2017-04-20

Distance-Based Behaviors for Low-Complexity Control in Multiagent Robotics

Pietro Pierpaoli
University of Miami, p.pierpaoli@umiami.edu

UNIVERSITY OF MIAMI

DISTANCE-BASED BEHAVIORS FOR LOW-COMPLEXITY CONTROL IN MULTIAGENT ROBOTICS

By

Pietro Pierpaoli

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

May 2017

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

DISTANCE-BASED BEHAVIORS FOR LOW-COMPLEXITY CONTROL IN MULTIAGENT ROBOTICS

Pietro Pierpaoli

Approved:

Ryan L. Karkkainen, Ph.D.
Professor of Mechanical and Aerospace Engineering

Amir Rahmani, Ph.D.
Robotics System Engineer
Jet Propulsion Laboratory

Kamal Premaratne, Ph.D.
Professor of Electrical and Computer Engineering

Manohar Murthi, Ph.D.
Professor of Electrical and Computer Engineering

Guillermo J. Prado, Ph.D.
Dean of the Graduate School

PIERPAOLI, PIETRO (Ph.D., Mechanical and Aerospace Engineering)

Distance-Based Behaviors for Low-Complexity Control in Multiagent Robotics (May 2017)

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Ryan L. Karkkainen.
No. of pages in text. (96)

Several biological examples show that living organisms cooperate to collectively accomplish tasks impossible for single individuals. More importantly, this coordination is often achieved with a very limited set of information. Inspired by these observations, research on autonomous systems has focused on the development of distributed control techniques for control and guidance of groups of autonomous mobile agents, or *robots*.

From an engineering perspective, when coordination and cooperation is sought in large ensembles of robotic vehicles, a reduction in hardware and algorithms' complexity becomes mandatory from the very early stages of the project design. The research for solutions capable of lowering power consumption, cost and increasing reliability are thus worth investigating.

In this work, we studied low-complexity techniques to achieve cohesion and control on swarms of autonomous robots. Starting from an inspiring example with two-agents, we introduced effects of neighbors' relative positions on control of an autonomous agent. The extension of this intuition addressed the control of large ensembles of autonomous vehicles, and was applied in the form of a herding-like technique. To this end, a low-complexity distance-based aggregation protocol was defined. We first showed that our protocol produced a cohesion aggregation among the agent while

avoiding inter-agent collisions. Then, a feedback leader-follower architecture was introduced for the control of the swarm. We also described how proximity measures and probability of collisions with neighbors can also be used as source of information in highly populated environments.

to Laura

Table of Contents

List of Figures

CHAPTER 1

Introduction

The concept of *multiagent mobile robotics* refers to systems composed by ensembles of autonomous vehicles, collaborating to achieve a common goal, while exchanging information. In the past few decades, an extensive amount of research on multiagent systems has been developed and practical applications are now emerging in many fields.

Several examples in nature show the benefit arising from cooperation and organization among individuals. Flocks of birds are capable of self-organizing their flight formation, improving efficiency and traveled distance. Schools of fish exploit their wakes in order to reduce drag and adjust distances from neighbors in order to deceive predators. Wolves distribute roles among the pack, maximizing foraging success. The fact that these behaviors are often results of simple interactions with limited information is of great interest to the multi-agent research community. This ultimately highlights the fundamental importance of the communication structure topology when compared to the information itself. This great coordination in biological systems and their ability to perform group tasks that are beyond the capabilities of a single agent is the motivation for the multi-agent robotics.

A particular class of multiagent systems, namely *swarm* or *swarm robotics*, has largely diffused in many contexts related with engineering and computer science. The term *swarm* is used in biology to indicate an ensemble of similar individuals specialized in accomplishing a common task, relying on some degree of collaboration. Although relatively inexpensive sensors and actuators allow the design of advanced robotic systems at moderate costs, there is still an interest in investigating low-complexity solutions. In particular, on large production scales, even small reductions in the complexity results in reduced maintenance and manufacturing costs, lower power consumption, and higher reliability. In many biological species, for instance, communication protocols are relatively simple processes and yet very effective; e.g. chemical communication (release of pheromones), group signal (waggling and dance), or environment modification.

In this work we embraced the potential benefit of multiagent robotics and we moved a step forward in the analysis and definition of low-complexity control techniques. In accordance with many other works, we focused our study on control algorithms where relative distance and/or position measurements represents the main source of information for the agents. The popularity of this choice is due to the fact that these measurements generally come at a low cost and limited complexity. Here we consider a somewhat relaxed version of a formation tracking problem which we refer to as *swarming* or *aggregation behavior*. As it will be shown in the next chapter, formations are generally expressed as a set of feasible distances to be maintained among agents. Conversely, our interest is in the definition of homogeneous distributed behaviors where inter-agent distances are maintained within the neighborhood of a reference distance. To this end, our swarming process could be thought as one of the

possible behaviors a multiagent system can switch to, when particular stages of their mission do not require an exact formation's shape (i.e. moving between successive points of interest, reduction of the swarm's size, narrow passages). The problem of controlling the group responding to our aggregation behavior was also investigated. To this end, we connect to the important problem of the leader selection.

The proposed algorithm presents many advantages over other available solutions. In particular, our agents are not required to distinguish their neighbors while trying to reach the stability of the formation. Moreover, as a completely decentralized algorithm, the complexity of the swarming process proposed does not rise with the number of individual in the team.

The remaining of this work is structured as follow. In the next chapter we describe some of the major techniques that have been proposed for the guidance and control of multiagent systems. Most relevant features connected with our problem are described together with their shortcomings. In Chapter 3 we introduce a motivating example for our main contribution. The effect of a distance-based behaviors will be shown by means of a pair of UAVs whose dynamic is coupled by the presence of a collision avoidance algorithm. A fundamental correlation between autonomy and neighbors' distance was introduced, showing trustworthiness and safety issues. In Chapter 4 our swarm aggregation process was described using elements from rigidity theory and Lyapunov stability. The effect of distance-based reaction introduced in Chapter 3 were extended in Chapter 5 to the control of large swarms. Lastly, in Chapter 6, the use of binary measurements of proximity with neighbors were used to define a filtering process useful for the agent self-localization in crowded environments.

1.1 Preliminaries and Nomenclature

We introduce here main notation and nomenclature assumptions used throughout this work. Additional details and theories will be introduced in each chapter as required. We will generally refer to the atomic element composing a swarm either as *agent*, *node*, vehicle or simply *robot*. We denote vectors and matrices by lower case and capitol letters respectively. We define $\mathbb{R}^{a\times b}$ the set of real valued matrices with a-rows and b-columns. Also, diag(a), with $a \in \mathbb{R}^n$, is the matrix of size $n \times n$ having a as diagonal entries; $\|a\|$ is the Euclidean 2-norm of a. We denote $\mathbf{1}$ the vector of appropriate size having all entries equal to 1 and I_n as the identity matrix of size $n \times n$. $A \otimes B$ indicates the Kronecker product between matrix A and B; $A^{\circ\frac{1}{2}}$ is the Hadamard root of matrix A.

For an ensemble of n autonomous vehicles, we let $x_i \in \mathbb{R}^2$, for $i \in \{1, \ldots, n\}$, be the state of agent i^{th} in a two dimensional domain. Where ground vehicles are considered, a two dimensional domain fully describes the problem. However, under particular scenarios, aerial applications might fit within this assumption as well; this is, for example, the case of single/multi-rotor flying vehicles whose downstream aerodynamic wake prevents them from being vertically overlapped. Similarly, given layered structure of the airspace, aircraft are not allowed to freely change their altitude.

The composite state vector for the whole system is defined as $x \in \mathbb{R}^{2n}$. We assume the single agent's dynamic to be modeled by a differential equation of the form:

$$\dot{x}_i = f_i(x_i, u_i) \qquad x_i \in \mathbb{R}^2 \qquad u_i \in \mathbb{R}^c \qquad i = \{1, \ldots, n\}. \tag{1.1}$$

where c is the dimension of the control space. We let $x = \{x_1, \ldots, x_n\}^T \in \mathbb{R}^{2n}$ and $u = \{u_1, \ldots, u_n\}^T \in \mathbb{R}^{2c}$ represents the composite state and input vectors of the ensemble of robots.

Similarly, we can express equation (1.1) on a discrete time domain indexed by $k = 1, 2, \ldots$ using the difference equation:

$$x_i[k+1] = x_i[k] + g_i(x_i[k], u_i[k]) \tag{1.2}$$

Many factors could lead to the choice of one time domain over the other. For example, if vehicles are allowed to contentiously communicate or the bandwidth is sufficiently large, a continuous model is used. On the other side, a model similar to (1.2) could represents scenarios where sharing of information only takes place at discrete instances of time. In the following chapters, it will be made clear when a discrete time model is considered. If not stated otherwise, a continuous time domain is assumed.

