phi(\tau)\mathbf{Q}\Phi^T(\tau)d\tau \tag{3.24}$$

Using the definitions above, covariance matrices and Kalman gains are calculated.

$$\begin{aligned} \mathbf{M}_k &= \Phi_k \mathbf{P}_{k-1}\Phi_k^T + \mathbf{Q}_k \\ \mathbf{K}_k &= \mathbf{M}_k \mathbf{H}^T(\mathbf{H}\mathbf{M}_k\mathbf{H}^T + \mathbf{R}_k)^{-1} \\ \mathbf{P}_k &= (I - \mathbf{K}_k\mathbf{H})\mathbf{M}_k \end{aligned} \tag{3.25}$$

Then, state estimate $\hat{\mathbf{x}}_k$ is constructed using the a priori state estimate $\bar{\mathbf{x}}_k$ (or predicted state estimate), Kalman gains, and the measurements.

$$\begin{aligned} \hat{\dot{\mathbf{x}}}_{k-1} &= f(\hat{\mathbf{x}}_{k-1}) \\ \bar{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + \hat{\dot{\mathbf{x}}}_{k-1}T_s \\ \hat{\mathbf{x}}_k &= \bar{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - h(\bar{\mathbf{x}}_k)) \end{aligned} \tag{3.26}$$

3.3.2 Implementation

An EKF design is given for an agent that is connected to its 6 neighbors. The AEA model agents move only in the direction of their current orientation, there is no side-slip. Therefore, it is suitable to use the unicycle model and define the states as cartesian positions, velocity and orientation.

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \\ v_i \\ \theta_i \end{bmatrix} \tag{3.27}$$

The focal agent ($i = 1$) requires the states of all agents, to be able to adjust its velocity and orientation. Therefore, the state array that is estimated by the Kalman Filter is a

28-by-1 array, filled with the states of the 7 agents.

$$\mathbf{x} = [\mathbf{x}_1\ \mathbf{x}_2\ \mathbf{x}_3\ \mathbf{x}_4\ \mathbf{x}_5\ \mathbf{x}_6\ \mathbf{x}_7]' \tag{3.28}$$

In this work, the focal agent measures its own position and orientation, and also the relative positions of the other agents. Other types of sensors that measure relative velocities or angular rates could be considered; but measuring only own position, own orientation and relative positions is a reasonable approach. Since measurements are directly states, partial derivatives used in the **H** matrix are static values. Definition of the **H** matrix for this implementation is given in Equation 3.29.

$$\begin{gathered}
\mathbf{H_a} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\mathbf{H_b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\
\mathbf{H} = \begin{bmatrix} H_a & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -H_b & H_b & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -H_b & \mathbf{0} & H_b & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -H_b & \mathbf{0} & \mathbf{0} & H_b & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -H_b & \mathbf{0} & \mathbf{0} & \mathbf{0} & H_b & \mathbf{0} & \mathbf{0} \\ -H_b & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & H_b & \mathbf{0} \\ -H_b & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & H_b \end{bmatrix}
\end{gathered} \tag{3.29}$$

There can be two different approaches to the EKF design for this problem. First, it can be approached as a standard tracking problem. Then, there would be no information about the interactions between agents, therefore system dynamics matrix and state propagation would be straightforward. Second, by taking the network interactions into account, the system dynamics matrix would be filled with the correct partial derivatives and states of the focal agent would be propagated accordingly.

3.3.2.1 Filter with no model information

Equations of motion for the system with no model information are given in Equation 3.30.

$$
\begin{aligned}
\dot{x}_i &= v_i \cos\theta_i + w_1 \\
\dot{y}_i &= v_i \sin\theta_i + w_1 \\
\dot{v}_i &= w_3 \\
\dot{\theta}_i &= w_4
\end{aligned}
\tag{3.30}
$$

Cartesian positions are iterated using velocity and orientation. However, position errors due to disturbances or imperfect velocity command execution are expected. Thus, white process noise w_1 is added to both position update equations. Derivatives of the velocity and orientation do not depend on other states or a known input but it is known that velocity and orientation are subject to change. Thus, white process noises, w_3 and w_4, are added to the model.

Partial derivatives of an agent's states depend only on its own states. Therefore, **F** matrix is a block diagonal matrix that consists of blocks from each agent.

$$
\begin{aligned}
\mathbf{F}_i &= \begin{bmatrix} 0 & 0 & \cos\theta_i & -v_i \sin\theta_i \\ 0 & 0 & \sin\theta_i & v_i \cos\theta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Bigg|_{x_i = \hat{\mathbf{x}}_i} \\
\mathbf{F} &= blkdiag(\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4, \mathbf{F}_5, \mathbf{F}_6, \mathbf{F}_7)
\end{aligned}
\tag{3.31}
$$

Since F matrix is a block diagonal matrix and Q matrix is a diagonal matrix filled with the process noise values, $\mathbf{Q}_k$ is also a block diagonal matrix. The recommended approach is evaluating the Equation 3.32 symbolically once, and substituting the variables in each calculation step, rather than doing the integration in each step.

$$
\mathbf{Q}_k = \int_0^{T_s} \Phi(\tau)\mathbf{Q}\Phi^T(\tau)d\tau \tag{3.32}
$$

3.3.2.2 Filter with model information

In this implementation, network interactions between the focal agent and other agents are considered. Equations of motion for the focal agent are given in Equation 3.33.

$$\begin{aligned} \dot{x}_1 &= v_1 \cos\theta_1 + w_1 \\ \dot{y}_1 &= v_1 \sin\theta_1 + w_1 \\ \dot{v}_1 &= K_{acc}(v_0 + \alpha F^{||} - v_1) + w_{3,1} \\ \dot{\theta}_1 &= \beta F^{\perp} + w_{4,1} \end{aligned} \tag{3.33}$$

$w_{3,1}$ and $w_{4,1}$ are the white process noise sources for the focal agent. Equations of motion for the other agents are same as the no model case.

Partial derivatives for this model require tedious calculations. First, partial derivatives of v_1 with respect to all states is given.

$$\partial \dot{v}_1 = K_{acc}\alpha \partial F^{||} - K_{acc}\partial v_1 \tag{3.34}$$

It can be seen that partial derivatives of v_1 are related to the partial derivatives of parallel force $F^{||}$. Only the partial derivative with respect to self velocity has an additional term, which equals to K_{acc}. Therefore, partial derivatives of the parallel force are focused.

The total force is the sum of spring force and damper force. Calculations of both force components are given in Equation 3.35.

$$\begin{aligned} F^{||} &= F^{||}_{spring} + F^{||}_{damper} \\ F^{||}_{spring} &= \kappa(d_{1n} - 1)(\vec{u}_1 \cdot \vec{u}_{link,1n}) \\ F^{||}_{damper} &= b(v_n(\vec{u}_n \cdot \vec{u}_{link,1n}) - v_1(\vec{u}_1 \cdot \vec{u}_{link,1n}))(\vec{u}_1 \cdot \vec{u}_{link,1n}) \end{aligned} \tag{3.35}$$

The distance term and the unit vectors used in Equation 3.35 are defined in Equation 3.36.

$$
\begin{aligned}
\bar{u}_n &= \begin{bmatrix} \cos\theta_n \\ \sin\theta_n \end{bmatrix} \\
\bar{u}_n^{\perp} &= \begin{bmatrix} -\sin\theta_n \\ \cos\theta_n \end{bmatrix} \\
\bar{u}_{link,1n} &= \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} / d_{1n} \\
d_{1n} &= ((x_n - x_1)^2 + (y_n - y_1)^2)^{\frac{1}{2}}
\end{aligned}
\tag{3.36}
$$

In this formulation, distances are mainly used in the scaling of the unit vectors. Therefore, partial derivatives of the distances are neglected for simplicity.

Positional derivatives of the parallel force are given in Equation 3.37.

$$
\begin{aligned}
\frac{\partial F^{\|}_{spring}}{\partial x_n} &= \kappa \cos\theta_1 (1 - \frac{1}{d_{1n}}) \\
\frac{\partial F^{\|}_{spring}}{\partial y_n} &= \kappa \sin\theta_1 (1 - \frac{1}{d_{1n}}) \\
\frac{\partial F^{\|}_{damper}}{\partial x_n} &= \frac{b}{d_{1n}}((\vec{u}_1 \cdot \vec{u}_{link,1n})(v_n \cos\theta_n - v_1 \cos\theta_1) \\
&\quad + \cos\theta_1 (v_n (\vec{u}_n \cdot \vec{u}_{link,1n}) - v_1 (\vec{u}_1 \cdot \vec{u}_{link,1n}))) \\
\frac{\partial F^{\|}_{damper}}{\partial y_n} &= \frac{b}{d_{1n}}((\vec{u}_1 \cdot \vec{u}_{link,1n})(v_n \sin\theta_n - v_1 \sin\theta_1) \\
&\quad + \sin\theta_1 (v_n (\vec{u}_n \cdot \vec{u}_{link,1n}) - v_1 (\vec{u}_1 \cdot \vec{u}_{link,1n}))) \\
\frac{\partial F^{\|}}{\partial x_1} &= -\sum_{n=2}^{N} \frac{\partial F^{\|}}{\partial x_n} \\
\frac{\partial F^{\|}}{\partial y_1} &= -\sum_{n=2}^{N} \frac{\partial F^{\|}}{\partial y_n}
\end{aligned}
\tag{3.37}
$$

Velocity derivatives of the parallel force are given in Equation 3.38.

$$\begin{aligned}\frac{\partial F^{\|}_{damper}}{\partial v_n} &= b(\vec{u}_1 \cdot \vec{u}_{link,1n})(\vec{u}_n \cdot \vec{u}_{link,1n}) \\ \frac{\partial F^{\|}_{damper}}{\partial v_1} &= b\sum_{n=2}^{N} -(\vec{u}_1 \cdot \vec{u}_{link,1n})^2\end{aligned} \tag{3.38}$$

Orientation derivatives of the parallel force are given in Equation 3.39.

$$\begin{aligned}\frac{\partial F^{\|}_{spring}}{\partial \theta_1} &= \kappa\sum_{n=2}^{N}(1 - \frac{1}{d_{1n}})(\vec{u}_1^{\perp} \cdot \vec{u}_{link,1n}) \\ \frac{\partial F^{\|}_{damper}}{\partial \theta_n} &= bv_n(\vec{u}_1 \cdot \vec{u}_{link,1n})(\vec{u}_n^{\perp} \cdot \vec{u}_{link,1n}) \\ \frac{\partial F^{\|}_{damper}}{\partial \theta_1} &= b\sum_{n=2}^{N}(v_n(\vec{u}_n \cdot \vec{u}_{link,1n})(\vec{u}_1^{\perp} \cdot \vec{u}_{link,1n}) - 2v_1(\vec{u}_1^{\perp} \cdot \vec{u}_{link,1n})(\vec{u}_1 \cdot \vec{u}_{link,1n}))\end{aligned} \tag{3.39}$$

Then, partial derivatives of θ_1 are calculated.

$$\begin{aligned}\dot{\theta}_1 &= \beta F^{\perp} \\ \partial\dot{\theta}_1 &= \beta\partial F^{\perp}\end{aligned} \tag{3.40}$$

Similar to the relation between v_1 and the parallel force, θ_1 derivatives are related to the perpendicular force, $F^{\perp}$.

$$\begin{aligned}F^{\perp} &= F^{\perp}_{spring} + F^{\perp}_{damper} \\ F^{\perp}_{spring} &= \kappa(d_{1n} - 1)(\vec{u}_1^{\perp} \cdot \vec{u}_{link,1n}) \\ F^{\perp}_{damper} &= b(v_n(\vec{u}_n \cdot \vec{u}_{link,1n}) - v_1(\vec{u}_1 \cdot \vec{u}_{link,1n}))(\vec{u}_1^{\perp} \cdot \vec{u}_{link,1n})\end{aligned} \tag{3.41}$$

Positional derivatives are given in Equation 3.42.

$$\begin{aligned}
\frac{\partial F^{\perp}_{spring}}{\partial x_n} &= -\kappa \sin\theta_1 (1 - \frac{1}{d_{1n}}) \\
\frac{\partial F^{\perp}_{spring}}{\partial y_n} &= \kappa \cos\theta_1 (1 - \frac{1}{d_{1n}}) \\
\frac{\partial F^{\perp}_{damper}}{\partial x_n} &= \frac{b}{d_{1n}}((\vec{u}_1^{\perp} \cdot \vec{u}_{link,1n})(v_n \cos\theta_n - v_1 \cos\theta_1) - \sin\theta_1 (v_n(\vec{u}_n \cdot \vec{u}_{link,1n}) \\
&\quad - v_1(\vec{u}_1 \cdot \vec{u}_{link,1n}))) \\
\frac{\partial F^{\perp}_{damper}}{\partial y_n} &= \frac{b}{d_{1n}}((\vec{u}_1^{\perp} \cdot \vec{u}_{link,1n})(v_n \sin\theta_n - v_1 \sin\theta_1) + \cos\theta_1 (v_n(\vec{u}_n \cdot \vec{u}_{link,1n}) \\
&\quad - v_1(\vec{u}_1 \cdot \vec{u}_{link,1n}))) \\
\frac{\partial F^{\perp}}{\partial x_1} &= -\sum_{n=2}^{N} \frac{\partial F^{\perp}}{\partial x_n} \\
\frac{\partial F^{\perp}}{\partial y_1} &= -\sum_{n=2}^{N} \frac{\partial F^{\perp}}{\partial y_n}
\end{aligned} \tag{3.42}$$

Velocity derivatives are given in Equation 3.43.

$$\begin{aligned}
\frac{\partial F^{\perp}_{damper}}{\partial v_n} &= b(\vec{u}_1^{\perp} \cdot \vec{u}_{link,1n})(\vec{u}_n \cdot \vec{u}_{link,1n}) \\
\frac{\partial F^{\perp}_{damper}}{\partial v_1} &= -b \sum_{n=2}^{N} (\vec{u}_1^{\perp} \cdot \vec{u}_{link,1n})(\vec{u}_1 \cdot \vec{u}_{link,1n})
\end{aligned} \tag{3.43}$$

Orientation derivatives are given in Equation 3.44.

$$\begin{aligned}
\frac{\partial F^{\perp}_{spring}}{\partial \theta_1} &= -\kappa \sum_{n=2}^{N} (1 - \frac{1}{d_{1n}})(\vec{u}_1 \cdot \vec{u}_{link,1n}) \\
\frac{\partial F^{\perp}_{damper}}{\partial \theta_n} &= b v_n (\vec{u}_1^{\perp} \cdot \vec{u}_{link,1n})(\vec{u}_n^{\perp} \cdot \vec{u}_{link,1n}) \\
\frac{\partial F^{\perp}_{damper}}{\partial \theta_1} &= b \sum_{n=2}^{N} (-v_n(\vec{u}_n \cdot \vec{u}_{link,1n})(\vec{u}_1 \cdot \vec{u}_{link,1n}) + v_1((\vec{u}_1 \cdot \vec{u}_{link,1n})^2 \\
&\quad - (\vec{u}_1^{\perp} \cdot \vec{u}_{link,1n})^2))
\end{aligned} \tag{3.44}$$

All of the calculations above are used to fill the 3rd and 4th rows of the $\mathbf{F}$ matrix. Other elements are same as the case with no interaction model.

Since the $\mathbf{F}$ matrix is not a block diagonal matrix anymore, calculation of $\mathbf{Q}_k$ is even more tedious. The recommended approach is to evaluate the integral by symbolic calculation once, then use the evaluated partial derivatives accordingly in the construction of $\mathbf{Q}_k$ in each filter iteration.

CHAPTER 4

EXPERIMENTAL ANALYSIS

4.1 Particle-based Simulations

4.1.1 Active-Elastic and Vicsek-Network Model Results

Multiple simulations are carried out for the VN and AE models, using interaction topologies that interpolate either between NN and ER networks or between NN and SF networks.

In order to reduce the parameter space, the same AE model parameters are used in all runs, setting $\alpha = 0.01$, $\beta = 0.12$, $v_0 = 0.002$, and $\kappa = 5$ while integrating Equations 3.6 and 3.7 with a timestep of $\Delta t = 0.1$. These are the same parameters used in [7], which were shown to produce rapid and reliable self-organization into the aligned state for systems with NN interaction topologies. The parameter space for the VN model is not restricted, since Equation 3.9 only depends on the noise intensity η and has no additional parameters.

All systems are arranged in a square lattice configuration. Three different system sizes are considered: a small system consisting of 1024 agents (arranged in a 32×32 lattice) with 3906 connections, an intermediate system of 10000 agents (100×100 lattice) with 39402 connections, and a large system of 90000 agents (300×300 lattice) with 358202 connections. Given that reaching a well-converged statistically stationary state has different computational cost for different system sizes, models, and network structures; the number and duration of the runs are varied accordingly. These additional simulation details are thus reported for each case in the corresponding figure caption. To evaluate the order in the group, the polarization order metric given in [15] is used. Calculation of the order metric, ψ, is given in Equation 4.1. $\psi \approx 1$ means the headings of the agents are aligned with each other and the group is moving collectively. Conversely, $\psi \approx 0$ means headings are randomly distributed in the group and the group is in disorder. In all simulation results presented in this subsection, the swarm is initialized in the fully aligned state ($\psi = 1$).

$$\psi = \frac{1}{N} \left| \sum_{i=1}^{N} \hat{n}_i \right| \tag{4.1}$$

For each model, topology, and system size, the degree of order achieved as a function of the angular noise η is studied. Then, the critical noise η_c of the order-disorder transition is identified to analyze how it depends on the topology.

Once all agents are placed in a rectangular lattice (as in Figure 4.1) and the network topology is defined, the l_{ij} values are set to $l_{ij} = \|\vec{x}_j(0) - \vec{x}_i(0)\|$, corresponding to the initial distances between agents i and j. This way, there would be no elastic forces at $t = 0$ and that all stresses would be produced later by the self-propulsion dynamics.

4.1.1.1 Phase transition diagrams

Phase transition diagrams that display the order metric ψ as a function of noise intensity η for different topologies are presented, first in simulations of the VN model and then of the AE model. Each point in these phase transition diagrams is the result of averaging the mean ψ values of multiple runs. For $p > 0$, each one of these runs are performed on a different random superposition of a NN network and a random ER or SF interaction network, which remained fixed throughout the simulation. The degree distributions of all the random SF networks considered in this section are generated using a $b = -2$ exponent.

Figure 4.2a displays the phase transition diagrams for the VN model, with interaction topologies interpolating between NN and random ER network structures, whereas Figure 4.2b displays the corresponding diagrams with topologies interpolating between NN and random SF networks. The interpolating parameter p controls the topology, with $p = 0$ corresponding to the NN case and $p = 1$ to the random case.