CHAPTER 2

Control of Multiagent Mobile Systems

In this chapter we provide a brief introduction on multiagent mobile systems, discussing some of the most relevant results in the field. Given the great amount of research available on the topic, this is not meant to be an exhaustive review on the topic but rather to highlight some of the most relevant solutions and relative shortcomings. More complete survey on the topic have been developed by Parker [1], Murray [2], Brambilla et al. [3] and Cao et al. [4].

2.1 Group Behavior Studies in Biology and Physics

Early works on group behaviors have been developed by biologist and physicists. Primary focus was the definition of proper models capable to described frequently observed behaviors in animals such as aggregation, team foraging and herding. Mogilner and Edelstein-Keshet [5] proposed a integro-differential advection-diffusion equations model representative of the attraction repulsion forces taking place between organisms. The proposed models describes the evolution of the population density using different model of diffusion and stability results are discussed. Agent belonging to swarms have been described in physics literature as *self-driven particles*. Vicsek et al. [6] describe a model composed by particles traveling with constant absolute ve-

locity equal to the average of their neighborhood subject to the effect of a uniformly distributed random perturbation. They show that when high noise are present no transportation effect takes place in the swarm and as it is reduced transition to coherent motion is achieved. Similar results are presented in [7] where self-propelled particles respond to long-range interactions. In this case, the system presents either coherent traveling states and incoherent oscillatory state depending on noise intensity. Shlizerman et al. [8] present a model for directional flight used by migrating monarch butterflies which is based on a time-compensated sun-compass. As described by Pratt [9], colonies of ants can choose nesting site and propagate the decision using a quorum process simply based on frequency of encounters with nest mates.

2.2 Multiagent System in Engineering

Multiagent systems have gathered much attention in the lest decades thanks to broad field investigation conducted not exclusively by community affiliated with engineering and computer science but also in economic and social studies. The arising of a large variety of possible applications for multiagent systems has also favorably contributed to the growing interest in the topic. Scientific explorations and rescue activities in areas where human life or safety are not guaranteed could be enhanced by a coordinated team of aerial, ground, marine vehicles or possibly a fusion of them. Similarly, environmental monitoring, area coverage (i.e. border control, meteorology), mapping, target tracking and sensor networks represents some of the many application where multiagent robotics might generate a positive impact. Appealing features includes extended autonomy, absence of traditional pilots training and accessibility to remote/dangerous environments. Group of small autonomous agents can also re-

place the massive traditional machine used for field cultivation or to realize collective transportation or manipulation of objects.

Control and estimation, which are the two fundamental problem in automatic systems, are tightly connected in multiagent based platforms, where the limited capability of sensors require an actual motion of the robot in the environment. For instance, let us consider a team of UAV employed for a map exploration task. The benefit arising from specific maneuvers and trajectories is directly dependent on the degree of knowledge of the environment (state of the swarm and external conditions). Conversely, given the limited range of real sensors, the degree of knowledge of the surrounding environment depends on the number of locations visited and the exploration paths. Nevertheless, in many application separation principle can be assumed when studying control/planning and estimation/perceptions in distributed system.

The problem of multiagent robotics guidance address the definition of algorithms capable to realize the routing of agents towards the desired targets, while respecting possible constraints in the environment. A general autonomous motion problem can be partitioned in macro tasks, including but not limited to:

- reach destination or way-points;
- avoid collision with static or moving obstacles as well as other with other vehicles implementing a collision avoidance algorithms;
- platooning and formation control;
- task allocation and scheduling;

The particular mixture of these tasks is peculiar of the problem under consideration. Often times, due to the complexity of the original problem or given the interest in

a particular area of research, a separation principle is employed in order to address these aspects separately. However is important to keep in mind possible coupling effects in order to produce effective solutions for the problem.

Diversification and classification of multiagent systems, considered as collections of autonomous and mobile vehicles, have been performed following several different approaches such as, size of the group, level of heterogeneity, degree of decentralization. Analysis and models based on both Eulerian and Lagrangian approaches have been used. Moreover, a group of agents could also be modeled using continuum, discrete or hybrid formulations and both continuous and discrete time domains models can be found in literature. Finally, homogeneity or heterogeneity can be used to classify configurations respectively formed by ensembles of vehicles being all equal to each other or having different skills and features. An other option is to separate *centralized* and *decentralized* (or *distributed*) techniques.

2.3 Centralized Solutions for Control of Multiagent Systems

In centralized control, a supervisor or *leader* collects information on the state of the agents and the surrounding environment; once these inputs have been processed, appropriate instructions are sent back to each agent. Given the necessary flow of information between nodes and control unit, these techniques are best suited where a limited number of agents is involved. In fact, the communication burden arising from the continuous share of information prevents this approach from being applied to very large scale applications. Moreover, the continuous communication and the presence of

single node of failure limit centralized approaches to few field of application. However, given the usually realistic assumption of full state/objective knowledge, properties such as efficiency and stability are easier to achieve and prove.

Formation control is one of the most investigated behavior in multiagent robotics. Aircraft and drones can reduce their radar print when flying in tight formations. Particular disposition of ground vehicles can increase the probability of detection of enemies activity or prevent access to restricted areas. The problem is the definition of required controls to reach and maintain relative positions among agents. Formation control techniques have been developed in both centralized and decentralized frameworks.

Trajectory planning for multiagent systems can be formulated as multiple traveler sales man problem. Algorithm to define travel paths the minimize total agents traveled distance can be formulated. These algorithms can include constraints on the trajectories, such as collision avoidance, as well as particular agents/location constraints.

A large body of research has been directed toward algorithms for air traffic control. The current air traffic control structure is an example of a centralized control, where a supervising unit issues dedicated routing instructions to each aircraft. Air traffic control is a particularity challenging problem due to the high safety standard to be maintained and its high operative costs.

2.4 Distributed Solutions for Control of Multi-Agent Systems

The term *distributed* (or *decentralized*) control refers to techniques based on the assumption that agents accomplish a desired task in a somehow collaborative fashion, without any kind of external coordinator or leader. This kind of solutions have gathered much attentions in last decades thanks to appealing properties they posses when compared with centralized counterparts. The original idea of distributed systems take its inspiration from an abundant number of biological examples observable in nature. Many species are in fact capable of achieving surprisingly complicate results only relying on a set of local information and limited skills. The synergy and cooperation among agents make the swarm capable of achieving tasks impossible to the single. Moreover, as frequently observed in nature, individuals perform much better when operate collectively compared to solo performances.

A *swarm* of robots is defined as a group of autonomous agents, that operates in cooperation relying on limited internal complexity, skills and communication. Limited cognitive capabilities prevent nodes from perceive the swarm behavior and its large scale outcome; nonetheless, inputs and decisions are defined at single agents level. Give the large number of vehicles and the lack of a single point of control, swarm of robots have higher resilience to loss and failure when compared to centralized architectures. In addition to that, their natural inclination to operate in parallel, leads to higher efficiency and flexibility to adapt to changes in the environment or in their tasks.

Due to the lack of a global swarm self-observability on its state and performance, the definition of stable protocols and efficient trajectories is not trivial. This is especially true when we consider a state dependent time-varying interaction structure between the individuals. Moreover, when considering systems composed by an high number of individuals, practical limitations must be accounted for when investigating physical realizations. For this reason, control algorithms must be scalable, in a way that complexity is as much independent as possible by the number of members in the swarm. In these scenarios, even little savings in the realization, maintenance and operation costs could result in overall great savings. For example, wireless communications or absolute localization systems (i.e. GPS) might be considered too expensive and power inefficient for some applications; therefore, there is a great interest in alternative solutions. On the other side, distance measurements performed using *sonar, lidar* or *IR* sensors can be significantly cheaper, although their limited range and cannot distinguish objects when they are very close.

Camera and video devices are commonly used in robotic localization and estimation. In SLAM (Simultaneous Localization and Mapping) techniques, image processing and inertial navigation are coupled in order to provide location estimation and mapping of the environments at once. SLAM techniques are very popular and increasingly diffused; however, the intensive computational resources and large power storage they need, makes them unsuitable form some applications.

Relative low power-consumption sensors include optical and tactile sensors that can be used to read landmarks or detect barriers and obstacles. Finally, actuators and motors represents in many circumstances more than half of the power consumption.

For this reason, great interest is in developing control algorithms that require simple movements at optimized regimes and velocities.

Following these consideration, great interest is devoted to *low-complexity* solutions, where agent can achieve complete tasks relying on reduced sensing capabilities while operating in a distributed fashion. In the remaining of this section we review some of the most relevant solution proposed for distributed control for multiagent systems, highlighting relative advantages and shortenings.

2.4.1 Optimal Control Techniques

In Dunbar and Murray [10] a cost function representative of the *cumulative formation error* and the control efforts is defined as:

$$L(x,u) = \sum i = 1,\ldots,n \left(f(x_i, x_{-i}) + \|u_i\|^2 \right). \tag{2.1}$$

Distributed optimal solution is found using a receding horizon approach. The model requires each agent to communicate its optimal control trajectory. Stability can be achieved if each control do not deviate significantly from the transmitted one and the receding horizon updates is performed sufficiently fast. This technique has been used to perform a distributed point tracking while maintaining formation.

2.4.2 Probabilistic Guidance Algorithm

In probabilistic guidance algorithms (PGA), agents follow a random walk over a partitioned domain in accordance with specific transition probabilities. Transitions occur in accordance with a Markov chain whose limiting distribution reflects the desired configuration for the swarm.

The physical domain $\mathcal{R}$ is defined as a disjoint union of m bins R_i, $i = 1, 2, \ldots, m$ such that:$\mathcal{R} = \bigcup_{i=1}^{m} R_i$, $R_i \cap R_j = \emptyset$. Let N be the number of agents. Each agent has position $r(t)$ at time t. The probability that one agent will be at time t in cell R_i is defined through the probability vector $x_i(t)$:

$$x_i(t) := \mathcal{P}(r(t) \in R_i) \qquad \text{and} \qquad x \in \mathbb{R}^m \tag{2.2}$$

therefore $x(t)$ is the swarm distribution. Since each agent acts independently, equation (2.2) is true for N agents. The ensemble of the all agents position is $r_k(t)_{k=1}^{N}$, which by the law of large numbers has a distribution that approaches $x(t)$ as the number of agent is increased. Purpose of the guidance algorithm is to guide the agents toward a specific distribution described by a probability vector π. If we have m bins or vertex, $\pi[i]$ is the desired probability of finding an agent in cell R_i:

$$\lim_{t\to\infty} x_i(t) = \pi_i \qquad \text{for} \qquad i = 1, \ldots, m \tag{2.3}$$

$N\pi(i)$ is the expected number of agents to be found in ν_i. As $N \to \infty$, $\frac{n}{N} \to \pi$ (π may be seen as desired average fraction of agents in ν_i). The entries of Markov matrix M, where $M \in R^{mxm}$, are defined as transitional probabilities, in particular:

$$M_{i,j} = \mathcal{P}(r(t+1) \in \nu_i | r(t) \in \nu_j) \tag{2.4}$$

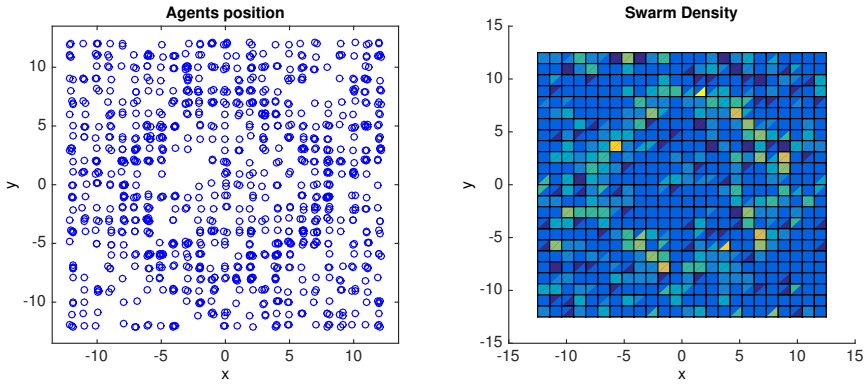
The evolution of the probability vector x is defined by the Markov chain:

$$x(t+1) = Mx(t) \qquad t = 1, 2, \ldots \tag{2.5}$$

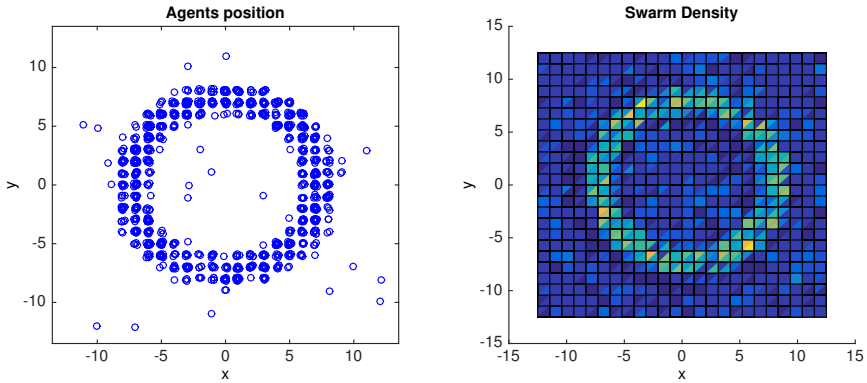
Each agent is provided with a copy of the matrix M and independently propagate its positions as independent process.

In Akmee and Bayard [11] appropriate Markov chain having the desired configuration as limiting distribution are constructed using the Metropolis-Hastings (M-H)

algorithm and a Linear Matrix Inequality (LMI) approach. In both cases rate to convergence to the desired distribution can be tuned as desired. The resulting behavior ensure self-healing properties when a subset of the swarm is removed and can treat domains with keep-out regions. No estimation of swarm distribution is performed and transitions are contously performed. In Bandyopadhyay et al. [12] an inhomogeneous Markov chain is designed such that the number of transitions required by the agent is minimized. Communication among agents are used to estimate swarm convergence to desired formation and maintain positions once reached. Example of the result from application of PGA is represented in Figure 2.1. Although appealing for its highly



(a) Time $t = t_0$



(b) Time $t = t_f$

Figure 2.1: Simulation results from PGA with inhomogeneous Markov chain

distributed realization, PGA requires each agent to have exact knowledge of its loca-

tion. As discussed previously, in some circumstances absolute localization cannot be performed and generally corresponds to high power consumption and costs.

2.4.3 Consensus

Consensus related problems have been extensively studied in recent years [13], [14] [15], [16]. The most general consensus algorithm is represented by the following dynamics:

$$\dot{x}_i = -\sum_{j=1}^{n} w_{ij}(t)\,(x_i(t) - x_j(t)), \quad i = 1, \ldots, n \tag{2.6}$$

where in the most general case w_{ij} is the (i, j) entry of the adjacent matrix associated with the given network topology. The same dynamics can be represented in matrix form as:

$$\dot{x} = -L(t)\,x(t) \tag{2.7}$$

where $L(t) \in \mathbb{R}^{n\times n}$ is the Laplacian of the graph. Consensus is achieved among agents if for all $x_i(0)$, $|x_i(t) - x_j(t)| \to 0$ as $t \to \infty$ for all $i, j = 1, \ldots, n$ (Figure 2.2).

Rendezvous Problem In the rendezvous problem a group of vehicles meet at time or location determined through negotiation. In [17] an algorithm for controlling multiple UAVs is studied. Each aircraft has to reach the boundary of a radar detection area at the same time in order maximize the surprise effect.

Formation Stabilization In formation stabilization vehicles must maintain a predefined relative distance one from each other. In distributed realization, each member must negotiate its position with its peers.

Formation Maneuvering and Flocking Consensus algorithm can be used to obtain a moving formation that adapt to changes in the environment or to obstacles. These techniques have been largely applied to computer animations which have inspired several robotics developments [15].

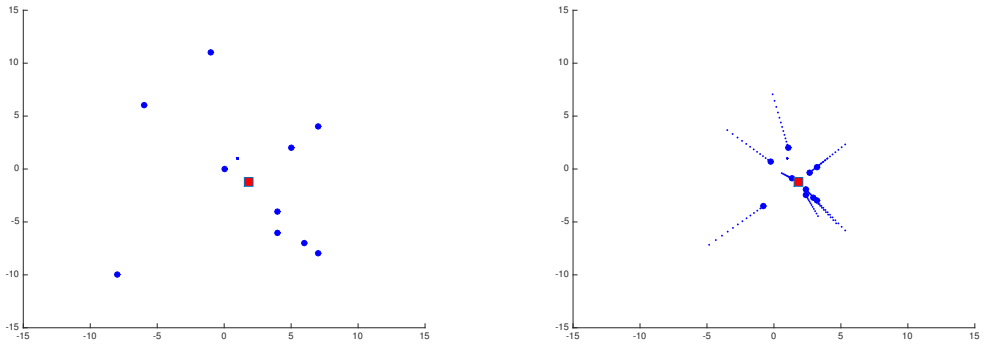


Figure 2.2: Consensus on multiagent system centroid.