For $p = 0$, the same curve is displayed in both plots. It appears to show a continuous

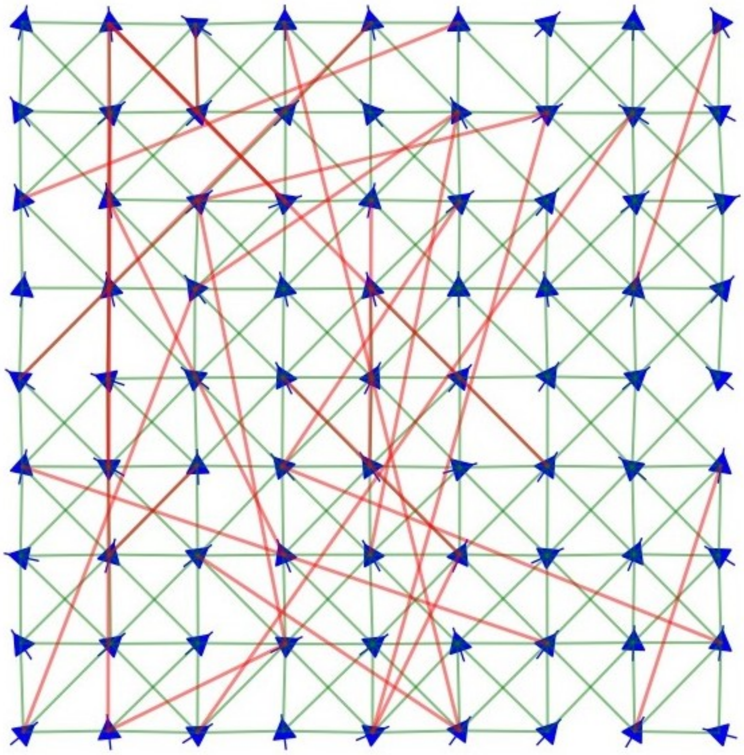
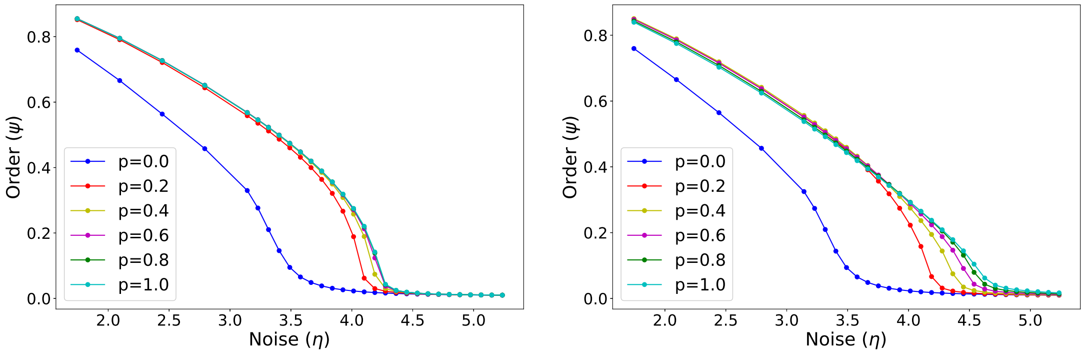


Figure 4.1: Simulation snapshot of a 9×9 agent system with its corresponding connectivity diagram. The blue arrows show the positions and orientations of the agents and the lines represent their interactions. The displayed interaction topology is the superposition of a nearest neighbor network (green links) and a homogeneous ER random network (red links). The displayed state presents partial alignment, with agents mostly heading upwards.

transition from an aligned to a disordered state.

The curves with intermediate values of p in Figure 4.2 show that a few random nonlocal interactions (with ER or SF topology) are enough to significantly increase the critical noise, and therefore the resilience to noise of the system. This is not surprising, since a small fraction of long-range interactions is expected to strongly increase system integration, and thus resilience to noise, as it has been shown for small-world networks [38]. In addition, the plots show that long-range links with SF topology increase more effectively the resilience to noise than those with ER topology. This is also consistent with expectations, since in SF networks the mean distance between nodes is significantly reduced [39], which improves the convergence of the distributed



(a)

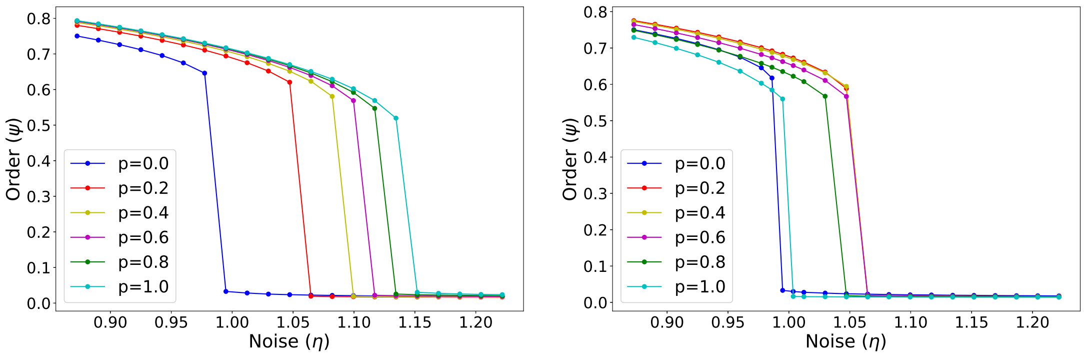
(b)

Figure 4.2: Phase transition diagrams of the VN model with different interaction topologies, ranging from NN ($p = 0.0$) to ER ($p = 1.0$) networks in Panel (a) and from NN ($p = 0.0$) to SF ($p = 1.0$) networks with $b = -2$ in Panel (b). The transition appears as continuous for all cases. A larger fraction of random long-range connections (either ER or SF) increases the critical noise. Each point results from 8 runs (each one with a different random superposition of a NN and a random ER or SF network) of 5×10^5 timesteps per noise value, for an intermediate system size ($N = 100 \times 100$).

average consensus protocol that underlies the VN model, thus producing higher critical noise values.

Figure 4.3a presents the phase transition diagrams for the AE model with topologies interpolating between NN and random ER networks, whereas Figure 4.3b displays the corresponding diagrams with topologies interpolating between NN and random SF networks. Here again, p is the interpolating parameter.

In this model, the transition appears to be first-order for all values of p. This is consistent with previous results obtained in the $p = 0$ case [7], where it was shown that the ordered and disordered branches can co-exist for a range of η values close to the critical noise. As explained at the beginning of this section, only the ordered branch is found (in the bistable region), because all simulations are started from a fully aligned ($\psi = 1$) initial condition. When a small fraction of long-range random connections is included, for $p = 0.2$, a significant increase in the resilience to noise is observed. In the ER case, the critical noise then continues to increase with p, as in the previ-



(a)

(b)

Figure 4.3: Phase transition diagrams of the AE model with different interaction topologies, ranging from NN ($p = 0.0$) to ER ($p = 1.0$) networks in Panel (a), and from NN ($p = 0.0$) to SF ($p = 1.0$) networks with $b = -2$ in Panel (b). All transitions appear to be discontinuous. For each one, only the upper solution branch is displayed. In contrast to the VN case presented in Fig. 4.2, the critical noise only increases monotonically with the fraction of random long-range connections in Panel (a), while it decreases for $p > 0.6$ in Panel (b). Each point results from 40 runs (each one with a different random superposition of a NN and a random ER or SF network) of 10^6 timesteps per noise value, for an intermediate system size ($N = 100 \times 100$).

ously presented VN case, which is consistent with the aforementioned expectation that more long-range interactions favor system integration. In the SF case (Figure 4.3b), however, the critical η starts decreasing for $p >\sim 0.6$, so resilience to noise is maximized for intermediate values of p. This is surprising, since it implies that including more long-range interactions can hinder system cohesion. Also, fully random SF topology ($p = 1$) displays an even lower critical noise value, which contradicts the common notion that SF networks should improve system integration by reducing the mean distance between nodes [39].

4.1.1.2 Critical noise as a function of topological structure

The critical noise value η_c is presented here as a function of the topological control parameter p, for all the phase transition diagrams presented above and for other sys-

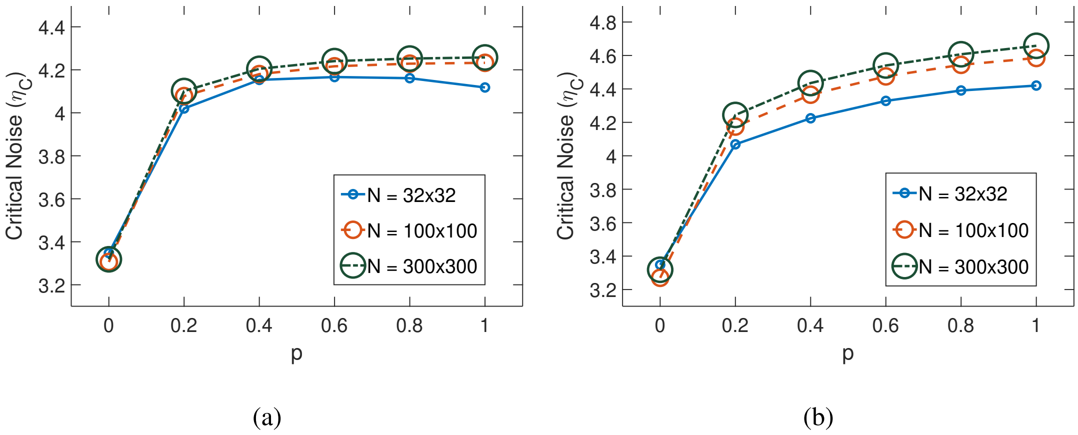
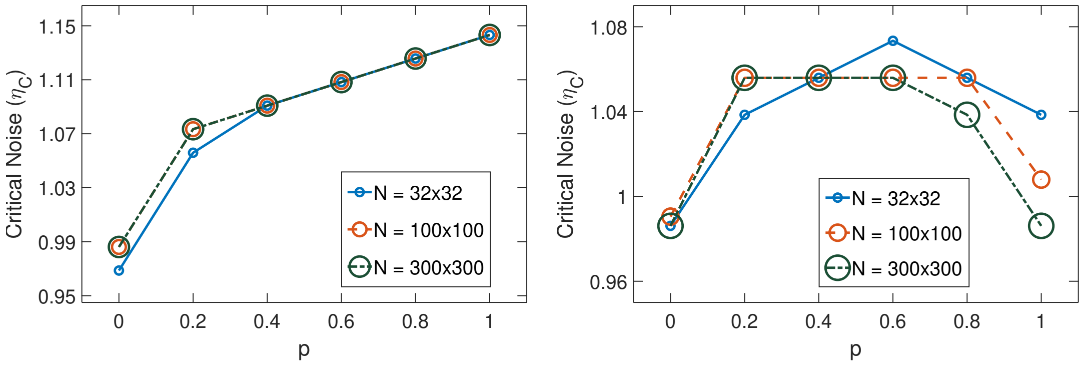


Figure 4.4: Critical noise η_c as a function of the topological control parameter p for the VN model with different system sizes. The parameter p interpolates between NN and ER random networks in Panel (a), and between NN and SF (with $b = -2$) random networks in Panel (b). As the fraction of random, long-range connections is increased, the critical noise increases in both cases. Small system ($N = 32 \times 32$): 40 runs of 2×10^5 timesteps per noise value. Intermediate system ($N = 100 \times 100$): 8 runs of 5×10^5 timesteps per noise value. Large system ($N = 300 \times 300$): 8 runs of 10^5 timesteps per noise value.

tem sizes. In order to do this, first an objective criteria must be defined for computing the transition point from the numerical data.

For the VN model, given that the transition appears as continuous, a standard method for detecting the critical noise in second-order phase transitions is used. η_c is identified as the point where the variance of the order metric ψ is maximized. To interpolate between the discrete simulated values of η, the η value with the highest $\mathrm{Var}(\psi_\eta)$ (the variance of the order metric over all runs performed with noise level η) and the two adjacent η values are selected. Then, η_c is computed as the location of the maximum of a quadratic curve that passes through the three corresponding $(\eta, \mathrm{Var}(\psi_\eta))$ points. For the AE model, since in this case the transition is discontinuous, η_c is defined as the point where the ordered solution ceases to exist. The critical noise level is computed as the midpoint between the highest η value at which an ordered solution is identified and the next tested value of η, for which no ordered stationary state could be found. Using these criteria, critical noise η_c is computed for three different system sizes in



(a)

(b)

Figure 4.5: Critical noise η_c as a function of the topological control parameter p for the AE model with different system sizes. The parameter p interpolates between NN and ER random networks in Panel (a), and between NN and SF (with $b = -2$) random networks in Panel (b). The ER case displays higher η_c for higher p values, as in Fig. 4.4. In the SF case, however, the maximum η_c is reached here at intermediate p values. Small system (32×32): 40 runs of 10^6 timesteps per noise value. Intermediate system (100×100): 40 runs of 10^6 timesteps per noise value. Large system (300×300): 8 runs of 5×10^5 timesteps per noise value.

each model and topology considered.

Figure 4.4 displays η_c as a function of p for the VN model, with p interpolating between NN and ER networks in Panel (a) and between NN and SF networks in Panel (b). As previously discussed, the same behavior is observed in both cases: the resilience to noise increases with p (i.e., with the fraction of random connections). The only exception is the $N = 32 \times 32$ case in Panel (a), where strong finite size effects appear to reduce the resilience to noise as p approaches 1. In both cases, it should be noted that the benefit of adding more random links starts saturating for larger p.

Figure 4.5 shows η_c as a function of p for the AE model. When p interpolates between NN and ER networks, as shown in Panel (a), the resilience to noise increases with p in a way similar to the VN case. However, it should be noted that η_c appears to grow linearly with p for $p \geq 0.4$ in all system sizes, without starting to saturate as in the VN case. When p interpolates between NN and SF networks, in Panel (b), the resilience to noise increases initially with p, but then decreases as p approaches 1. This reduction of the critical noise for larger p values is seen to depend on system size, with η_c becoming smaller at $p = 1$ in bigger systems.

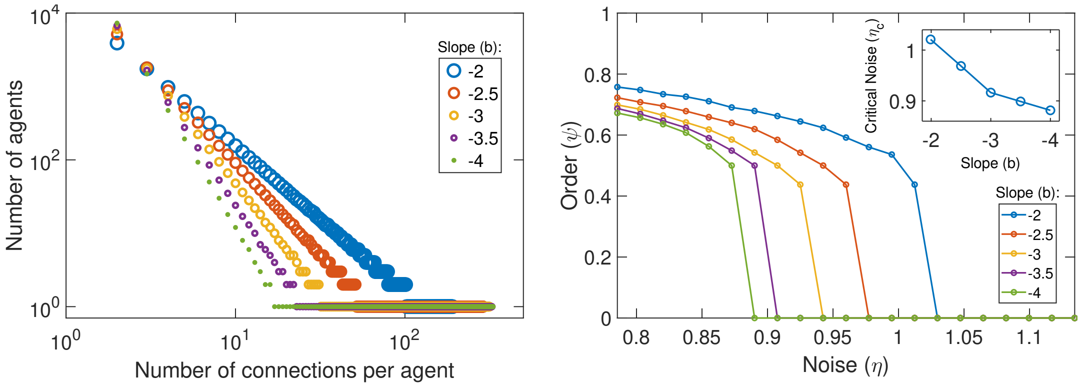
The reduced resilience to noise for a larger fraction of random SF connections observed above in the AE model appears to be a consequence of the overabundance of nodes with low degree. This is because nodes with low degree are more easily excitable by noise, since they represent weakly coupled components of the elastic network that mediates the interactions in the AE model. This is also consistent with the decrease of resilience in larger systems, since the fraction of nodes with few connections increases with N. In order to test this hypothesis, the relationship between critical noise and the exponent of the degree distribution in SF interaction networks is analyzed in the next part.

4.1.1.3 Critical noise as a function of scale-free exponent

Here, simulations of the AE model interacting through fully random ($p = 1$) SF topologies are considered, and relationship between the critical noise and the exponent b of the degree distribution in Equation 3.4 is analyzed.

Figure 4.6a shows the different degree distributions that is implemented in the intermediate ($N = 100 \times 100$) system size. In order to keep N and K constant, $k_{\max}$ must be increased for higher values of $|b|$, which extends the $n_k = 1$ region where the distribution is flat.

For each exponent b, multiple simulations of the AE model are performed, using different realizations of a SF interaction network with the corresponding degree distribution. Figure 4.6b presents the resulting phase transition diagrams. The inset displays the respective critical noise values as a function of b. For all SF exponents considered, a discontinuous order-disorder transition is found, as in the previously explored $b = -2$ case. The figure shows that η_c is reduced as the slope of the distribution becomes steeper. This implies that the resilience to noise decreases as b becomes more negative, that is, as the fraction of nodes with a low number of connections is



(a)

(b)

Figure 4.6: Simulation details and results of the AE model analysis of the relationship between the critical noise and the SF interaction network exponent. Panel (a) presents the different degree distributions, with various b exponents, that are implemented in the fully random ($p = 1$) SF interaction networks considered. Each b corresponds to a different slope in this log-log plot. Panel (b) displays the resulting phase transition diagrams for each of these distributions. The inset shows the critical noise as a function of b. It is observed that η_c decreases for steeper distribution slopes. All simulations are carried out in an intermediate size system ($N = 100 \times 100$), performing 16 runs of 2×10^5 timesteps per b value.

increased, in agreement with the hypothesis in the previous part.

4.1.2 Active-Elastic-Anticipation Model Simulations

Particle agent model simulations are conducted for the AEA model, in order to uncover its novel properties and similarities to AE and VN models. Agents are placed in a bounding hexagon (that lies in a 2D infinite space) with triangular tessellation. This method is chosen over the square tessellation used in the previous analyses, because initial distance (l_{ij}) between all neighbors is same. Agents inside the swarm have 6, agents on the sides have 4, and agents at the corners have 3 connections. For a given number of agents per side value l, the number of agents in the swarm equals to $N = 3l^2 - 3l + 1$.

Parameters used in the AEA model simulations are $\alpha = 0.004$, $\beta = 0.12$, $v_0 = 0.002$, and $\kappa = 5$; while integrating Equation 3.20 with timestep $\Delta t = 0.1$. Damper coefficient, b, is set according to the scenario. Same order metric given (ψ) in the previous subsection (Equation 4.1) is used for evaluating the order in the group for AEA model analyses.

Figure 4.7 displays the initial positions, initial orientations, and fixed Nearest-Neighbor Network links between the agents.

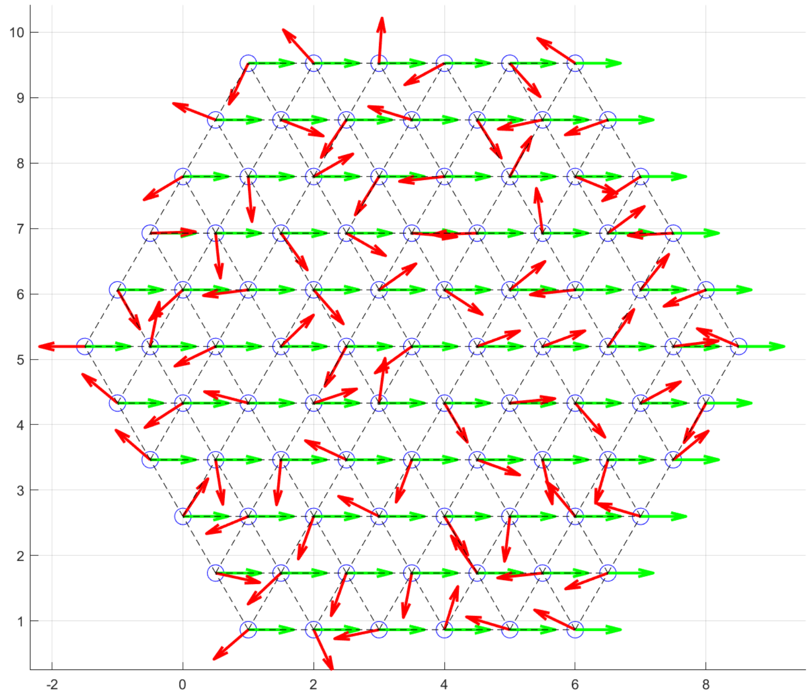


Figure 4.7: Initial positions, initial NN links, random and aligned initial orientations, for a hexagon with 6 agents on the sides.

4.1.2.1 Phase Transition Diagrams

Phase transition diagrams that display the order metric ψ as a function of noise intensity η are presented. Effect of damper coefficient b and topological control parameter p is studied. Only random network considered for the AEA model is the ER random network.