2.4.4 Artificial Potential Function

Artificial potential fields have been extensively used in distributed control of multiagent systems, some more relevant examples can be found in [18], [19], [20]. In artificial potential based control, a function is chosen in such a way that its gradient corresponds to a suitable control input for the robot. Artificial potential functions have been originally introduced by Khatib [21] for control of object avoiding robotics manipulators. The idea has been further developed by Rimon and Koditschek in [22] where a class of admissible potential function is introduced. The paper also introduces the notion of *navigation function* defined as a class of potential function with particular properties.

Algorithms based on collective potentials where introduced by Olfati-Saber in [15] for distributed flocking. In [23] a combination of navigation function and swirling

function is designed to realize a stable deconfliction among UAVs and route them towards their goal. The techniques consider constraints in the vehicles turning rate and cooperativeness among agents, however linear velocities are assumed to be constant.

2.4.5 Game Theory

In recent year, game theory or decision theory has been applied to multiagent systems. Game theory is a mathematical techniques that can be used to describe and characterize a decision process when outcome of single actions are not know. In multi-agent systems, these theory become interesting due to the uncertainty on the external environment. In [24] each robot is described by traits of personality which cumulatively describe the robot behavior. A learning process based on game theory is introduced and complex behaviors can be represented and solved as zero-sum games.

CHAPTER 3

Non-cooperative Trajectory Modification

3.1 Overview

In this chapter we present a motivating case study for the problem that will be addressed in the following chapters. The example, investigates the extent to which a collision avoidance algorithm can represent a source of vulnerability for an autonomous vehicle. The interest in this problem is motivated by the fact that when distance based behaviors directly affect the dynamics of system, such as in a collision avoidance framework, the trajectory of one vehicle can be modified by particular trajectories of its neighbors.

In this chapter we consider a UAV employing a collision avoidance algorithm, whose objective is to reach its destination while avoiding collisions with other vehicles. On the other side, a team of UAV, without employing any collision avoidance algorithms, tries to steer the trajectory of the first UAV from its original direction towards a new target.

3.1.1 UAV Collision Avoidance Exploitation for Noncooperative Trajectory Modification

Due to the increasing degree of automatization in air traffic managment, *cyber-physical* security has become a major topic in ATM research. Along with that, the combination of fully or partial automated aerial systems give rise to a complicate hybrid structure whose verification must be deeply investigated before a large scale implementation can be accomplished. A thorough security analysis must include both unintentional malfunctions and adversarial attacks. For examples, given the same working frequency band, possible interference between radar and ADS-B communication can also take place. Towards this end, a performance optimization technique for ADS-B interference is proposed by Park and Tomlin [25].

Critical security vulnerabilities subject to malicious attacks can be targeted to both ground stations and aircraft. Three general classes have been identified by McCallie et al. [26]: interception of transmission, jamming and injection of messages into a data link. In addition to these, several other inherent GPS vulnerabilities have been investigated over the years [27]. For example, spoofing GPS through the transmission of counterfeit signals can induce a false drift in UAV's localization which, in turn, can be used to take control over the vehicle [28]. Jamming of wireless network is considered a *denial-of-service* attack caused by the presence of a high power transmitting device which prevent other lower power device connectivity. In ATM this could be considered a vulnerability mostly affecting ground control station and, in turn, threatening take off and landing operations [29].

Several mitigation and defense techniques have been proposed against the aforementioned vulnerabilities. For example, cross-verification of ADS-B data through

sensor fusion techniques seems to improve trustworthiness of broadcast messages and GPS data disruptions [30] [31] [32].

In autonomous mobile robotics a major class of threats is represented by losses of minimum separation with surrounding objects (collisions). Conflict detection and resolution (CDR) techniques have been largely investigated in the last few decades. Obstacle avoidance in dynamic environment still represents a big challenge, especially when efficient trajectories and online real time solutions are sought.

Main sources of penalty when designing CDR techniques are 1) chances of collisions and 2) deviation from the original path. Nevertheless, despite the enormous progresses accomplished in the enhancements of these techniques, there is further class of vulnerability, to which an agent is exposed when avoiding a collision. In fact, as introduced in [33], the presence of a collision avoidance layer in the agents' control algorithm, introduces a fundamental dependency from the external environment (Figure. 3.1). Upon detection of collisions, original navigation plans must be revised and trajectory are modified in accordance with deconfliction rules or shared policies. In this sense, the event of collision represents an additional constraint on the vehicle's dynamics. In this chapter we show that when deterministic collision resolution policies are engaged, avoiding maneuvers can be anticipated. Consequentially, the dependency from other agents or external environments can be exploited; this results in partial or total loss of control on the autonomous agent.

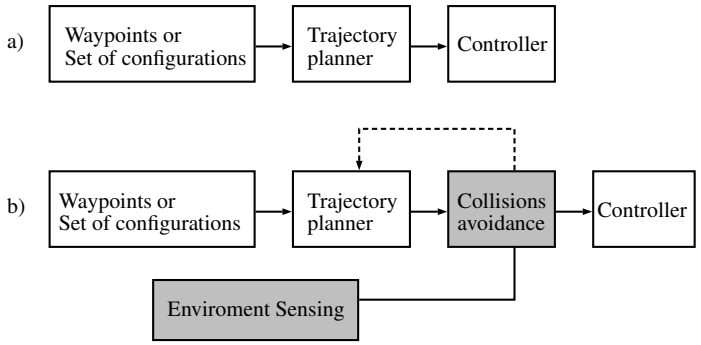


Figure 3.1: Autonomous mobile agent layered structure for motion planning without (a) and with (b) sense and avoid layer

The resulting threat is emphasized when knowledge of exact location of aircraft is made available or can be precisely measured. For example, *automatic dependent surveillance broadcast* (ADS-B) provides a framework for sharing of navigation information among aircraft. Main purpose of ADS-B is providing each aircraft with precise position and velocity of neighbors. The introduction of this system is the first effort toward an increase of automation in air traffic control (ATC) required to accommodate larger volume of flights and ease the integration between unmanned and traditional aircraft. Given the typical distance between aircraft, satellite based information yield accuracy impossible to obtain otherwise. In circumstances where agents are allowed to travel at lower distances, range sensors could provide similar accuracy.

We refer to *evader* as an autonomous vehicle employing a collision avoidance algorithm whose goal is to sequentially reach a set of fixed destinations or way-points. On the other side, a *pursuer*, without adopting any kind of collision avoidance reasoning, aims to steer the evader towards a predefined set of target poses. We will refer to this framework as *unilater collision avoidance*. Purpose of this work is to twofold:

first, we investigate the constrained dynamics of a vehicle during a collision avoidance maneuver and then, we define a strategy a pursuer agent can perform in order to steer the evader towards a predefined target or *capture* set. A full understanding of the problem is the first objective of this and following works, having as ultimate goal the design of avoiding strategies capable of mitigating this and other vulnerabilities.

3.2 Velocity Obstacle Exploitation

3.2.1 Model definition

Planar motions within a two dimensions space are considered. Where a layered airspace structure does not allow for significant aircraft changes in altitude this assumption reflects a real constraint. In addition to this, aerodynamic wake produced by rotor-craft such as helicopter or quadcopter prevent these vehicles from being safely overlapped. Here, this simplification is introduced to limit the complexity and represents a first step towards a complete understanding of the problem.

A global Cartesian reference system attached to the earth is considered. Vehicles configurations are denoted as $(\mathbf{q}, \phi)^T$, where $\mathbf{q} = (x, y)^T \in \mathbb{R}^2$ are coordinates in the global system and $\phi \in [0, 2\pi) = \mathbb{S}$ is the agent's angular orientation (or heading) with respect to it [1]. Given the planar assumption and considering small variation in the velocity vector, aircraft dynamics can reasonably be approximated using the unicycle

[1] Note that, for the extension of this chapter only, bold face style is used for vector quantities.

model ([34]):

$$\begin{aligned} \dot{x} &= v\, cos\phi \\ \dot{y} &= v\, sin\phi \\ \dot{\phi} &= \omega. \end{aligned} \tag{3.1}$$

Communications delays or interruptions are neglected and the information flow between agents takes place in real time. The information set at each time for each agent is represented by the current neighbors configuration. Such a complete information pattern can be obtained in different ways: appropriate on-board sensors, a network based sharing protocol (i.e. ADS-B), an overhead tracking system or, most likely, a fusion of them.

We also assume non-cooperativeness among agents. That is, final targets and routes do not belong to the set of shared information. Without a secure and consolidated *trustworthiness criteria*, neighbors cannot be arbitrarily trusted nor navigation plans allowed to be freely shared. It is worth noting that the existence of a information sharing protocol among neighbors is not in contrast with the hypothesis of non-cooperativeness. Finally, we assume avoiding maneuvers to be performed as heading changes maneuver only. This assumption is introduced here in order to simplify the analysis of the problem. However, for fuel economy, a purely turning maneuver could in fact be the preferred choice when compared to changes in speed and altitude.

We define $\mathcal{T} \subset \mathbb{R}^2 \times \mathbb{S}$ to be the set of target poses toward which the purser aims to steer the evader and $\mathbf{q}_G \in \mathbb{R}^2$ to be the evader's fixed goal (i.e. final or part of a set of checkpoint).

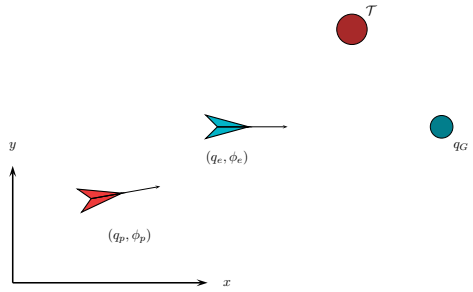


Figure 3.2: The problem studied here is involves two agents; an *evader* agent (blue in figure) whose objective is to reach a point $\mathbf{q}_g$ and a *pursuer* agent (red in figure) that tries to steer the evader into a capture set $\mathcal{T}$.

3.2.2 Revised Collision Cone Method

In cooperative navigation frameworks the communication of intents between agents and the agreement on common rules allows the sharing of the avoiding efforts between neighbors and long horizon optimization of trajectories. On the other side, in noncooperative scenarios, decisions are based on frequently updated measurements; in this case, online reactive CDR algorithms are best suited. Reactive CDR algorithms rely on a proper projection of neighbors configurations. Three main different projections have been proposed in litterature: linear, worst case and probabilistic projection [35]. In this work we build our reactive CDR algorithm based on a revised concept of *Velocity Obstacle* (VO). The idea behind VO method, first introduced by Fiorini and Shiller [36] is to define a set of velocity vectors for which, given the current neighbors poses and assuming no variation in their velocities, a collision will occur for some time $t < \infty$. The VO method is closely related to the Collision Cone method introduced by Chakravarthy and Ghose [37], where collision configurations are mapped into agents absolute velocity space and a set (cone region) of collision headings is then defined. Both methods assumes linear trajectory projections, which means that

obstacles are assumed to move at constant velocities between consecutive iterations of the algorithm.

Assuming identical and circular protected zone with radius R_{pz}, and denoting $\|\cdot\|$ the euclidean norm operator, we introduce the following definition:

Definition 1 *A collision is defined as the event of two agents separation being less then the protected zone radius, therefore, agents E and P experience a collision if and only if* $\|\mathbf{q}_e - \mathbf{q}_p\| < R_{pz}$

It is worth to note that the definition above does not differ from collisions considered as intersection between two protected zones. In fact, the protected zone radius can be assumed to be the sum of both agents radii and hence, without loss of generality, the collision problem can again be reduced to a point-circle collision problem, where the circle has in this case radius $2R_{pz}$.

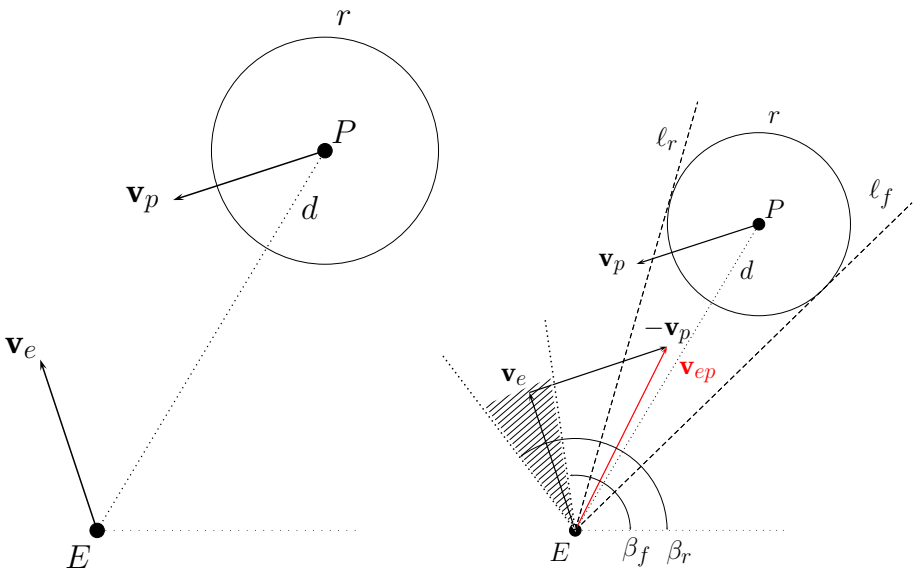


Figure 3.3: Collision avoidance technique geometry and nomenclature.

With reference to Fig.3.3(a), let us consider the collision between a point E and a circle with radius R_{pz} centered in P separated by distance $\mathbf{d} = \|\mathbf{q}_p - \mathbf{q}_e\|$. The relative velocity (red in Fig. 3.3(b)) is defined as:

$$\mathbf{v}_{ep} = \mathbf{v}_e - \mathbf{v}_p = v_{ep} \begin{bmatrix} cos\phi_{ep} \\ sin\phi_{ep} \end{bmatrix} = \begin{bmatrix} v_e \, cos \, \phi_e - v_p \, \cos \, \phi_p \\ v_e \, \sin \, \phi_e - v_p \, \sin \, \phi_p \end{bmatrix} \tag{3.2}$$

and the two tangents from E to P's protected zone, are λ_f and λ_r for which the following expression hold:

$$\angle \ell_{f,r} = \lambda_{f,r} = \arctan\left(\frac{y_p - y_e}{x_p - x_e}\right) \pm \arcsin\left(\frac{R_{pz}}{\|\mathbf{q}_e - \mathbf{q}_p\|}\right) \tag{3.3}$$

Theorem 1 *At time t_0, consider two agents E and P, having fixed velocity and heading. A collision between E and P occurs at time $t_0 < t < \infty$ if and only if the relative velocity vector belongs to planar sector (or collision cone) formed by the two tangents to the protected area of agent P having apex in E, that is if:*

$$\lambda_f < \arctan \frac{\sin \phi_{ep}}{\cos \phi_{ep}} < \lambda_r \tag{3.4}$$

For a complete proof of this theorem we refer the reader to [38]. We provide a revised version of the same proof applied to the following result.

Lemma 1 *When the relative velocity vector is oriented as λ_f, that is:*

$$\phi_{ep} = \arctan\left(\frac{y_p - y_e}{x_p - x_e}\right) + \arcsin\left(\frac{R_{pz}}{\|\mathbf{q}_e - \mathbf{q}_p\|}\right)$$

then:

$$min\|\mathbf{d}(t)\| = R_{pz}.$$

Proof: Given the definition of the relative velocity as defined in (3.2), the distance between E and P can be expressed as:

$$\mathbf{d}(t) = \mathbf{d}_0 - \mathbf{v}_{ep}\, t$$

Minimizing the square of the following quadratic form and setting the result equal to 0:

$$\frac{d}{dt}\left(\frac{1}{2}\|\mathbf{d}(t)\|^2\right) = \|\mathbf{d}(t)\| = \mathbf{d}_0 - \mathbf{v}_{ep}\, t = 0$$

and pre-multiplying by $\mathbf{v}_{ep}^T$:

$$\begin{aligned} 0 &= \mathbf{v}_{ep}^T\, \mathbf{d}_0 - \mathbf{v}_{ep}^T\, \mathbf{v}_{ep}\, t \\ t_m &= \frac{\mathbf{v}_{ep}^T\, \mathbf{d}_0}{\|\mathbf{v}_{ep}\|^2} \end{aligned} \tag{3.5}$$

we obtain the expression of the time t_m at which the two vehicles experience the minimum separation. With reference to Fig.3.3(b), if we assume $\phi_{ep} = \lambda_f$ and using (3.3):

$$\begin{aligned} \mathbf{v}_{ep}^T\, \mathbf{d}_0 &= \|\mathbf{v}_{ep}\|\, \|\mathbf{d}_0\| \cos(\angle \mathbf{v}_{ep} - \angle \mathbf{d}_0) \\ &= \|\mathbf{v}_{ep}\|\, \|\mathbf{d}_0\| \cos\left(\phi_{ep} - \arctan\frac{y_p - y_e}{x_p - x_e}\right) \\ &= \|\mathbf{v}_{ep}\|\, \|\mathbf{d}_0\| \cos\left(\arcsin\frac{R_{pz}}{\|\mathbf{d}_0\|}\right). \end{aligned} \tag{3.6}$$