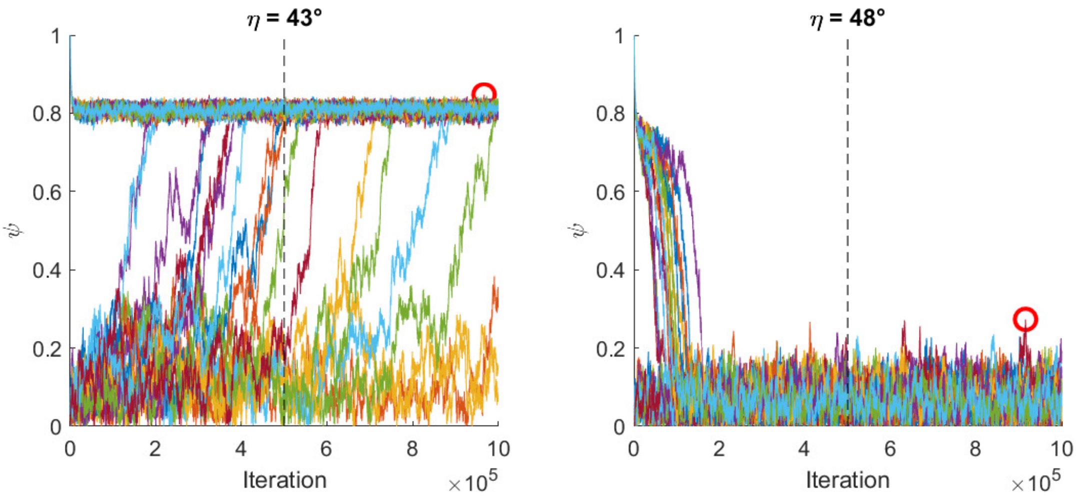


Figure 4.8: AEA model ψ timeseries, for $p = 0$ and $b = 0$. The black dashed line represents the half point and the red circle is the supremum of the remaining data.

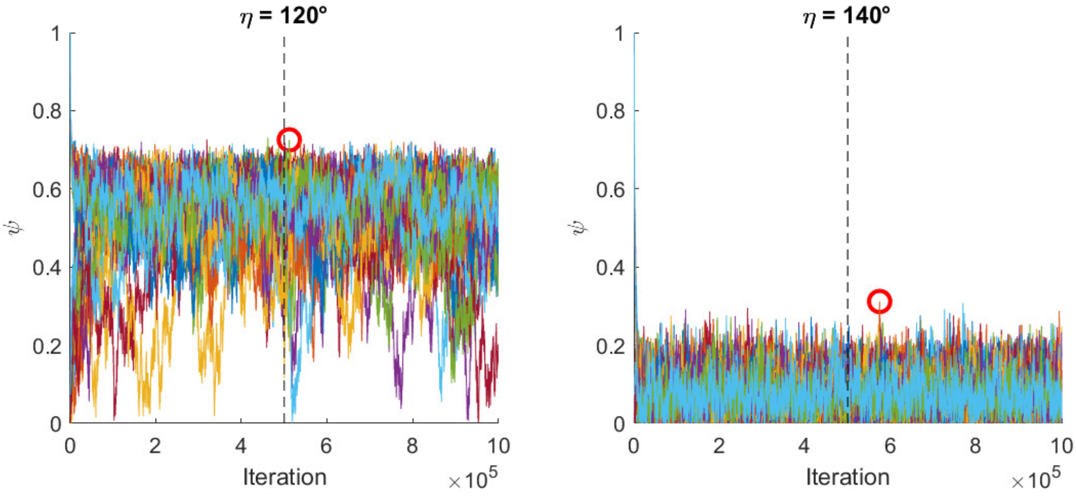


Figure 4.9: AEA model ψ timeseries, for $p = 0$ and $b = 150$. The black dashed line represents the half point and the red circle is the supremum of the remaining data.

For the AEA model with NN network, evolution of the order metric for $b = 0$ is given in Figure 4.8, and for $b = 150$ is given in Figure 4.9. To be able to get the steady-state order value for a given noise intensity, the first half of the iterations are discarded. Then, the maximum order metric value captured in the ensemble of all remaining data is recorded and displayed in the phase transition diagram for the respective noise intensity value.

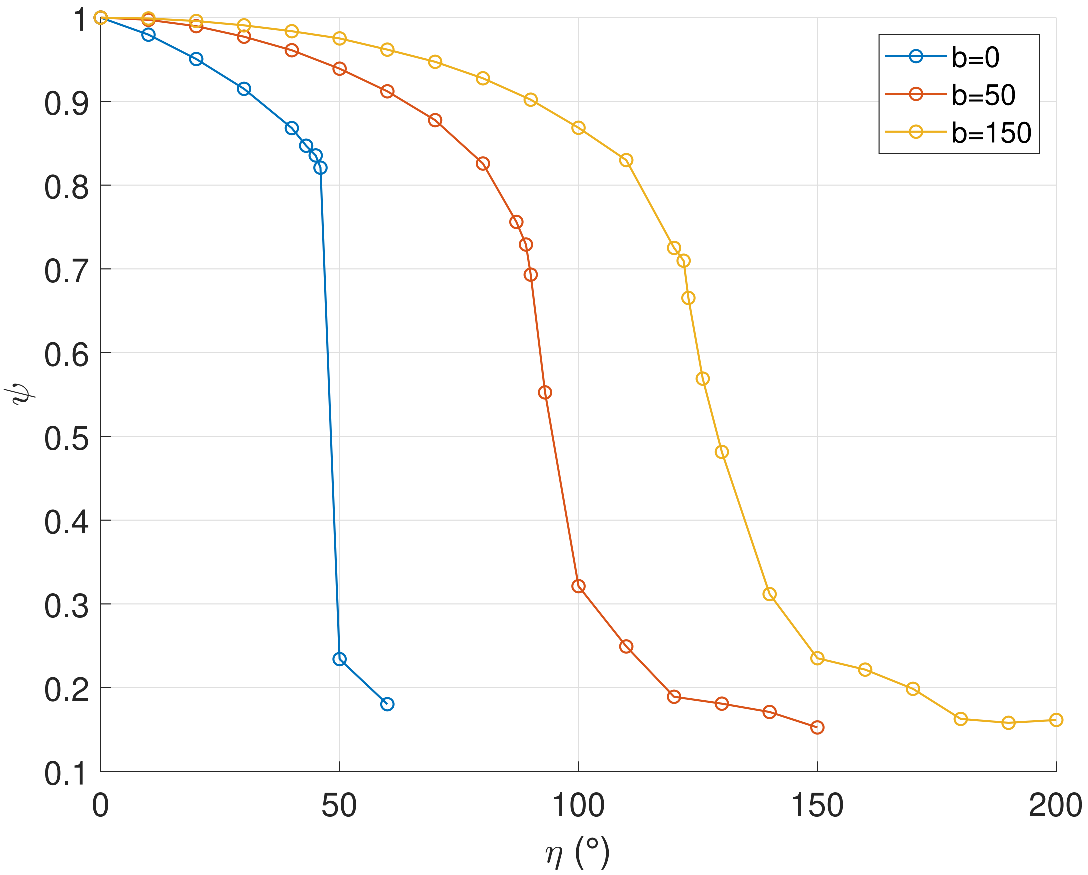


Figure 4.10: Phase transition diagrams of the AEA model with NN network for different b values. Each point results from 40 runs of 10^6 timesteps per noise value, for a system size $N = 1141$.

Figure 4.10 presents the phase transition diagrams for the AEA model with the NN network ($p = 0$). For each noise intensity (η) value in the phase transition diagram, 40 simulations with 10^6 simulated time steps are done. 3 different damper coefficient values are studied: $b = [0, 50, 150]$.

For $b = 0$, there is no anticipatory interaction between agents as in the case of the AE model. The only difference with the AE model is the existence of inertia. The phase transition diagram shows that the order metric stays high until the $\eta = 40°$ value, and then declines sharply. Therefore, the AEA model with zero damper coefficient can achieve collective motion up to a critical noise value and its phase transition is first-order. This result shows that the inclusion of inertia in the AEA model does not hinder the collective motion ability and did not change the phase transition characteristics when compared to the AE model.

For $b = 150$ case, phase transition characteristic seems to be continuous and it is similar to the alignment-based VN model rather than the position-based AE model. And for $b = 50$ case, the transition characteristic seems to be a mixture of the AE model and the VN model. This result is in line with the expectations. As the damper coefficient increases, which means the anticipation horizon is increasing, alignment-based interactions become dominant to position-based interactions. Therefore, phase transition characteristics switch gradually from first-order to second-order.

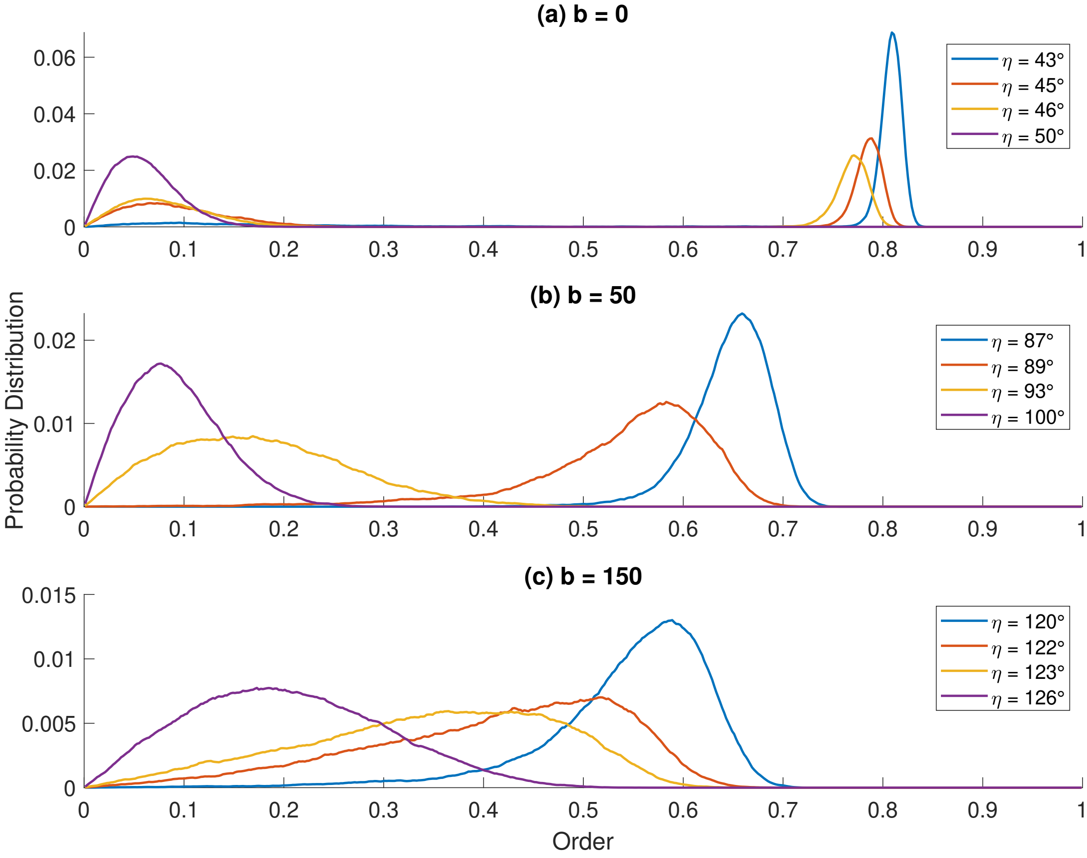


Figure 4.11: Distributions of order metric timeseries for different b and η values

Figure 4.11 gives the histograms of the order metric timeseries that are used in the construction of the phase transition diagram in Figure 4.10, for different noise and b values. It can be seen that for $b = 0$, the distribution is bi-modal, two peaks are apparent in the graph and the intensity of the ordered peak decreases as the noise intensity increases. When b is increased to 50, tails of the curves are elongated. At $b = 150$, the distribution is unimodal and the peak can be at any order value. These results conclude that as the anticipation horizon increases, interactions between agents become more alignment-based.

Phase transition analyses with different p values that interpolates between NN and ER random networks is conducted for the AEA model. In the damper coefficient analyses, a hexagon shape is preferred since it gives equal distances between all neighboring agents. However, agents in the bulk have 6 connections, which lowers the links per agent (K/N) ratio, compared to the square lattice with 8 connections. Creating a fully random network that is connected would be more difficult and created networks would have a significant amount of agents with only 1 link because the expected value of the Poissonian distribution would shift from 8 to 6. Hence, for the p analyses, agents are put in a square lattice with 8 connections, similar to the analyses for the AE and VN models.

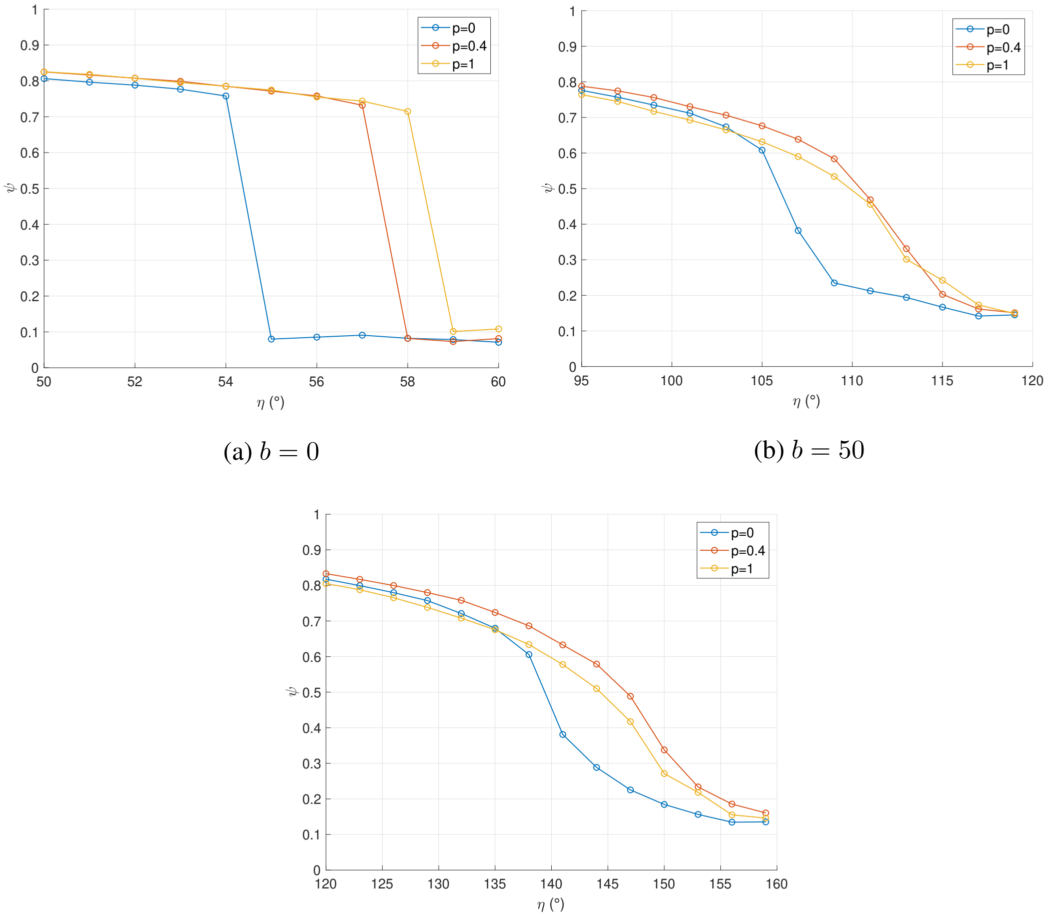
Figure 4.12 depicts the phase transition diagrams with p interpolating between the NN network and the ER random network. Number of agents in the system is 4900 ($N = 70 \times 70$). Each point in the diagrams results from 20 runs of 4×10^5 timesteps per noise value. For $p > 0$, each one of these runs is performed on a different random superposition of a NN network and a random ER network, which remained fixed throughout the simulation. Also, each simulation has a different noise generator seed.

For $b = 0$ case, results are similar to the AE model. As p increases, resilience to noise increases. For $b = 50$ and $b = 150$, networks with long-range links perform better than the NN network against the noise. However, the fully random network performs worse than the interpolated network.

4.1.2.2 Narrow passage scenario

The new design parameter, b, opens up new possibilities for accomplishing tasks. Anticipation brings alignment-based interactions, therefore more fluid-like motion is expected, compared to the no anticipation case.

Passing through a narrow passage can be used as a qualitative example. Figure 4.13 shows the results of the $p = 0, b = 0$ case and Figure 4.14 shows the results of the $p = 0, b = 150$ case (same as the case ’a’ and case ’c’ in Figure 4.11, respectively),



(a) $b = 0$

(b) $b = 50$

(c) $b = 150$

Figure 4.12: AEA model phase transition diagrams for different b values with p interpolating between NN and ER random networks.

in a simulation where agents initially go towards a narrow passage with zero noise.

In the no-anticipation case shown in Figure 4.13, agents could not pass through the passage. A consensus in the direction of movement is not reached, thus the swarm could not shrink in size and then agents turn back from the passage. In Figure 4.14, the swarm contracted in size in an orderly manner and passed to the other side easily. In this example, the only difference is the addition of anticipation.

In order to study the effect of anticipation in passing through a narrow passage scenario, multiple simulations with varying parameters are conducted. A hexagon swarm with 8 agents on the sides is placed in the arena. The length of the swarm in the y-axis

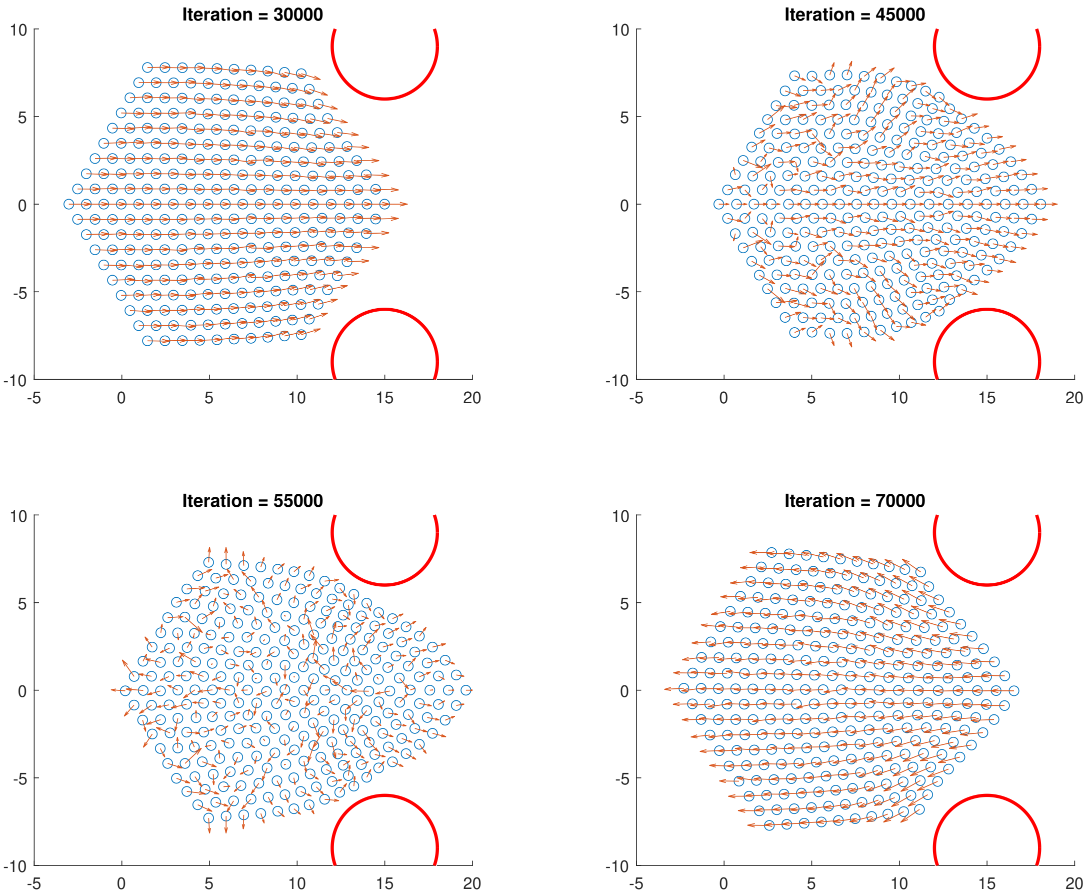


Figure 4.13: AEA model with no anticipation - passing through a narrow passage scenario

is $7\sqrt{3}$ and this value is set as one normalized width, denoted as w_n. Two circular obstacles with $r = 15$ are placed in $x = 15$ and the separation between them varies from 0.7 to 0.9 normalized width. Noise intensity is set as $20°$, which is below the critical noise for any configuration, and 40 different simulations with different noise generator seeds are done for each case. Damper coefficient values used in the simulations are 0, 10, 50, 150 and 450. Runs that can reach the other side ($x = 30$) without any collision are regarded as successful and elapsed times of the successful runs are recorded.