We can then write the time to the minimum distance defined in (3.5), when $\phi_{ep} = \lambda_f$ as:

$$\begin{aligned} t_m &= \frac{\|\mathbf{v}_{ep}\|\, \|\mathbf{d}_0\|}{\|\mathbf{v}_{ep}\|^2} \cos\left(\arcsin\frac{R_{pz}}{\|\mathbf{d}_0\|}\right) \\ &= \frac{\|\mathbf{d}_0\|}{\|\mathbf{v}_{ep}\|} \cos\left(\arcsin\frac{R_{pz}}{\|\mathbf{d}_0\|}\right) \end{aligned} \tag{3.7}$$

The norm of the distance can be expressed as:

$$\begin{aligned} \|\mathbf{d}(t)\| &= \sqrt{(\mathbf{d}_0 - \mathbf{v}_{ep}\, t)^T (\mathbf{d}_0 - \mathbf{v}_{ep}\, t)} \\ &= \sqrt{\|\mathbf{d}_0\|^2 - 2\, \mathbf{v}_{ep}^T\, \mathbf{d}_0\, t + \mathbf{v}_{ep}^T\, \mathbf{v}_{ep} t^2} \end{aligned}$$

At the time of minimum separation and assuming again $\phi_{ep} = \lambda_f$, we can finally rearrange the last result substituting (3.5) for $\mathbf{v}_{ep}^T \mathbf{d}_0$ and then (3.7) for t:

$$\begin{aligned}
\|\mathbf{d}_m(t)\| &= \sqrt{\|\mathbf{d}_0\|^2 - 2\|\mathbf{v}_{ep}\|^2 t_m^2 + \|\mathbf{v}_{ep}\|^2 t_m^2} \\
&= \sqrt{\|\mathbf{d}_0\|^2 - \|\mathbf{v}_{ep}\|^2 t_m^2} \\
&= \sqrt{\|\mathbf{d}_0\|^2 - \frac{\|\mathbf{v}_{ep}\|^2 \|\mathbf{d}_0\|}{\|\mathbf{v}_{ep}\|} \cos^2\left(\arcsin\frac{R_{pz}}{\|\mathbf{d}_0\|}\right)} \\
&= \sqrt{\|\mathbf{d}_0\|^2 \left(1 - \cos^2\left(\arcsin\frac{R_{pz}}{\|\mathbf{d}_0\|}\right)\right)} \\
&= \sqrt{\|\mathbf{d}_0\|^2 \left(\sin^2\left(\arcsin\frac{R_{pz}}{\|\mathbf{d}_0\|}\right)\right)} \\
&= \|\mathbf{d}_0\| \frac{R_{pz}}{\|\mathbf{d}_0\|} = R_{pz}
\end{aligned}$$

■

The proof can easily be extended for the case in which $\phi_{ep} = \lambda_r$. This result shows what should be the direction for the velocity in the relative frame, in order to have the minimum allowed separation between the vehicles. As we will see, this is relevant for the avoiding policy that will be introduced in 3.2.3.

From the definition of relative velocity, if we fix $\mathbf{v}_p$, when ϕ_{ep} is oriented as λ_f (or λ_r), evader's heading will be equal to β_f (or β_f). Hence, when $\phi_{ep} = \lambda_f$, using equation (3.2) we can write:

$$\frac{v_e \sin\beta_f - v_p \sin\phi_p}{v_e \cos\beta_f - v_p \cos\phi_p} = \frac{v_{ep} \sin\lambda_f}{v_{ep} \cos\lambda_f} = \tan\lambda_f. \tag{3.8}$$

and rearranging:

$$v_e(\sin\beta_f - \tan\lambda_f \cos\beta_f) + v_p(\tan\lambda_f \cos\phi_p - \sin\phi_p) = 0 \tag{3.9}$$

Equation (3.9) defines the collisions mapping from the relative velocity space to the agents heading. Now, letting ϕ_{ep} span the whole set of angles between λ_f and

λ_r it is possible to define the complete set of *collision headings* B_{ep} for the agent E. At time t, the set of heading angles $B_{ep}(t) \subset [0, 2\pi)$ which will lead E into a loss of minimum separation with its neighbor P can be defined as:

$$B_{ep}^t = \{\, \beta \in [0, 2\pi) \,:\, \|\mathbf{q}_e(\tau) - \mathbf{q}_p(\tau)\| < R_{pz}\,,\, t < \tau < \infty\}. \tag{3.10}$$

In order to ease the reading, we will drop the time superscript on B_{ep} acknowledging that the set is constantly updated at each time step.

Since the method just introduced is a pairwise method, each neighbor P_i, with $i = 1, \ldots, n$, is considered separately and all n sets B_{ep_i} are overlapped obtaining the final set of unfeasible headings as:

$$B_{ep} = \bigcup_{i=1}^{n} B_{ep_i} = \{\beta_f^1, \ldots, \beta_r^1\} \cup \cdots \cup \{\beta_f^n, \ldots, \beta_r^n\} \tag{3.11}$$

3.2.3 Deconfliction Policy Definition

In addition to a collision detection algorithm, autonomous navigation requires a suitable *deconfliction policy* in order to produce an evading maneuver such that 1) the minimum separation from obstacles is guaranteed, 2) produces the least deviation from original trajectory and 3) bring agents to the desired destination.

In a cooperative environment, predefined engagements or rules could distribute the evading effort among agents, accordingly to some established criteria such as aircraft's available fuel, aircraft maneuverability or current phase of flight. On the other side, under the hypothesis of non-cooperativeness, the agents must assume no avoiding effort from its neighbors. We introduce here the deconfliction policy employed in the chapter. With reference to Fig.3.4, we assume the evader originally flying with heading $\phi_e = \phi_0$. When an incoming conflict is detected, the evader sets its desired heading to a new feasible heading $\phi_e \notin B_{ep}$.

Definition 2 *An evading maneuver is an heading maneuver towards a new heading* $\hat{\beta}$*, such that:*

$$\hat{\beta} = \arg\min_{\beta \in B_{ep}} \delta(\phi_e, \beta) \tag{3.12}$$

where $\delta(\cdot,\cdot) : [0, 2\pi) \times [0, 2\pi) \to \mathbb{R}$ *is a proper metric between angles that takes into account the periodicity of the angular dimension.*

We also assume that if $B_{ep} \neq \emptyset$, then $\phi_e \notin B_{EP}$ at all times (Fig.3.4c). This mean that the agent is never allowed to cross the set of collision headings in order to resolve an incoming conflict. If we consider smaller UAVs, where size and power resources are more limited, it might not be possible to have an accurate estimation of the expected time to impact. For this reason, we do not consider resolution strategies that bring aircraft on a collision route as part of the evading maneuver.

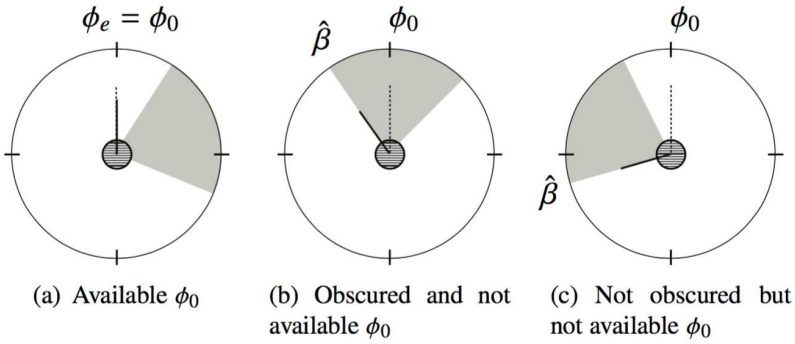


Figure 3.4: Deconfliction policy outcome examples; whenever the desired heading ϕ_0 belongs to B_{ep} or B_{ep} is between current and desired heading, the evader switches its heading towards the closest boundary of B_{ep} aiming for a grazing maneuver.