In the trials, it was observed that swarms with $p > 0$ mostly fail to pass through the passage because long-range links prevent the swarm to constrict. Therefore, only the NN network is considered for this scenario.

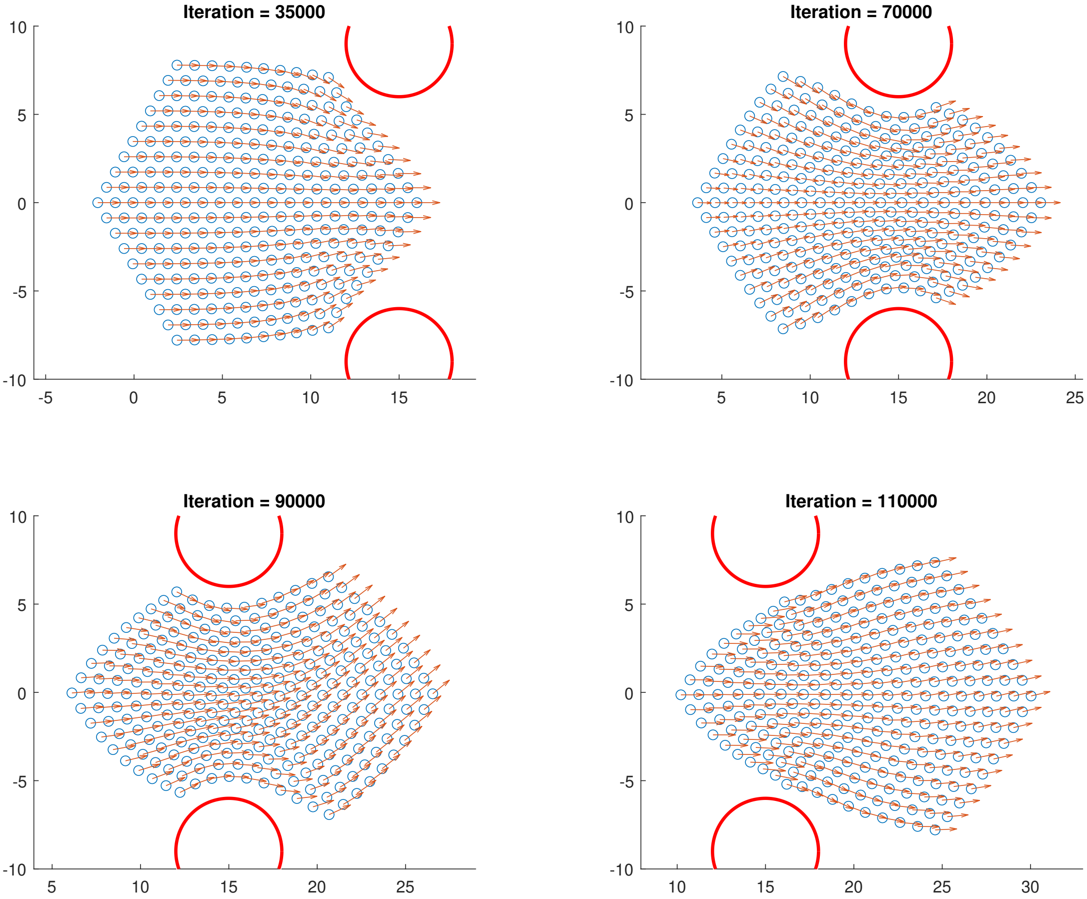


Figure 4.14: AEA model with anticipation - passing through a narrow passage scenario

Figure 4.15 gives the ratio of the successful runs for different b and w_n values. No anticipation case ($b = 0$) has %0 success with $0.7w_n$ and %10 success with $0.8w_n$. With a small increase in b from 0 to 10, success ratio in $0.8w_n$ jumps from %10 to %60. It can be seen that as b increases, success ratio increases for any w_n. Also, only $b = 450$ case has non-zero success ratio for $0.7w_n$.

Figure 4.16 gives the mean and standard deviations of the elapsed times of the successful runs given in Figure 4.15. Regardless of the b value, means of the elapsed times increase as the width of the passage decreases. This is reasonable, since the smaller gap may be interpreted as larger friction to the swarm. Also, as b increases, means of the elapsed times decrease for any w_n value. These results support the argument that as b (or anticipation horizon) increases, more fluid-like motion is observed. Also, anticipation helps the swarm in passing through a narrow passage task.

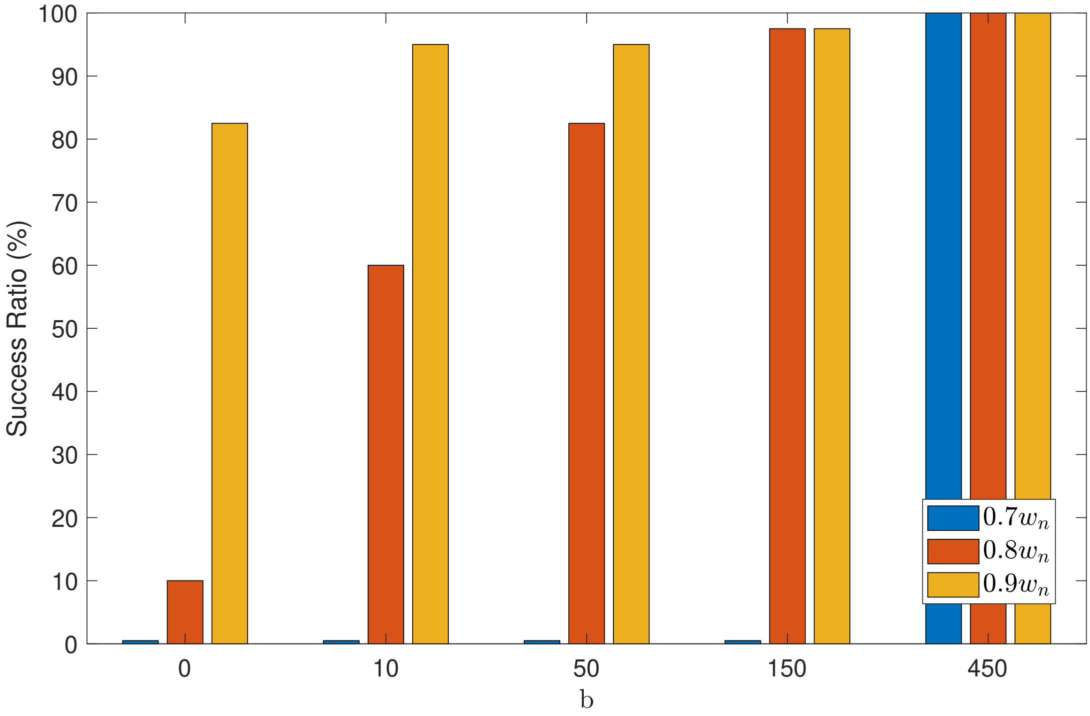


Figure 4.15: Success ratios of passing through the passage scenario, for different b and w_n values.

4.1.2.3 Reflecting from a wall scenario

Long-range links prevent the swarm to constrict and are not helpful in the narrow passage task. However, they can be helpful in other ways. Long-range links decrease the mean distance between nodes, thus they can improve the speed of information transfer between the agents.

Figure 4.17 depicts the order metric timeseries for reflecting from a wall scenario. A swarm with 169 agents and 462 NN links is used in the simulations. There are no anticipation and with anticipation cases, and also $p = 0.01$ with anticipation case. 40 different networks are used for the random network case. NN links are not broken as the random links are added, to remove the possibility of collision. This would not affect the results since the number of random links added to the system is only 5.

Mean of the lowest order metric and the respective time is shown with a cross. It can be seen that no anticipation case ($b = 0$) has the lowest order metric value, which means that it has the maximum discrepancy in the alignment between the parts of the

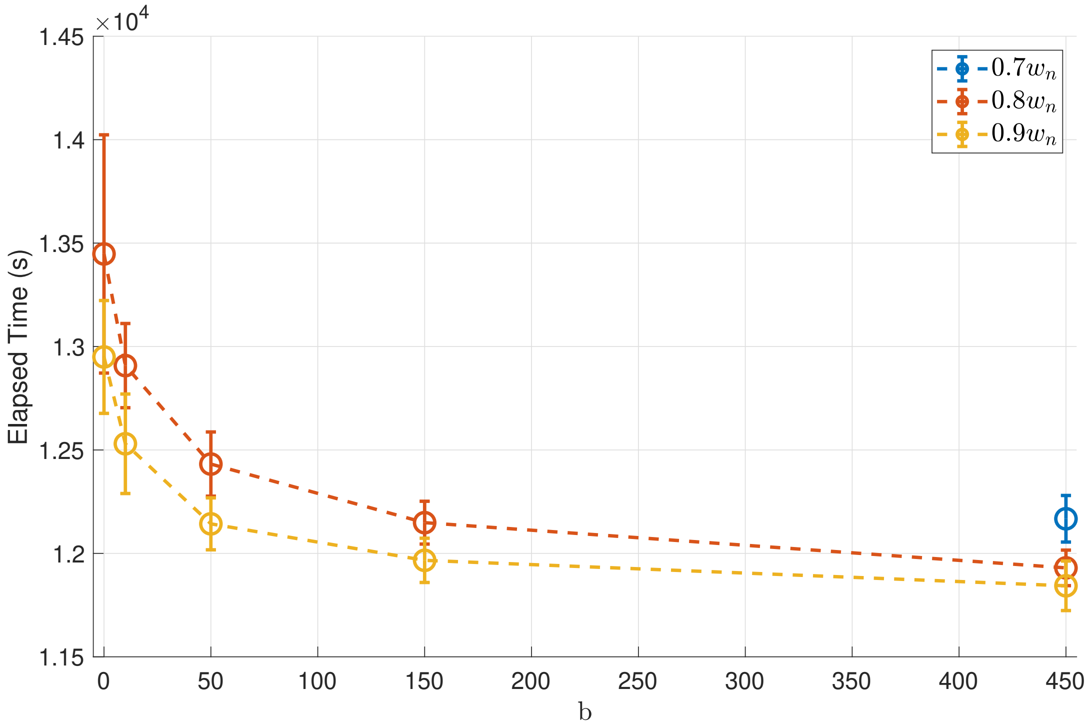


Figure 4.16: Mean and standard deviation of the elapsed times of the successful runs, for different b and w_n values.

swarm. The case with anticipation has a better lowest order metric value. The agents respond to changes without waiting for a position displacement to occur, thus they align better within a shorter time. Runs with random links have the best response time and highest alignment. Even though only 5 links are added, improvement is significant: order metric drops to 0.87 at most. Agents have a high degree of alignment, whether they are at the front or back.

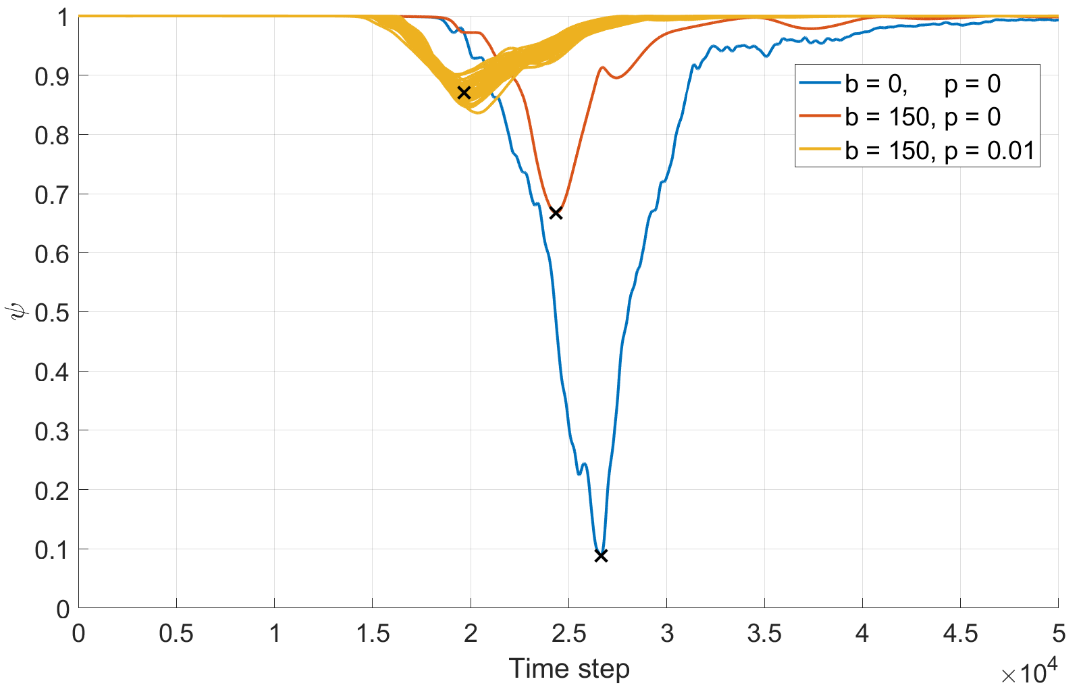


Figure 4.17: Order metric ψ values as a function of the time step, for different configurations in reflecting from a wall scenario.

4.2 Analysis of the Extended Kalman Filter for the AEA Model

In this section, the results of the experiments with EKF are given. First, performances of EKFs with and without model information are compared. Then, narrow passage experiment is conducted for a swarm with every agent running their own EKF to create motion commands.

As described in the EKF design, agents measure cartesian positions of their own and connected agents. In the experiments given in this section, the standard deviation of the position measurement is 3 centimeters. Also, agents measure their own orientation and the standard deviation of this measurement is 5 degrees. The measurement sampling period is 0.05 seconds (20 Hz as sampling frequency), which is the same period of simulation time-step.

Position measurements are used as initial values of the position state estimates. For velocity and orientation states, self-propulsion speed and orientation measurements are used. Position measurement standard deviations are used to initialize the position elements of the state covariance matrix. Initial standard deviation values used in the state covariance matrix for velocity and orientation states are $v_0/2$ and $45°$, respectively. Velocity and orientation process noise values for the focal agent are chosen 20 times smaller than the ones for the neighbors because the model of the focal agent is more accurate.

4.2.1 Comparison of the EKFs with and without Model Information

Extended Kalman Filter designs with and without model information are compared in this subsection. The self-propulsion speed of the agents in phase transition analyses is $0.002m/s$. This value is very low for a real robot. It is raised to 1m/s and other parameters are adjusted to accommodate this change. Rest of the parameters are as follows: α is 0.5, β is 0.12, κ is 5, b is 25 and K_{acc} is 0.1.

To evaluate the performance of the filters, one benchmark scenario is used. In this scenario, 7 agents move towards an obstacle and bounce off from it. Snapshots from the scenario are given in Figure 4.18. To be able to compare both designs, they should

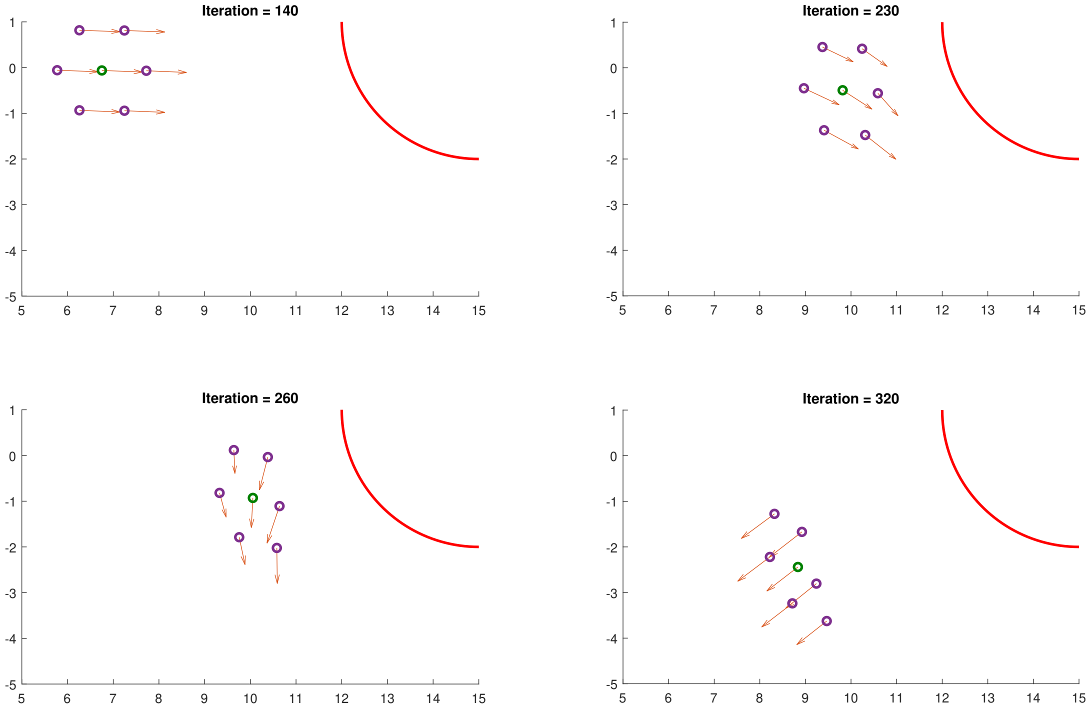


Figure 4.18: Snapshots from bouncing off from an obstacle scenario.

be tested in the same scenario with the same measurement set. Therefore, for this analysis, Kalman Filter estimations are not used in the calculation of motion commands because filters would create different motion commands, and then scenarios for each filter would be different. The AEA model with zero noise is used to update the positions and orientations of the agents.

Figure 4.19 gives the real and estimated values of velocity and orientation of the agent in the center. It can be seen that the filter without the model information tracked the states with a significant delay, compared to the filter with the model information. Position estimates are not given because positions are measured states and their estimates are close to the real values.