3.2.4 Complete System

Using the collision avoidance technique defined in the previous section, we can define a suitable model for the evader. The evading policy just described, can be

formally included in the dynamic of the agent by adding a switching behavior to the original model. We index the two possible states of the system using a boolean variable $\nu = [0, 1]$. We will refer to the state corresponding to $\nu = 0$ as *collision free state* and *collision state* when $\nu = 1$. Transitions from one state to the other are dependent from the current neighbors' state and processed online by the collision detection algorithm. Therefore, (3.1) can be rearranged as :

$$\begin{aligned} \dot{x}_e &= v_e \, cos\phi_e \\ \dot{y}_e &= v_e \, sin\phi_e \\ \dot{\phi}_e &= \gamma_p(\phi_0(1-\nu) + \hat{\beta}\nu - \phi_e) \end{aligned} \qquad \nu = \begin{cases} 1 & \text{if } \phi_0 \in B_{EP} \\ 0 & \text{if } \phi_0 \notin B_{EP} \end{cases} \tag{3.13}$$

where we introduced a simple proportional controller for the heading governed by a positive gain γ_P. The set of equation (3.13), reveal the dependency existing between evader trajectory and its neighbors.

Using the same unicycle model introduced in (3.1), pursuer agent dynamics, which is not affected by a collision avoidance method can simply be expressed as:

$$\begin{aligned} \dot{x}_p &= v_p \, cos\phi_p \\ \dot{y}_p &= v_p \, sin\phi_p \\ \dot{\phi}_p &= \omega_p \end{aligned} \tag{3.14}$$

Including the constraint resulting from the collision avoidance technique introduced in (3.9) and (3.3), the complete model of the system with one pursuer can then

be represented as follow:

$$
\begin{aligned}
\dot{x}_e &= v_e\, cos\phi_e \\
\dot{y}_e &= v_e\, sin\phi_e \\
\dot{\phi}_e &= \gamma_p(\phi_0(1-\nu) + \hat{\beta}\nu - \phi_e) \quad \nu = \begin{cases} 1 & \text{if } \phi_0 \in B_{ep} \\ 0 & \text{if } \phi_0 \notin B_{ep} \end{cases} \\
\dot{x}_p &= v_p\, cos\phi_p \\
\dot{y}_p &= v_p\, sin\phi_p \\
\dot{\phi}_p &= \omega_p
\end{aligned}
\tag{3.15}
$$

$$
\begin{aligned}
0 &= v_e \left(\tan\lambda\, \cos\hat{\beta} - \sin\hat{\beta} \right) + v_p \left(\tan\lambda\, \cos\phi_p + \sin\phi_p \right) \\
0 &= \lambda - \arctan\left(\frac{y_p - y_e}{x_p - x_e} \right) \mp \arcsin\left(\frac{R_{pz}}{\|q_p - q_e\|_2} \right).
\end{aligned}
$$

where $\hat{\beta}$ is the closest boundary of the unfeasible heading set B_{ep} as defined in (3.12).

The switched system defined in (3.15) is a differential-algebraic equation (DAE) dynamics representing the dynamics of two vehicles E and P coupled by a unilateral collision avoidance method. The switching behavior of the system is governed by the variable ν and as we can see, when $\nu = 1$ the original dynamics of the evader is constrained by the collision avoidance algorithm.

3.2.5 Connection with Game Theory

Beside the nomenclature used for the agents, the problem presented in this section presents, in fact, game features. In particular, a noticebly similar problem where studied in [39] where a two vehicles game problem is solved in order to define a collision

reachability set. In general, a game ([40]) is defined as an interactive decision-making problem where each players choose the action in order to maximize a certain gain. However, in the contest of this paper, given the assumptions and the deconfliction policy we just introduced, once a collision is detected, the evading action is uniquely defined. A game theoretic approach could serve the study of this problem if more relaxed assumptions for the evading maneuver were considered, such as an evader variable velocity. Further details are presented in the final section on future works.

3.3 Trajectory Modification Design

In the previous section we derived the equations that define the dynamics of a system composed by of a pair of vehicles employing a unilateral collision avoidance policy. In this section we describe how appropriate inputs on the system, namely v_p and ω_p, can be designed in order to steer a component of the complete system, namely $\{\mathbf{q}_e^T, \phi_e\}^T$ from its original trajectory towards a new desired one $\{\mathbf{q}_T^T, \phi_T\}^T \in \mathcal{T}$.

Given the switching behavior of the model described in (3.15) we focus our attention on the two states corresponding to $\nu = [0, 1]$ separately.

As described in section 3.2.3, when $\nu = 1$ the evader switches its original heading from ϕ_0 towards $\hat{\beta}$. Therefore, the evader's heading can be affected by producing an appropriate set B_{ep} of collision directions. In this section we present a technique that can be used to produce desired $\hat{\beta}$ and consequently steer the evader towards the capture set $\mathcal{T}$.

3.3.1 Switching Guard Conditions

The first step is to define for what conditions the system switches from state $\nu = 0$ to $\nu = 1$. In order to so, let's introduce an evader's reference frame centered in $\mathbf{q}_e$ and oriented along its heading ϕ_e, and define $\mathbf{q}_{p'} = \{x_{p'}, y_{p'}\}^T$ and $\phi_{p'}$ to be the pursuer pose and heading in this system (figure 3.5).

Since the switching between collision and collision-free states is happens when the set B_{ep} intersect with ϕ_0, we are interested in finding what is set of pursuer's states that lies along the switching surface. In other words, we want to finds the set of relative states $\{\mathbf{q}_{p'}^T, \phi_{p'}\}^T$ such that in the global system one boundary of the set B_{ep} coincides with the evader's heading, that is, $\phi_e = \hat{\beta}$.

Without loss of generality we assume the case where $\hat{\beta} = \beta_f$, but a similar argument is valid when $\hat{\beta} = \beta_r$.

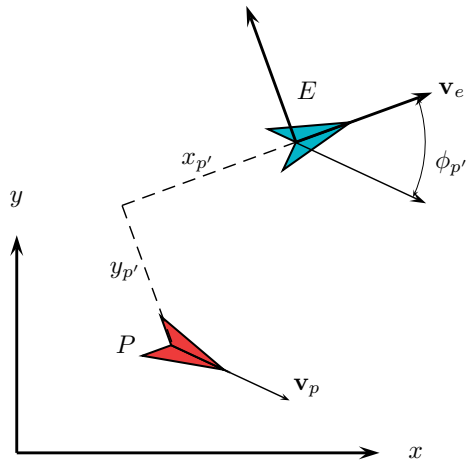


Figure 3.5: Relative frame of reference centered with the evader.

In the new frame of reference, it is possible to rewrite our collision avoidance constitutive equations (3.3) and (3.9) as:

$$\lambda' = \arctan\left(\frac{y_{p'}}{x_{p'}}\right) + \arcsin\left(\frac{R_{pz}}{\|\mathbf{q}_{p'}\|}\right) \tag{3.16}$$

$$0 = \tan\lambda'\left(v_e - v_p\cos\phi_{p'}\right) + v_p\sin\phi_{p'} \tag{3.17}$$

Assuming equal linear velocities between the two agents, that is $v_e = v_p$ it is possible to rewrite (3.17):

$$\tan\lambda' = \frac{\sin\phi_{p'}}{\cos\phi_{p'} - 1} \tag{3.18}$$

and substituting (3.16):

$$\tan\left(\arctan\left(\frac{y_{p'}}{x_{p'}}\right) + \arcsin\left(\frac{R_{pz}}{\|\mathbf{q}_{p'}\|}\right)\right) = \frac{\sin\phi_{p'}}{\cos\phi_{p'} - 1}. \tag{3.19}$$

Every pose $\{q_{p'}^T, \phi_{p'}\}^T$ that satisfy equation (3.19) represents a relative configuration between evader and pursuer such that $\phi_e = \beta_f$.

We will refer to the set of conditions satisfying equation (3.19) as *Switching Guard Conditions* and we will call this set $\mathcal{C}$:

$$\mathcal{C} = \{\{\mathbf{q}_{p'}^T, \phi_{p'}\}^T \in \mathbb{R}^2 \times \mathbb{S} \mid \phi_e = \hat{\beta}\} \tag{3.20}$$

Figure 3.6 represents these set of points within an arbitrary hyper-rectangle $[x_{min}, x_{max}]\times [y_{min}, y_{max}] \times [\phi_{min}, \phi_{max}]$.