Figure 4.20 gives the state estimation errors and covariance matrix bounds of the filter of the center agent. In this plot, it is easier to see the effect of delay in the filter without model information. Even though the covariance bands for both filters are very close, state errors go further out of the covariance bands for the filter without the model information, which is also an indicator of modeling errors. The process noise can be increased to accommodate for unmodeled effects. Then, states would converge faster,

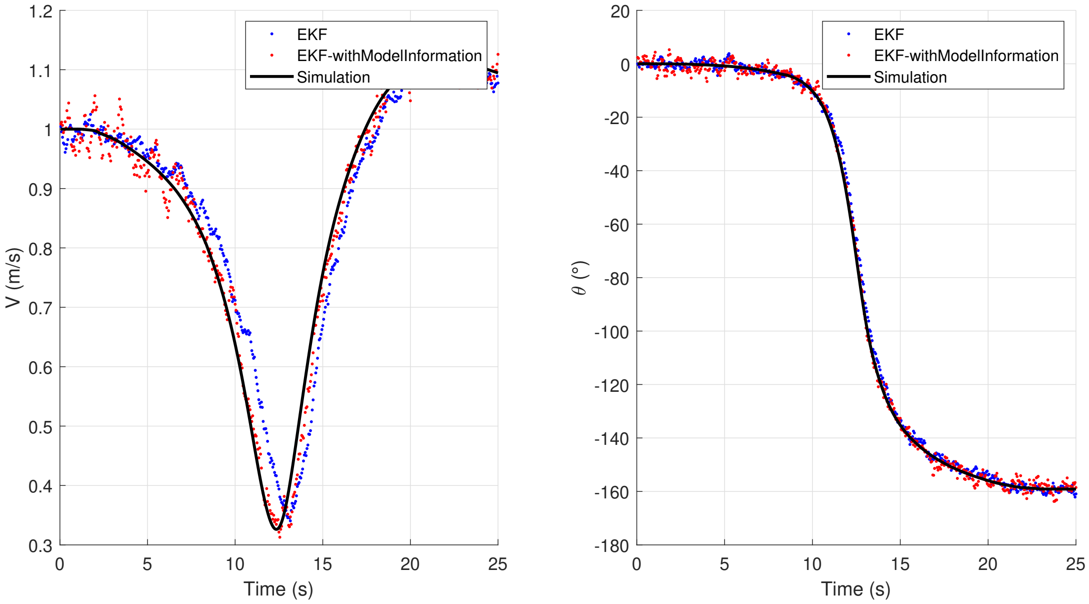


Figure 4.19: Velocity and orientation estimates of the agent in the center

but error bands would be larger. Considering these findings, it is reasonable to use the filter with model information.

Figure 4.21 gives estimation errors and covariance bands of the center agent's filter, for all states of its neighbors. As mentioned in the experimental setup, process noise for the velocity and orientation states of neighbors is higher. Covariance bands for states of neighbors are larger compared to the states of the focal agent.

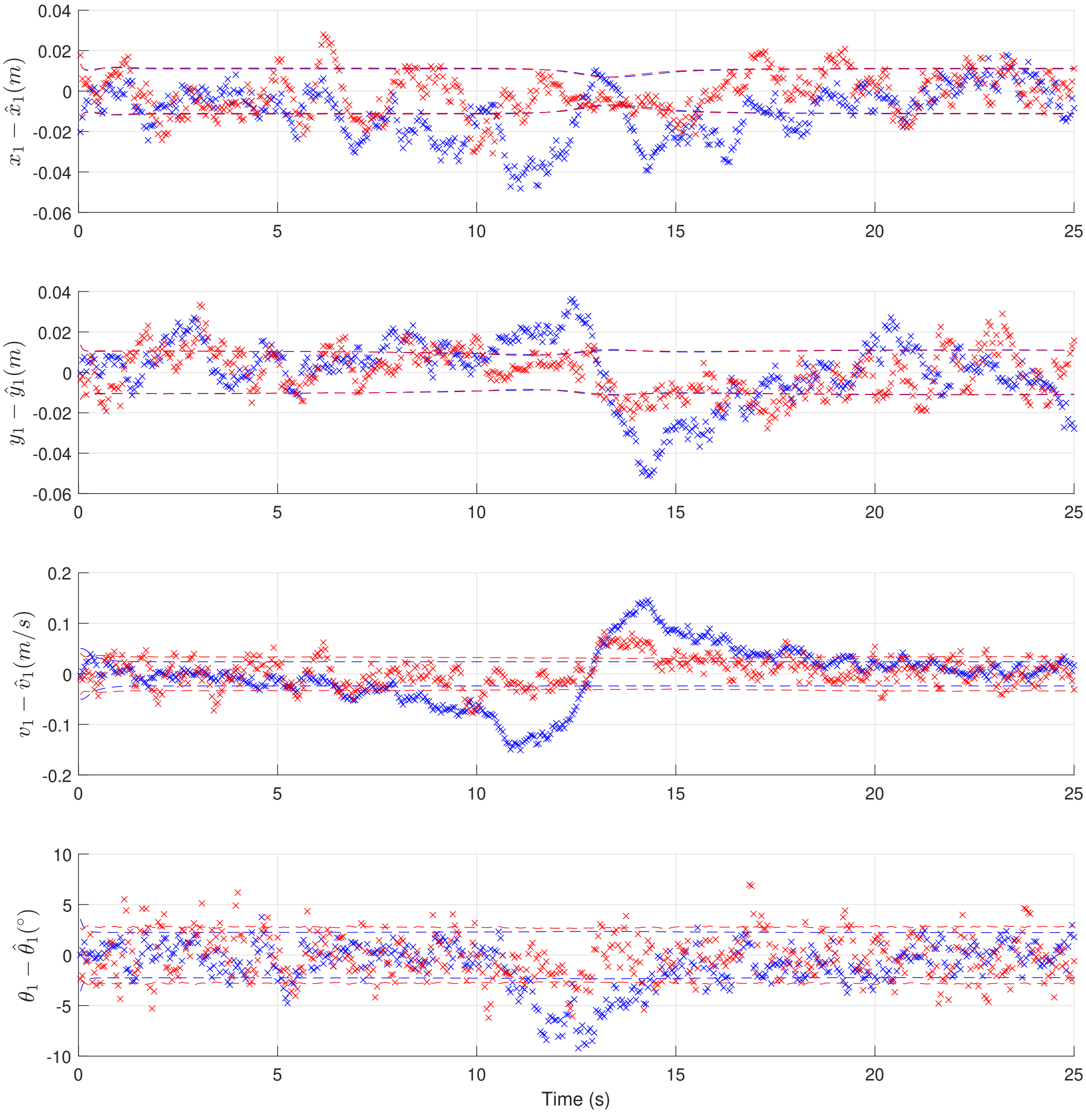


Figure 4.20: Position, velocity and orientation estimation errors of the filters with no information and with model information. Blue color is used for the filter with no information and red color is used for the filter with model information. Dashed lines represent the covariance bands calculated by the filters.

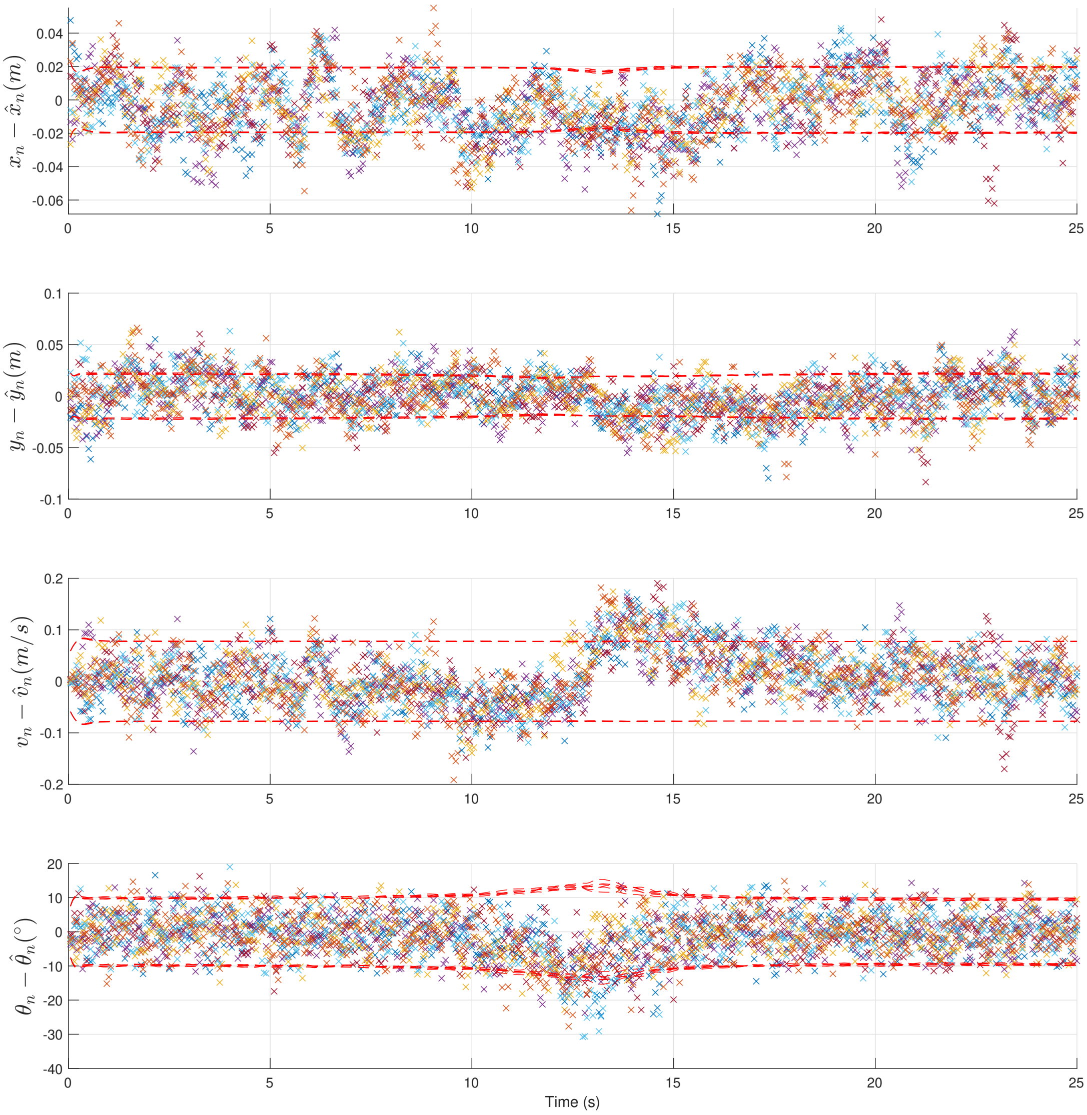


Figure 4.21: Estimation errors in states of the neighbors.

4.2.2 Narrow Passage Scenario with Extended Kalman Filter

Performance of the AEA model with the designed EKF is evaluated with a narrow passage scenario. A swarm with 19 agents goes toward a narrow passage, with a self-propulsion speed of 0.3m/s. Every agent runs its own EKF and creates motion commands using the estimates of its EKF. Separation between the obstacles is equal to 0.8 normalized width. 2 different configurations are tested: $b = 0$ and $b = 10$. Rest of the parameters are as follows: v_0 is 0.3, α is 0.1, β is 0.4, κ is 10 and K_{acc} is 0.1.

Figure 4.22 gives the snapshots from the simulation with no anticipation. Interference of the errors and delays created by the filters of each agent is not tolerated. Collective motion is lost and collisions are observed.

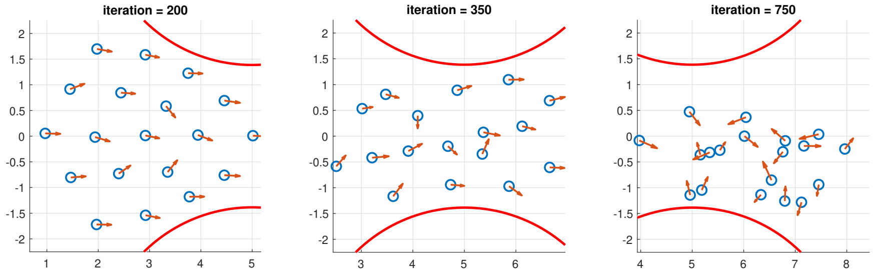


Figure 4.22: Snapshots from the scenario with no anticipation ($b = 0$).

Figure 4.23 gives the snapshots from the simulation with anticipation. It can be seen that the swarm successfully passes through the passage. Overshoots in the control of orientation and velocity disappeared, compared to the case with no anticipation. Filter errors and covariance bands of the agent in the center are given in Figure 4.24. State estimation errors are in reasonable bounds and covariance calculation of the filter is in agreement with the errors.

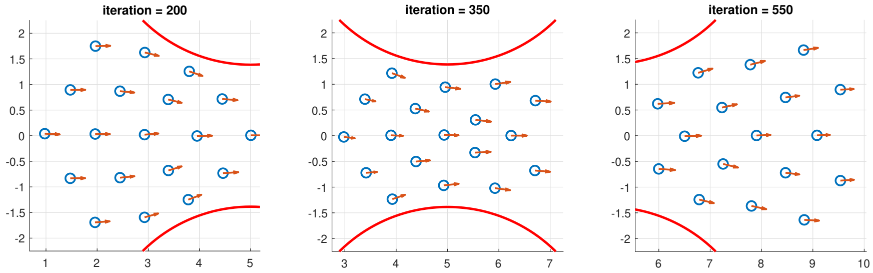


Figure 4.23: Snapshots from the scenario with anticipation ($b = 10$).

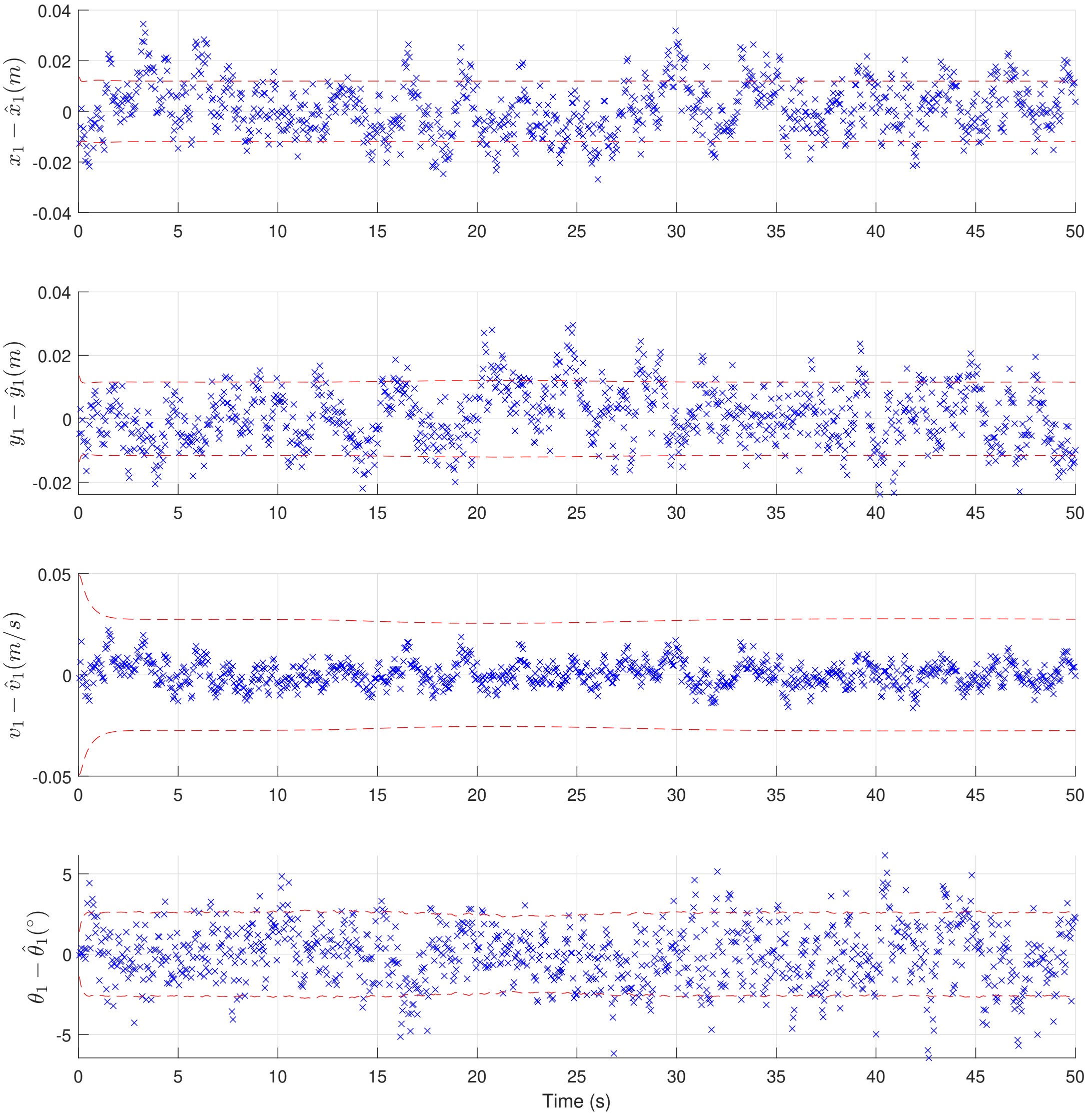
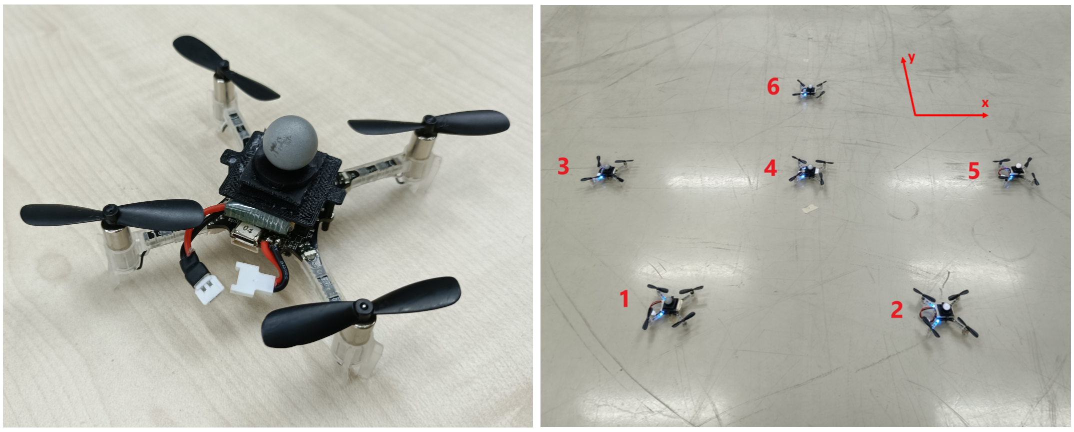


Figure 4.24: Position, velocity, and orientation estimation errors of the EKF of the center agent, for $b = 10$ case. Dashed lines represent the covariance bands calculated by the filter.

4.3 Real Robot Experiments

Crazyflie 2.x quadcopter UAVs are developed by Bitcraze AB [40]. It is an agile platform that weighs only 27 grams. It is used in many studies in the literature, and a multitude of open-source tools can be found for simulating and modeling the platform. 6 Crazyflies are used in the experiments in this thesis. 3 of the Crazyflies are version 2.1 and the other 3 are version 2.0. Each Crazyflie has an identification number for correct radio communication and starting position. Other than that, identification numbers have no significance for the experiments. Figure 4.25a gives a photograph of the Crazyflie 2.0 quadcopter UAV used in the experiments, and Figure 4.25b shows the 6 Crazyflies on the ground with their numbers.

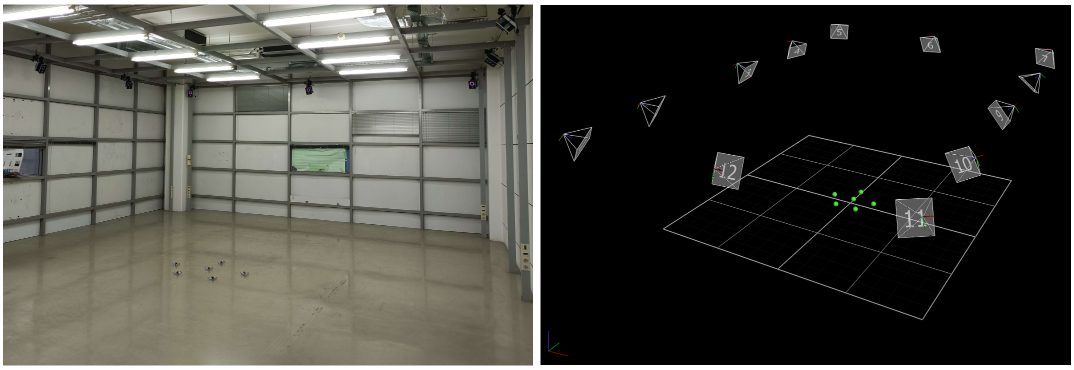


(a)

(b)

Figure 4.25: (a) A photograph of the Bitcraze Crazyflie 2.0 quadcopter UAV used in the experiments. (b) 6 Crazyflie robots on the ground. Enumeration of the robots is given in red.

Experiments with Crazyflie robots are conducted in METU ROMER UAV Laboratory. There is a Vicon Vantage [41] motion capture system in the laboratory. The system consists of 12 cameras and comes with Vicon Tracker 3.9 software for monitoring and calibration. Infrared reflecting markers are placed on each Crazyflie robot and the system tracks the position of these markers. It is compatible to use with the Crazyswarm platform. The laboratory with the Vicon Vantage motion capture system and 6 Crazyflie robots can be seen in Figure 4.26a. 3D perspective view of the test arena from the Vicon Tracker software is given in Figure 4.26b.



(a)

(b)

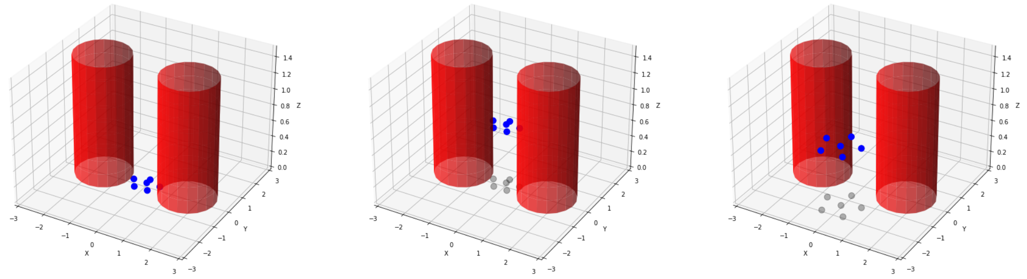
Figure 4.26: (a) Laboratory with the Vicon Vantage motion capture system and Crazyflie robots. (b) 3D perspective view of the test arena from the Vicon Tracker software

4.3.1 Crazyswarm Simulations

Crazyswarm [42] platform is used to fly Crazyflie robots, manage communication with agents and handle motion capture hardware outputs with ease. Crazyswarm platform has a simulation mode, in which the same algorithm that is developed to fly real Crazyflie robots can be tested. In simulations, measurements are created by adding a specified Gaussian noise to the simulated positions and orientations. By switching the measurement source to the real hardware, the same algorithm tested out in the simulation mode can be used to fly real Crazyflie robots.

Narrow passage experiments are repeated with 6 Crazyflie robots in Crazyswarm simulations. It is important to do the simulations first since design errors or software bugs are going to be captured in the simulations. Only successful experiments are repeated with real Crazyflie robots.

After take-off, Crazyflie robots move to their starting position, which is a pentagon with a side length of 0.8m. Enumeration of the agents can be seen in Figure 4.25b. Agent number 4 is addressed as the center agent. Distance between agents 3 and 5 is 1.3m, which is denoted as one normalized width (w_n) for Crazyflie experiments. The



(a) On the ground

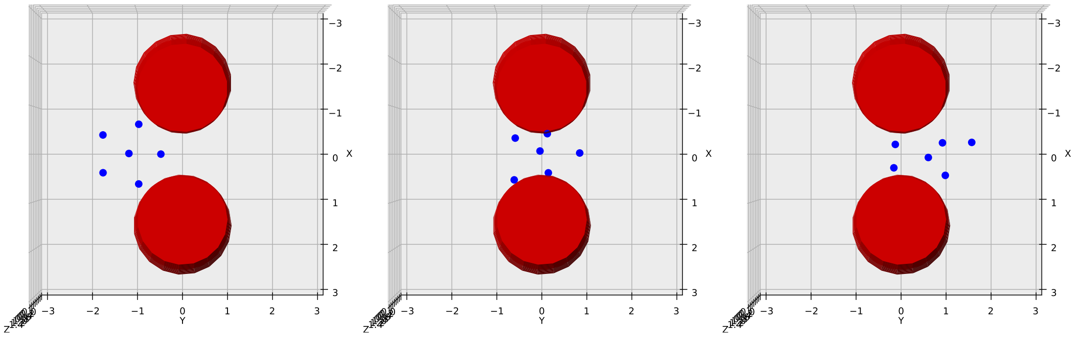
(b) Take-off

(c) Starting Position

Figure 4.27: Obstacles are shown in red, Crazyflie robots are shown in blue and their shadow after take-off is shown in gray. (a) 6 robots start from a pentagon formation. (b) robots take off from the ground and hover. (c) After take-off, robots are moved to the starting position.

swarm goes toward a narrow passage with a self-propulsion speed of 0.3m/s. There are 2 pillar-shaped obstacles with $r = 1$m inside the arena, which are placed on the x axis. Figure 4.27 depicts the simulation arena and the take-off sequence of the robots. Every agent runs its own EKF and creates motion commands using the estimates of its EKF. For the simulations, the standard deviation of the position measurement is 3 cm. Also, agents measure their own orientation and the standard deviation of this measurement is 5°. 2 different configurations are tested: no anticipation ($b = 0$) and with anticipation ($b = 10$). Rest of the parameters are as follows: α is 0.1, β is 0.4, κ is 10 and K_{acc} is 0.1.

Figure 4.28 gives the snapshots from the simulation without anticipation. Separation between obstacles is $0.7w_n$. The swarm passes through the passage but loses its order, errors created by the filters of the agents are not tolerated. Figure 4.29 gives the snapshots from the simulation with anticipation. Separation between obstacles is $0.7w_n$. Now, the swarm passes through the passage successfully. There are no oscillations in the positions and orientations.

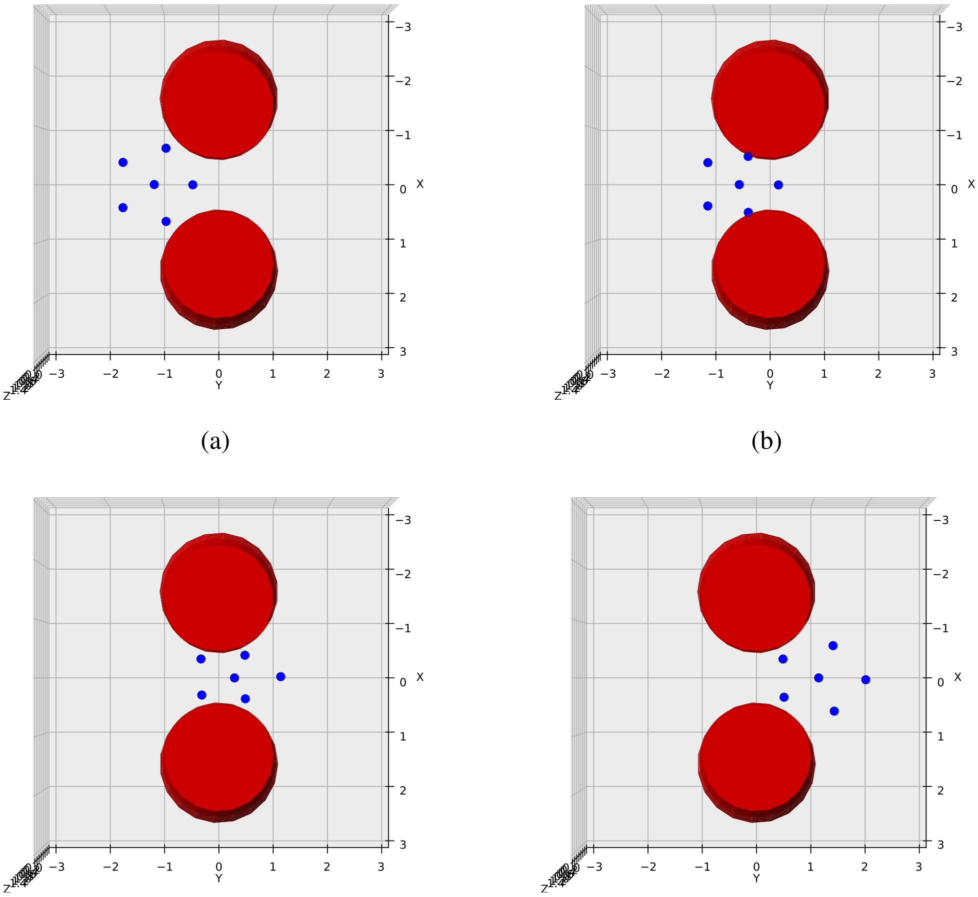


(a)

(b)

(c)

Figure 4.28: Crazyswarm simulation without anticipation. Separation between obstacles is $0.7w_n$.



(a)

(b)

(c)

(d)

Figure 4.29: Crazyswarm simulation with anticipation. Separation between obstacles is $0.7w_n$.

4.3.2 Experiments with real Crazyflie Robots

To try out the algorithm with the real Crazyflie robots, a few changes have been made. Position measurement source is switched from simulation to the Vicon motion capture system. Since the model is in 2D, the velocity command in the vertical direction is zero. A proportional altitude controller is implemented, so that expected drift in the altitude would not pose a problem. Due to a hardware failure or a software error, collisions may happen or erroneous commands may lead robots to the walls. The safety system is triggered when the distance between two agents gets smaller than 0.3m or gets larger than 2m, or any agent is out of the arena boundaries. If any of these conditions are met, all robots are landed immediately.

To identify the real system, a constant 0.3m/s velocity command in $+y$ direction is given to all Crazyflies. Velocity and orientation data derived from the motion capture system is shown in Figure 4.30. Jumps in the velocity data are due to the sampling made by the Crazyswarm platform. It can be seen that agents reach their velocity setpoint in around 1 second. Also, velocity and orientation deviations are observed due to process disturbances. These disturbances are expected in the experiments with real robots.

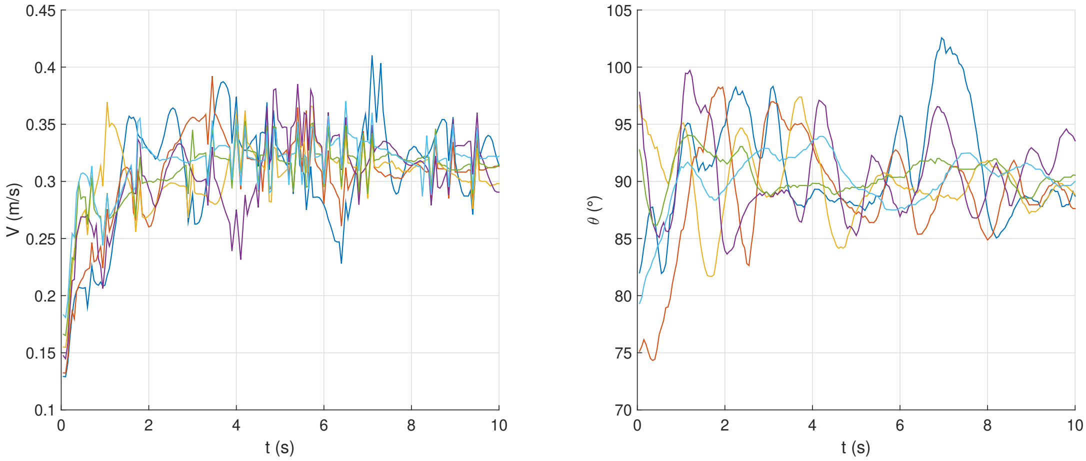


Figure 4.30: Velocity and orientation of the agents in the constant velocity scenario, derived from the motion capture system.

Then, the narrow passage scenario with anticipation, for $0.85w_n$ and $0.7w_n$ is conducted with real robots. All agents create motion commands using the estimates of their own EKF. Trajectories of the third and fifth agents for constant velocity, $0.85w_n$ and $0.7w_n$ cases are given in Figure 4.31. Also, Figure 4.32 gives the photographs from the experiment and visualization made by the Vicon Tracker software for the $0.7w_n$ case. It can be seen that the swarm constricts as much as needed depending on the obstacle positions and passes to the other side successfully.

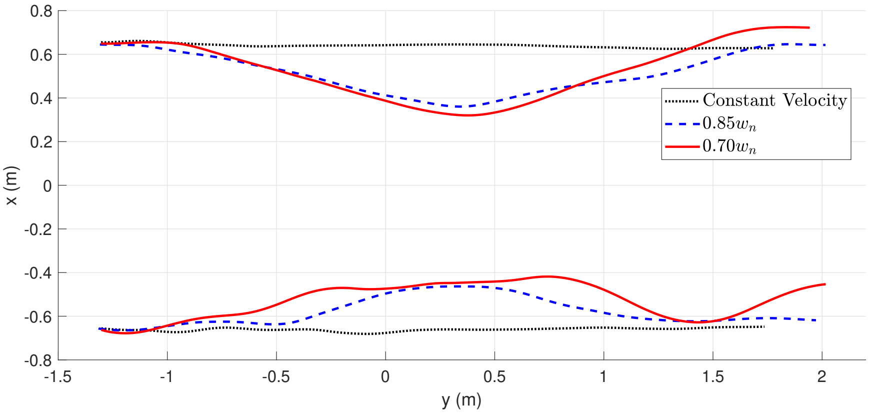
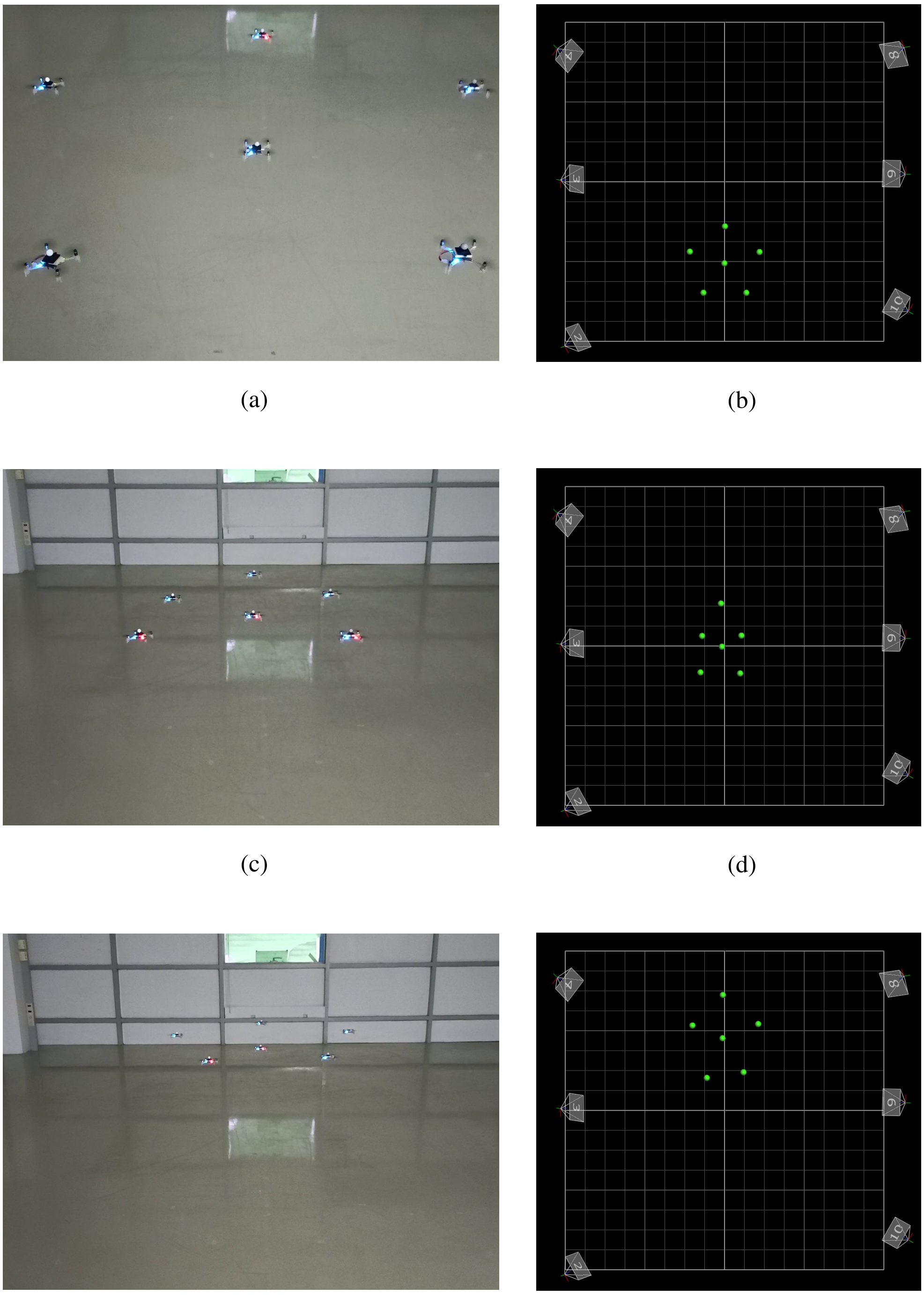


Figure 4.31: Trajectories of the third and fifth agents for constant velocity, $0.85w_n$ and $0.7w_n$ cases.

Velocity and orientation states derived from the motion capture system and estimated by the EKF are given in Figure 4.33, for the center agent in $0.7w_n$ scenario. All state estimation errors and covariance bands of the EKF of the center agent are given in Figure 4.34. Data derived from the motion capture system is used as the ground truth. Errors of the state estimations are higher than the particle agent and Crazyswarm simulations, but a discrepancy due to process disturbance is expected as shown in Figure 4.30. Covariance bands calculated by the EKF are consistent with the estimation errors.

Lastly, an experiment is conducted to test the performance of the filter in a scenario where the orientations of the agents change rapidly. In this experiment, agents bounce off a wall. Figure 4.35 gives the photographs from the experiment and visualizations made by the Vicon Tracker software. Trajectories of the agents can be seen in Figure 4.36.

Velocity and orientation states of the center agent in bouncing off a wall scenario, derived from the motion capture system measurements and EKF estimates, are given in Figure 4.37. Orientation of the center agent starts at $60°$ and goes to $-50°$ after bouncing off the wall. The EKF tracks the velocity and orientation states adequately. All state estimation errors and covariance bands of the EKF of the center agent are given in Figure 4.38. Data derived from the motion capture system is used as the ground truth. Estimation errors in the orientation states are higher than the narrow passage scenario but it does not pose a problem to the cohesiveness of the swarm.



(a) (b)

(c) (d)

(e) (f)

Figure 4.32: Snapshots from the $0.7w_n$ case. Left panels are photographs from the real experiment, and right panels are visualizations from the Vicon Tracker software.

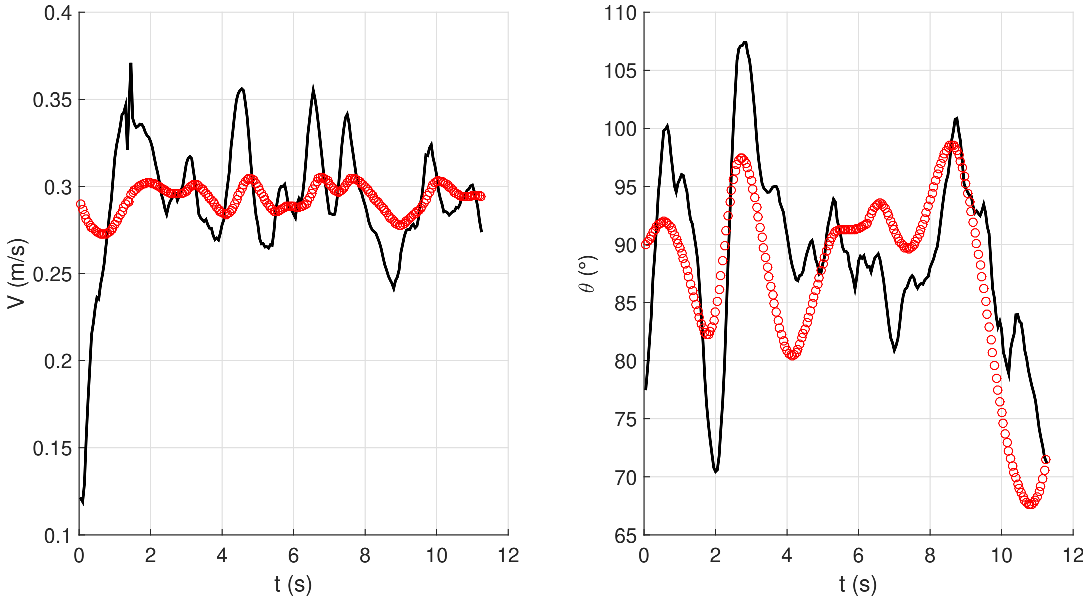


Figure 4.33: Velocity and orientation states of the center agent in $0.7w_n$ scenario. Data derived from the motion capture system measurements is shown in black and estimation by the EKF of the center agent is shown in red.

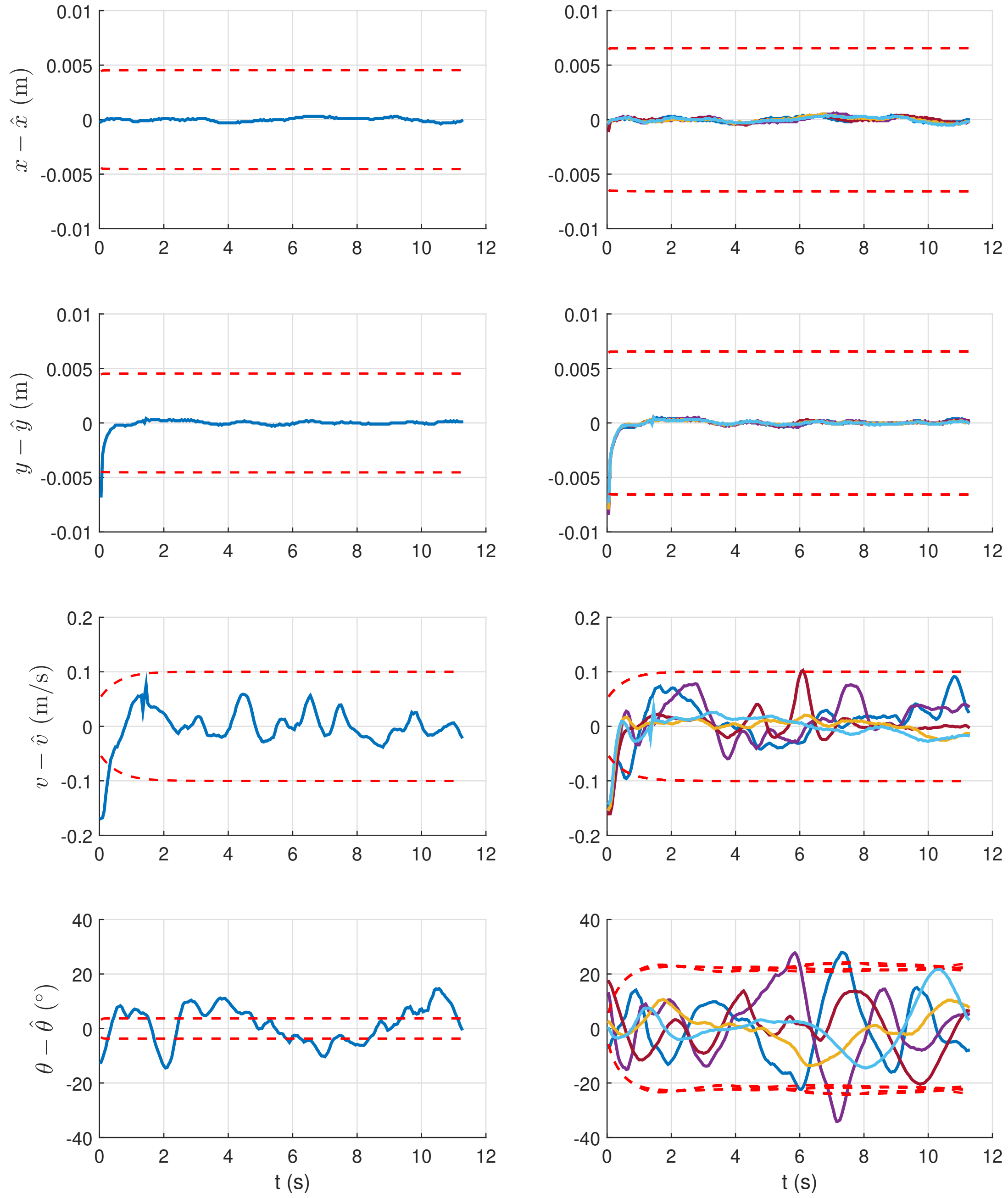
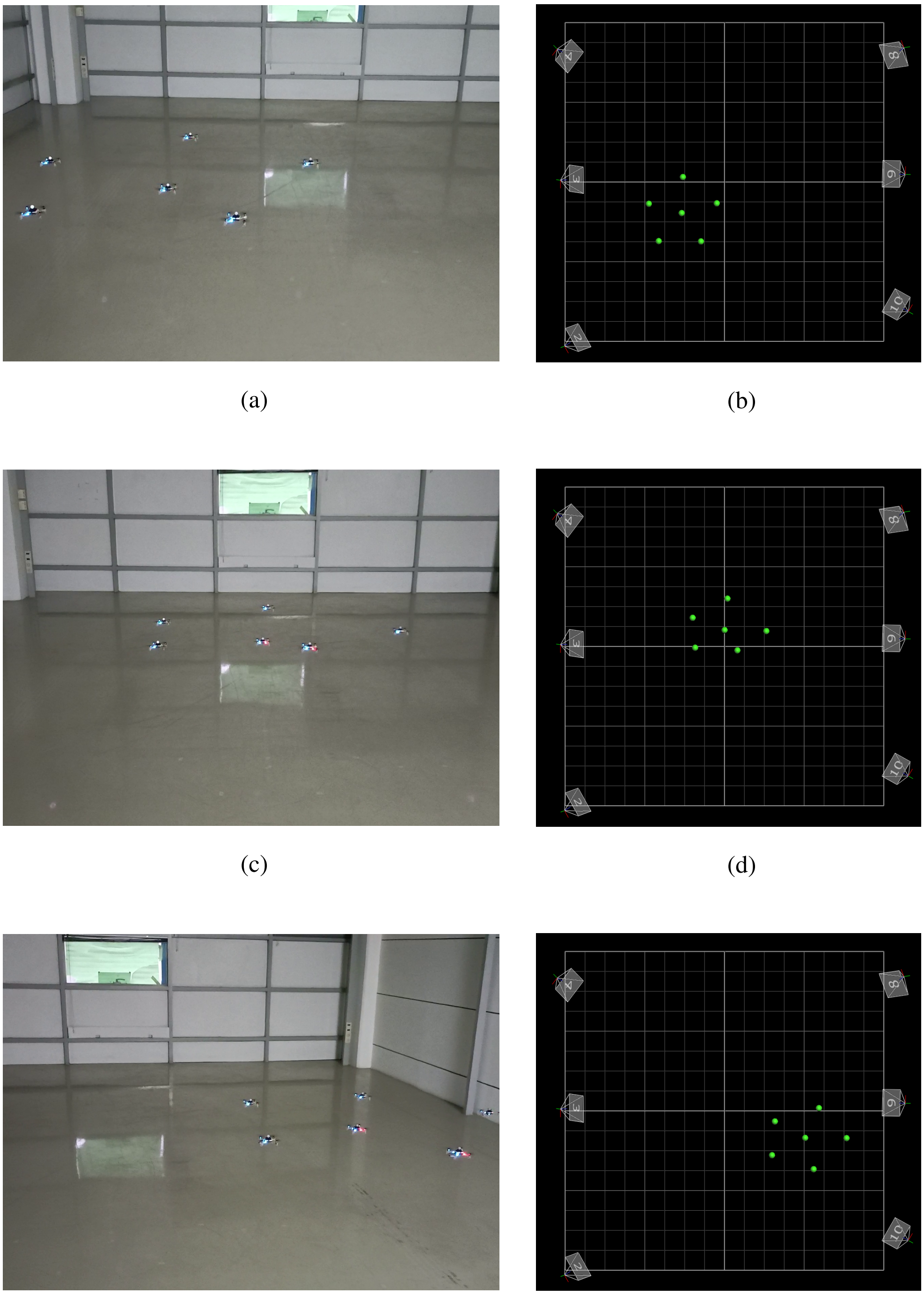


Figure 4.34: State estimation errors and covariance bands of the EKF of the center agent, for $0.7w_n$ scenario. Left panels are for the states of the center agent and right panels are for the states of its neighbors.



(a) (b)

(c) (d)

(e) (f)

Figure 4.35: Snapshots from the bouncing off a wall scenario. Left panels are photographs from the real experiment, and right panels are visualizations from the Vicon Tracker software.

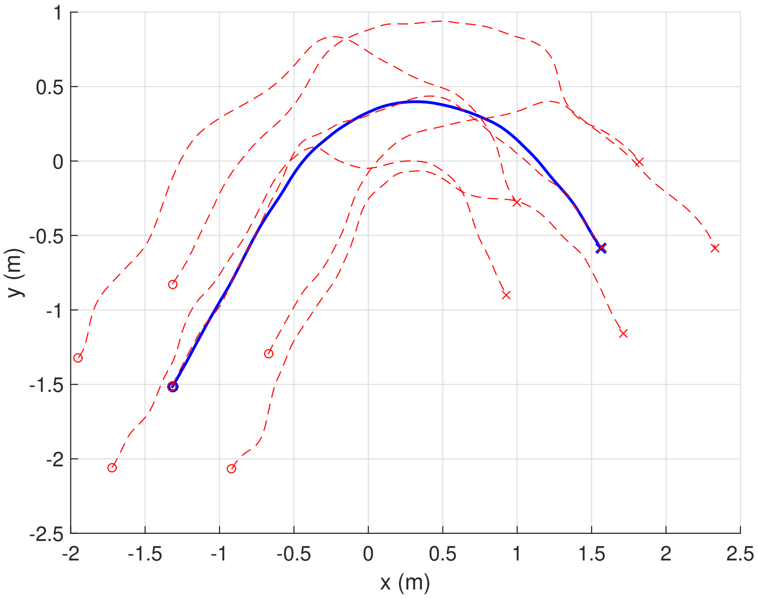


Figure 4.36: Trajectories of the agents in bouncing off a wall experiment. Red dashed lines give the individual trajectories and blue solid line gives the trajectory of the center of the swarm. Starting points are denoted with 'o' and final points are denoted with 'x'.

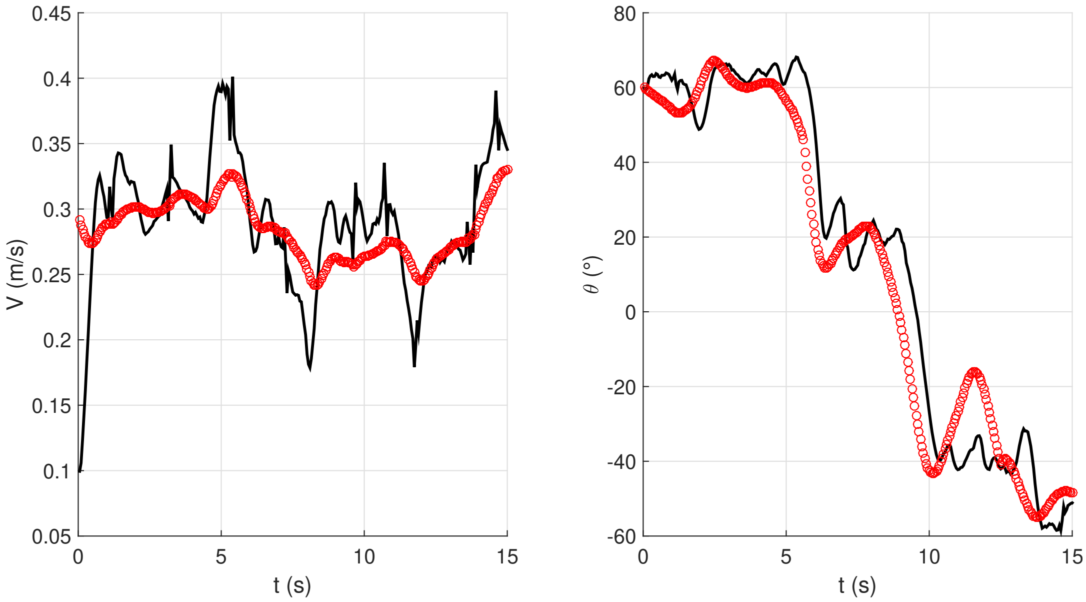


Figure 4.37: Velocity and orientation states of the center agent in bouncing off a wall experiment. Data derived from the motion capture system measurements is shown in black and estimation by the EKF of the agent is shown in red.

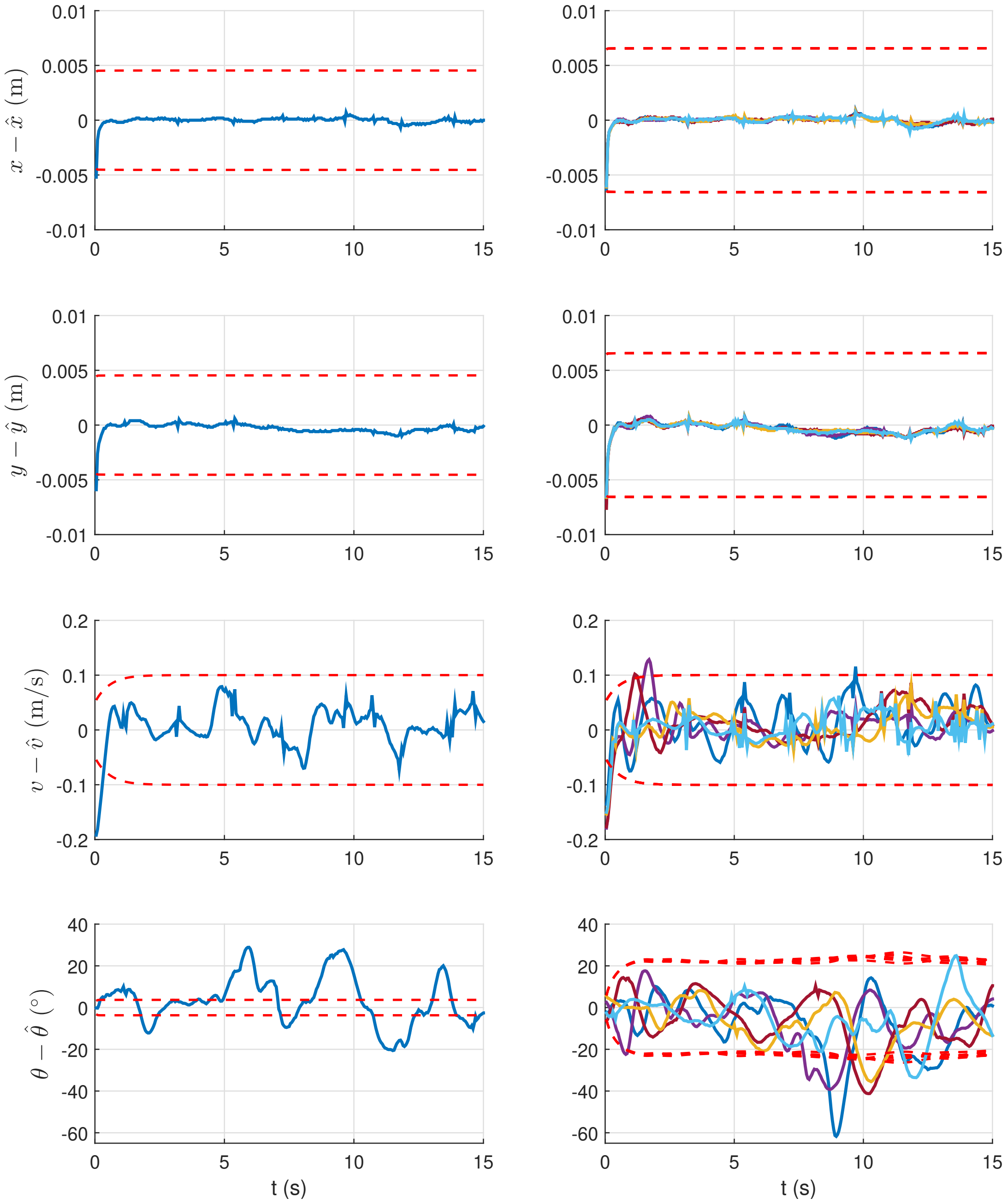


Figure 4.38: State estimation errors and covariance bands of the EKF of the center agent, in bouncing off a wall experiment. Left subplots are for the states of the center agent and right subplots are for the states of its neighbors.

CHAPTER 5

DISCUSSION

In this chapter, results of the experiments are discussed. Also, methodology and the experimental setup are criticized. First, particle agent model simulation results are studied. Then, EKF design and results are discussed. Lastly, simulations and experiments with Crazyflie quadcopter UAVs, which bring the AEA model and the EKF to the real world, are discussed.

5.1 Active-Elastic and Vicsek-Network Model Results

The presented results show that the structure of the interaction network affects very differently the resilience to noise of the ordered state in the VN model and the AE model. In the VN case, a larger fraction of random links with either Poissonian or power-law degree distribution always increases the critical noise. In the AE case, similar results are obtained only when interpolating between NN and random ER network topologies. When interpolating instead between NN and random SF networks, a larger fraction of random links can reduce the critical noise. As mentioned above, this is contrary to the common intuition that more long-range connections tend to improve the collective behavior of networked systems by decreasing the mean topological distance between nodes, which improves the system-wide propagation of information. This is believed to be especially true when these connections follow a SF topology, because this connectivity further reduces the mean distance between nodes.

A heuristic explanation of this phenomenon can be deduced from the difference in the self-organization mechanism that leads to collective motion in each model. The self-organizing dynamics of the VN model is based on a distributed average consensus process, where local orientation averages of the consensus variable propagate directly through the network. In the AE model, however, only the positional information is propagated through the network, while the consensus variable is still the orientation. This suggests that the self-organizing mechanism for the AE model must be different. Indeed, as detailed in [7, 8], the AE model self-organizes by focusing the self-propulsion energy into low-energy modes, through a combination of a well-known property of all elastic systems and the coupling (imposed by its dynamical equations) between the elastic forces and the individual heading directions. It is well known that higher energy modes dampen at a faster rate than slower ones in all elastic systems, because their higher rigidity produces higher oscillation frequencies that dissipate faster. The AE model is an active elastic system, however, where each agent is also continuously injecting energy at the individual level through its self-propulsion term, so motion cannot dampen out. Instead, the elastic forces will tend to steer agents away from the higher modes, which require more energy to excite. The self-propulsion energy will thus be channeled to lower and lower modes, until the first (rotational) mode or the zero (translational) mode is reached and collective motion is achieved.

This explains why SF networks can lead here to weaker ordered states. Previous works had only considered homogeneous elastic systems with NN interactions, in which low-energy elastic modes correspond to large scales of coherent motion. In complex, networked elastic systems, however, this correspondence does not hold, since there can be highly localized and disordered low-energy modes that do not correspond to any large-scale collective dynamics. In particular, the SF networks have an overabundance of agents with few connections, and therefore low elastic constraints, which results in multiple disordered low energy modes. For example, there will be a majority of agents with only two links, which can be easily excited at very low energy levels.

The discussion above helps understanding why disordered modes can be easily excited by noise in the AE model with SF interactions. This in turn justifies the lower critical noise that is observed for the AE model as more random SF interaction links are included, and when considering SF interaction networks that are larger or have steeper degree distributions.

5.2 Active-Elastic-Anticipation Model Simulation Results

Phase transition diagrams, which consist of the order metric (ψ) with respect to noise intensity (η), are useful for the classification of the collective motion models. Phase transition of the AE model is sharp and has a discontinuity in the first-order derivative. Therefore, the transition is classified as first order. For the VN model, order transitions to disorder continuously and it has a discontinuity in the second-order derivative. Therefore, the transition is classified as second order. Equations of motion for the AE model are position-based and for the VN model are orientation-based. This difference between the models is reflected in the phase transition diagram.

In Figure 4.10, the same analysis is conducted for the AEA model with different damper coefficient (b) values. First of all, for $b = 0$ case, the only difference between the AE model and the AEA model is inertia. The AEA model agents have inertia and their velocity can not change instantly. However, the behavior of the model with respect to the noise intensity is not different from the AE model. This is an important outcome because it proves that the addition of inertia did not create diverse complications and the AEA model behaves similarly to the well-established AE model. In this work, velocity is controlled with a proportional controller, and inertia is lumped with the proportional gain. A high proportional gain is helpful for reaching the velocity setpoint faster but as revealed in the eigenvalue analysis, it may bring oscillations and also instability after a certain point. On the other hand, a low gain makes the agents insensitive to the velocity commands. This would disrupt the consensus dynamic in the swarm and order would be lost. This parameter allows the design of velocity dynamics that are more suitable to the physical robots, as long as the limits are considered.

As the damper coefficient increases, the anticipation horizon increases and therefore velocity-based interactions are expected to be stronger. When the damper coefficient is increased to 50, the transition to disorder is not sharp as it was in the $b = 0$ case. And for $b = 150$ case, the transition is much similar to the VN model. Also, Figure 4.11 shows that bi-modality is lost as the anticipation horizon increases. These results show that interactions become predominantly orientation-based when the anticipation horizon is increased.

Phase transition diagrams of the AEA model for different damper coefficient b and topological control parameter p values are given in Figure 4.12. Similar analyses are conducted for AE and VN models and critical noise η_c is presented as a function of p. Criteria for identifying the critical noise differs for the AE model and the VN model, since their phase transition characteristics are different. For the AEA model, b parameter allows the continuous transition between first-order and second-order systems. Therefore, either separate criteria should be chosen for each b value, or a single criterion that can be justified for all b values should be created. Setting up critical noise criteria and critical noise analysis for the AEA model is left as future work. Still, the AEA model with $b = 0$ diagram shows that long-range interactions increase the resilience to noise, same as the AE model. For $b = 50$ and $b = 150$, networks with long-range interactions have better resilience against noise compared to the NN network. However, it can be seen that fully random network ($p = 1$) has performed worse than the interpolated network ($p = 0.4$). This kind of effect is also displayed in the p analysis for the AE model with the SF network. Detailed investigation of this result is a valuable future work topic to better understand the AEA model.

The anticipation parameter opens up a new dimension in the design space. Figure 4.13 and Figure 4.14 shows a scenario where anticipation is beneficial. It can be seen that the AEA model is more suitable when the task is to go through a narrow passage. For a fair comparison, the spring constants of the models are the same, the only difference is the addition of anticipation.

A systematic analysis is conducted to show the effect of b in a narrow passage scenario. Figure 4.15 shows the success ratio as a function of b, in different passage widths. The ratio of the successful runs for $0.8w_n$ case increases significantly as b value grows. When the separation between obstacles is reduced to $0.7w_n$, only $b = 450$ is able to successfully pass through. As the hexagon shape with the NN network is compressed, links between agents become more parallel to their orientation. Therefore, their ability to repel each other in the constrained direction decreases and rows of the swarm fold onto themselves. This experiment can be repeated with a square lattice with 8 connections. Also, instead of the static NN network, a proximity-based dynamic network can be also used, such as the one in the original Vicsek model.

This way, the swarm can be designed to exhibit more fluid-like motion, which can be helpful for the swarm to assume the shape of the constraining environment. Nonetheless, it can be seen that anticipation is beneficial in passing through the passage task. Figure 4.16 gives the mean and standard deviation of the elapsed times of the successful runs. The swarm bumps into the walls of the obstacle more as the passage width decreases. This creates velocity fluctuations in the group, which in turn reduces the mean velocity towards the exit. Therefore, for the same b value, mean elapsed time increases as the passage width decreases. For the same passage width, increasing b decreases the mean elapsed time. As b increases, velocity fluctuations become smaller, and the swarm passes through the passage with more fluid-like motion and less friction with the obstacles.

Shrinking the swarm to a smaller size would create significantly more displacement in the long-range links, compared to the displacement in the links between the neighbors. For this reason, long-range links prevent the swarm to shrink and are not useful in the narrow passage scenario. However, long-range links decrease the mean distance between nodes and in turn bring faster information transfer in the swarm. Figure 4.17 gives the order metric ψ in the reflecting from a wall scenario. It can be seen that anticipation improves both the response time and the lowest order metric value. This result is in line with the brake light analogy for anticipation. Agents react to the velocity change immediately and response time is improved. Additionally, aligning effect of the anticipation helps with keeping the order metric high. Case with anticipation and 5 long-range links has the shortest response time and highest order metric. 40 different networks are used, to randomize the location of the long-range links in the case with anticipation and 5 long-range links. On average, they have the shortest response time and the order metric drops only to 0.87. With the NN network, agents in the front change their orientations, and the changes are transmitted towards the back of the group with the neighboring links. During this process, agents at the front and agents at the back may have different orientations, thus the order metric has a low value. With only 5 long-range links, the mean distance between the nodes is much shorter and all agents in the swarm have high alignment with each other.

5.3 Extended Kalman Filter Design and Results

Since there are linear dampers in the model, agents need to know the relative velocities and headings of the other agents. However, most of the measurement systems for robots are position-based. The majority of the robot swarm implementations in the literature depend on communication to obtain the heading information. In this thesis, an EKF that uses position and self-heading measurements is designed. This way, heading and velocity state estimates are constructed without communication or a specific measurement system.

Type of the filter used in this thesis is the Extended Kalman Filter. The EKF requires the partial derivatives of the state matrix with respect to the current states. In the formulation, partial derivatives of the agent to agent distance with respect to the positions are neglected because they are mostly used in the scaling of the unit vectors. Proposed implementation works adequately, but there are other filtering techniques that are used when the system of interest has a highly nonlinear system or measurement model. Unscented Kalman Filter (UKF) and Particle Filter are good examples. Comparing the performance of the UKF to the existing EKF is an interesting topic for future work.

Figure 4.18 depicts a scenario where agents bounce off an obstacle. The scenario is designed this way so that the velocities of the agents change rapidly. Figure 4.19 shows the rapid change in velocity and orientation. Figure 4.20 shows that the filter with the model information performs better than the generic filter. Estimation errors in the states of the neigbors are given in Figure 4.21. The current design is satisfactory for this scenario, but process noise values can be adjusted for a better fit in a different scenario.

Snapshots from the narrow passage scenario where every agent creates motion commands using its own EKF estimations are given in Figures 4.22 (no anticipation) and 4.23 (with anticipation). When motion commands are created by the EKFs of the agents, the model with no anticipation loses its order and collisions occur. Errors of the agents do not damp out, they interfere and amplify each other. However, the model with anticipation has no such problem, the swarm passes through the passage with ease. Anticipation is not only helpful for the passage scenario, it also helps with damping out the disturbances created by the agents in the swarm. State estimation errors and covariance bands of the center agent for the $b = 10$ case are given in Figure 4.24. Orientation and position state estimations are better than the measurements. Velocity state estimation has no bias and the variance is tolerable.

5.4 Real Robot Experiments

Crazyswarm [42] platform has a Python API for writing complex algorithms easily for the Bitcraze Crazyflie 2.x quadcopters. The AEA model and the designed EKF are implemented in Python and the written algorithm is evaluated on the Crazyswarm platform's simulation mode. Figure 4.28 gives similar findings to the EKF analysis with particle agent simulations, the model with no anticipation can not stay cohesive when commands are created from the EKFs of the agents. However, the model with anticipation given in Figure 4.29 passes through without losing order or having collisions.

Real Crazyflie robots are subject to process disturbances; they can not stay perfectly still at a point or they can not fly in a perfectly straight line. The system model of the EKF assumes that the agent would move in its current orientation. However, a quadcopter is a holonomic system and disturbances may create movements in any direction. This can be detrimental to the performance of the EKF. The swarm is ordered to move with a constant velocity in a constant direction, and the actual velocities and orientations of the agents are given in Figure 4.30. The same amount of deviation is expected in the experiments with the EKF. Also, care must be taken when the velocity is smaller because disturbances would affect the orientation more.

Narrow passage experiments with passage widths $0.85w_n$ and $0.7w_n$ are conducted with real Crazyflie robots. Robots pass between two virtual objects with the proposed anticipative collective motion model. Photographs from the experiment and top view from the Vicon tracker software are given in Figure 4.32. Each agent uses position and self-heading measurements and creates motion commands using the erroneous state estimations. There is also process disturbance on each agent. Nonetheless, designed

EKF is able to guide the real Crazyflie robots through the passage collectively.

To further prove that the anticipative motion model and the EKF works, another experiment where the orientations of the agents change rapidly is conducted. Photographs from the experiment and top view from the Vicon tracker software are given in Figure 4.35. State estimation errors and covariance bands of the EKF of the center agent are given in Figure 4.38. Even though the errors in orientation estimations are higher than the narrow passage scenario, the swarm is still coherent and able to reflect from the wall collectively.

Overall, it can be said that anticipation is helpful for the stability of the swarm and its advertised properties hold in the Crazyswarm simulations and real flight tests. The concept of the anticipative collective model with agents running EKF to create motion commands is proven with real flight tests.

CHAPTER 6

CONCLUSIONS

In this thesis, a novel anticipative collective motion model, which is named as Active-Elastic-Anticipation model, is proposed. The phase transition diagram of this model is compared with 2 other well-known collective motion models: position-based Active-Elastic model and heading-based Vicsek model. The comparison shows that anticipation brings a new term that depends on the relative velocities of the agents and this term aligns the headings of the agents. With the help of anticipation, the swarm was able to pass through a narrow passage. The anticipative term in the AEA model depends on the relative velocities of the agents. Since most of the measurement systems are position-based, acquiring the velocity and orientation information of the other agents is not an easy task. An Extended Kalman Filter that takes the equations of motion for the AEA model into consideration is designed to estimate the velocity and heading states of the agents in the swarm. The designed filter allows the implementation of the proposed model in the real world with basic sensors. Also, when motion commands of the agents come from their own EKF, the system without anticipation is not able to stay cohesive. Anticipation helps damp out the oscillations created by the estimation errors of the EKFs. The proposed model and the designed filter are implemented in the Crazyswarm platform, which is used for flying and simulating the Crazyflie 2.x quadcopter UAVs. The same benefits of the anticipative effect are successfully displayed in the simulations in the Crazyswarm platform and the designed EKF worked adequately. Then, real flight tests with Crazyflie UAVs are conducted and results were similar to the simulations. Using all of the concepts proposed in this thesis, UAVs that use simple measurements were able to run EKF and collectively move with an anticipative model to pass through a narrow passage in the real world.

REFERENCES

[1] C. Reynolds, "Boids (flocks, herds, and schools: a distributed behavioral model)." `https://www.red3d.com/cwr/boids/`, Last accessed on 2021-08-15.

[2] R. Rosen, J. Rosen, J. J. Kineman, and M. Nadin, *Anticipatory Systems: Philosophical, Mathematical, and Methodological Foundations*. IFSR International Series on Systems Science and Engineering, Springer New York, 2014.

[3] M. J. Berry, I. H. Brivanlou, T. A. Jordan, and M. Meister, "Anticipation of moving stimuli by the retina," *Nature*, vol. 398, pp. 334–338, mar 1999.

[4] J. Voevodsky, "Evaluation Of A Deceleration Warning Light For Reducing Rear-End Automobile Collisions," Tech. Rep. 3, 1974.

[5] F. J. Roethlisberger and W. J. Dickson, *Management and the Worker*, vol. 5. Psychology press, 2003.

[6] P. Stahl, B. Donmez, and G. A. Jamieson, "Anticipation in driving: The role of experience in the efficacy of pre-event conflict cues," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 5, pp. 603–613, 2014.

[7] E. Ferrante, A. E. Turgut, M. Dorigo, and C. Huepe, "Elasticity-based mechanism for the collective motion of self-propelled particles with springlike interactions: A model system for natural and artificial swarms," *Physical Review Letters*, vol. 111, p. 268302, dec 2013.

[8] E. Ferrante, A. E. Turgut, M. Dorigo, and C. Huepe, "Collective motion dynamics of active solids and active crystals," *New Journal of Physics*, vol. 15, p. 095011, sep 2013.

[9] M. Nagy, Z. Ákos, D. Biro, and T. Vicsek, "Hierarchical group dynamics in pigeon flocks," *Nature*, vol. 464, no. 7290, pp. 890–893, 2010.

[10] Y. Katz, K. Tunstrøm, C. C. Ioannou, C. Huepe, and I. D. Couzin, "Inferring the structure and dynamics of interactions in schooling fish," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 46, pp. 18720–18725, 2011.

[11] A. Sokolov, I. S. Aranson, J. O. Kessler, and R. E. Goldstein, "Concentration dependence of the collective dynamics of swimming bacteria," *Physical Review Letters*, vol. 98, no. 15, 2007.

[12] J. Deseigne, O. Dauchot, and H. Chaté, "Collective motion of vibrated polar disks," *Physical Review Letters*, vol. 105, no. 9, pp. 1–4, 2010.

[13] A. Kudrolli, G. Lumay, D. Volfson, and L. S. Tsimring, "Swarming and swirling in self-propelled polar granular rods," *Physical Review Letters*, vol. 100, no. 5, pp. 2–5, 2008.

[14] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 25–34, 1987.

[15] T. Vicsek, A. Czirk, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, pp. 1226–1229, aug 1995.

[16] F. Cucker and S. Smale, "On the mathematics of emergence," *Japanese Journal of Mathematics*, vol. 2, no. 1, pp. 197–227, 2007.

[17] F. Cucker and S. Smale, "Emergent behavior in flocks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 852–862, 2007.

[18] H. Chaté, F. Ginelli, G. Grégoire, F. Peruani, and F. Raynaud, "Modeling collective motion: Variations on the Vicsek model," *European Physical Journal B*, vol. 64, no. 3-4, pp. 451–456, 2008.

[19] F. Cucker and J. G. Dong, "Avoiding collisions in flocks," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1238–1243, 2010.

[20] A. Morin, J. B. Caussin, C. Eloy, and D. Bartolo, "Collective motion with anticipation: Flocking, spinning, and swarming," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 91, jan 2015.

[21] H. Murakami, T. Niizato, and Y. P. Gunji, "Emergence of a coherent and cohesive swarm based on mutual anticipation," *Scientific Reports*, vol. 7, apr 2017.

[22] P. Gerlee, K. Tunstrøm, T. Lundh, and B. Wennberg, "Impact of anticipation in dynamical systems," *Physical Review E*, vol. 96, dec 2017.

[23] R. Shu and E. Tadmor, "Anticipation Breeds Alignment," *Archive for Rational Mechanics and Analysis*, vol. 240, pp. 203–241, apr 2021.

[24] G. Lin, Z. Han, and C. Huepe, "Order-disorder transitions in a minimal model of active elasticity," *New Journal of Physics*, vol. 23, feb 2021.

[25] E. Ferrante, A. E. Turgut, A. Stranieri, C. Pinciroli, M. Birattari, and M. Dorigo, "A self-adaptive communication strategy for flocking in stationary and non-stationary environments," *Natural Computing*, vol. 13, no. 2, pp. 225–245, 2014.

[26] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugniere, G. Di Caro, F. Ducatelle, E. Ferrante, A. Förster, J. M. Gonzales, J. Guzzi, V. Longchamp, S. Magnenat, N. Mathews, M. Montes De Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Rétornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard, "Swarmanoid: A novel concept for the study of heterogeneous robotic swarms," *IEEE Robotics and Automation Magazine*, vol. 20, no. 4, pp. 60–71, 2013.

[27] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, pp. 1–14, 2018.

[28] H. Balanji, E. Yilmaz, O. Cakmak, and A. E. Turgut, "Sürü robotları için görüntü tabanlı mesafe, açı ve yönelim sistemi tasarımı," in *TOK*, pp. 386–391, 2017.

[29] S. I. Roumeliotis and G. A. Bekey, “Distributed multirobot localization,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[30] N. Karam, F. Chausse, R. Aufrère, and R. Chapuis, “Cooperative multi-vehicle localization,” *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 564–570, 2006.

[31] A. Cornejo and R. Nagpal, “Distributed range-based relative localization of robot swarms,” *Springer Tracts in Advanced Robotics*, vol. 107, pp. 91–107, 2015.

[32] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley, “Classes of small-world networks,” *Proceedings of the National Academy of Sciences*, vol. 97, no. 21, pp. 11149–11152, 2000.

[33] A. Clauset, C. R. Shalizi, and M. E. J. Newman, “Power-law distributions in empirical data,” *SIAM Review*, vol. 51, no. 4, pp. 661–703, 2009.

[34] A. E. Turgut, İ. C. Boz, İ. E. Okay, E. Ferrante, and C. Huepe, “Interaction network effects on position- and velocity-based models of collective motion,” *Journal of The Royal Society Interface*, vol. 17, no. 169, p. 20200165, 2020.

[35] M. Aldana and C. Huepe, “Phase Transitions in Self-Driven Many-Particle Systems and Related Non-Equilibrium Models: A Network Approach,” *Journal of Statistical Physics*, vol. 112, pp. 135–153, jul 2003.

[36] M. Bando, K. Hasebe, K. Nakanishi, A. Nakayama, A. Shibata, and Y. Sugiyama, “Phenomenological Study of Dynamical Model of Traffic Flow,” *Journal de Physique I*, vol. 5, no. 11, pp. 1389–1399, 1995.

[37] P. Zarchan, H. Musoff, A. I. of Aeronautics, and Astronautics, *Fundamentals of Kalman Filtering: A Practical Approach*. Progress in astronautics and aeronautics, American Institute of Aeronautics and Astronautics, 2015.

[38] A. Barrat, M. Barthelemy, and A. Vespignani, *Dynamical processes on complex networks*. Cambridge university press, 2008.

[39] M. E. Newman, A.-L. Barabási, and D. J. Watts, *The structure and dynamics of networks*. Princeton university press, 2006.

[40] "Crazyflie 2.1." https://www.bitcraze.io/products/crazyflie-2-1/, Last accessed on 2021-11-15.

[41] "Vicon vantage motion capture system." https://www.vicon.com/hardware/cameras/vantage/, Last accessed on 2021-11-15.

[42] J. A. Preiss, W. Hönig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3299–3304, IEEE, 2017. Software available at https://github.com/USC-ACTLab/crazyswarm.