Finally, poses from evader's relative reference frame to global frame can easily be obtained using a proper rotation matrix:

$$\begin{bmatrix} x_p \\ y_p \\ \phi_p \end{bmatrix} = \begin{bmatrix} \cos\,\phi_e & -\sin\,\phi_e & 0 \\ \sin\,\phi_e & \cos\,\phi_e & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{p'} \\ y_{p'} \\ \phi_{p'} \end{bmatrix} \tag{3.21}$$

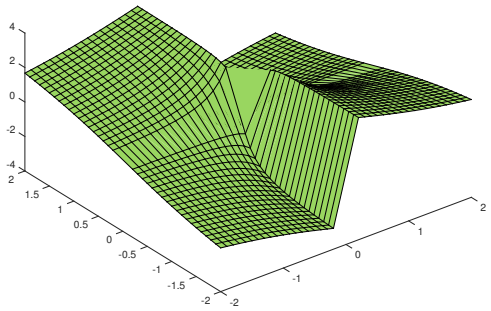


Figure 3.6: Set of pursuer's initial poses for the forcing maneuver case, expressed in evader's frame of reference.

Agents' configurations satisfying equation (3.19) represents switching conditions between the collision and collision-free states of the system. As such, when the relative state $\{q_{p'}^T, \phi_{p'}\}^T \in \mathcal{C}$, it is possible to consider the system switching from $\nu = 0$ to 1.

3.3.2 Collision and Collision-Free State

Assuming the relative state of the system $\{q_{p'}^T, \phi_{p'}\}^T \in \mathcal{C}$ and considering $\nu = 1$, the set of equations represented in (3.15) reduces to:

$$\begin{aligned}
\dot{x}_e &= v_e\, cos\phi_e \qquad & \dot{x}_p &= v_p\, cos\phi_p \\
\dot{y}_e &= v_e\, sin\phi_e \qquad & \dot{y}_p &= v_p\, sin\phi_p \\
\dot{\phi}_e &= k_e(\phi_e - \hat{\beta}) \qquad & \dot{\phi}_p &= \omega_p \\
0 &= v_e \left(\tan\lambda\, \cos\hat{\beta} - \sin\hat{\beta}\right) + v_p\, (\, \tan\lambda\, \cos\phi_p + \sin\phi_p) \qquad (3.22) \\
0 &= \lambda - \arctan\left(\frac{y_p - y_e}{x_p - x_e}\right) \mp \arcsin\left(\frac{R_{pz}}{\|q_p - q_e\|}\right).
\end{aligned}$$

where evader's heading dynamics is governed by an proportional controller having gain k_e weighting the error between the current heading ϕ_e and the desired heading $\hat{\beta}$. In this case, $\hat{\beta}$ is defined as in section 3.2. Assuming the state of the system $\nu = 1$,

our purpose is to define a suitable control inputs for the pursuer, namely v_p and ω_p, such that the sub-component $\{q_e^T, \phi_e\}^T$ of the system is driven towards a point in the target set $\{q_T^T, \phi_T\}^T \in \mathcal{T}$.

Before describing a solution for the control problem just introduced, we introduce the set of equations representative of the system where no collisions are detected and therefore $\nu = 0$. In this case the set of equations (3.15) reduces to:

$$\begin{aligned} \dot{x}_e &= v_e\, cos\phi_e & \dot{x}_p &= v_p\, cos\phi_p \\ \dot{y}_e &= v_e\, sin\phi_e & \dot{y}_p &= v_p\, sin\phi_p \\ \dot{\phi}_e &= \omega_e(\phi_e - \hat{\beta}) & \dot{\phi}_p &= \omega_p \end{aligned} \tag{3.23}$$

3.3.3 Model Predictive Control Formulation

In this section we describe the model predictive control (MPC) strategy used to steer the evader into the desired capture set. In MPC ([41]), a dynamic programming approach is used to minimize a cost function over a given finite number N of steps.

Purpose of the optimization problem is to find a set of control inputs for the pursuer, namely v_p and ω_p, such that the distance from the evader's state and the target $\{q_T^T, \phi_T\}^T$ is minimized. Similarily to what described in the previous section, the complete strategy will be composed once again by two sequential maneuvers which we will study separately:

1. **Approaching maneuver**: assuming the system to be in collision free state $\nu = 0$ at initial time, the pursuer will try to drive the relative state to the point $\{q_{p'}^T, \phi_{p'}\}^T$ which satisfies equation (3.19);

2. **Forcing maneuver**: when the system is in collision state, the pursuer will drive the evader's state $\{q_e^T, \phi_e\}^T$ to the target set, while maintaining $\nu = 1$ at all times.

3.3.3.1 Approaching Maneuver Predictive Model

In order to realize the control required to perform the set of maneuver just described, we first introduce a discrete time domain indexed by time index k and we let Δt being the constant time step. Given the nature of the control problem defined as *approaching maneuver*, a relative state between pursuer and evader must be reached. For this reason, we introduce the finite difference two vehicles model, where pursuer's dynamics is expressed in the evader's reference frame:

$$\begin{aligned} x_{p'}^{k+1} &= x_{p'}^{k+1} + (\omega_p y_{p'}^k - v_p + v_e \cos\phi_{p'}^k)\Delta t \\ y_{p'}^{k+1} &= y_{p'}^{k+1} - (\omega_p x_{p'}^k - v_e \sin\phi_{p'}^k)\Delta t \\ \phi_{p'}^{k+1} &= \phi_{p'}^{k+1} + (\omega_e - \omega_p)\Delta t \end{aligned} \tag{3.24}$$

Defining the target relative state as $\{\tilde{x_{p'}}, \tilde{y_{p'}}, \tilde{\phi_{p'}}\}^T$, we can define a measure y_a relative to the approaching maneuver:

$$y_a = \begin{bmatrix} \tilde{x}_{p'} \\ \tilde{y}_{p'} \\ \tilde{\phi}_{p'} \end{bmatrix} - \begin{bmatrix} x_{p'} \\ y_{p'} \\ \phi_{p'} \end{bmatrix}$$

which can be used in the definition of the following cost function:

$$c_a = y_a^T R y_a + \Delta u_p^T Q \Delta u_p \tag{3.25}$$

where we defined $u_p = \{v_p, \omega_p\}^T$ and weight control matrices $R \in \mathbb{R}^{3\times3}$ and $Q \in \mathbb{R}^{2\times2}$ have been introduced in order to scale the different terms contained in the

input vector. The resulting nonlinear MPC problem is then formulated as:

$$\begin{aligned}
\min_{y_a,u_p} \quad & \sum_{i=1}^{N} c_a^i(y_a^i, u_p^i) \\
\text{s. t.} \quad & x_{p'}^{i+1} = x_{p'}^{i+1} + (\omega_p^i y_{p'}^i - v_p^i + v_e \cos\phi_{p'}^i)\Delta t \\
& y_{p'}^{i+1} = y_{p'}^{i+1} - (\omega_p^i x_{p'}^i - v_e \sin\phi_{p'}^i)\Delta t \\
& \phi_{p'}^{i+1} = \phi_{p'}^{i+1} + (\omega_e - \omega_p^i)\Delta t \\
& u_{min} \leq u_i \leq u_{max}
\end{aligned} \tag{3.26}$$

It is worth to note that, as common practice in MPC problem, the presence of the term Δu is introduced in order to bound possible oscillating behavior of u. Moreover, an appropriate Q has been choose in order to bound the control effort so that explicit constraints on the inputs are not needed.

3.3.3.2 Forcing Maneuver Predictive Model

Let's now study the problem relative to $\nu = 1$. In this case, since the final target is defined in global coordinates, the use of the relative dynamics model does not represent a simplification. For this reason, the predictive model will be based on the complete set of equations and constrained contained in (3.22). However, given the significant nonlinearities introduced by the algebraic constraints, we will transform the continuous time DAE model into a discrete linear time varying (LTV) model.

We define the set of state variables x and control inputs u defined as follow:

$$x = \{x_e, y_e, x_p, y_p, \phi_p\}^T \qquad u = \{\lambda, \phi_e, v_p, \phi_p\}^T$$

from which the complete model in (3.22) can be expressed as $\dot{x} = f(x, u)$ and $g(x, u) = 0$. A linear prediction model is obtained using a first order Taylor approximation of

the complete DAE set around the point $\{x_0^T, u_0^T\}$, that is:

$$\begin{aligned} \dot{x^k} &= f(x_0, u_0) + f_{x|x_0}(x^k - x_0) + f_{u|u_0}(u^k - u_0) + \mathcal{O}^2 \\ g(x^{k+1}, u^{k+1}) &= g(x^k, u^k) + g_{x|x^k}(x^{k+1} - x^k) + g_{u|u^k}(u^{k+1} - u^k) + \mathcal{O}^2 \\ &= g_{x|x^k}(x^{k+1} - x^k) + g_{u|u^k}(u^{k+1} - u^k) + \mathcal{O}^2 = 0 \end{aligned}$$

where $g(x_0, u_0) = 0$ is imposed for consistency with the constraint. Neglecting the higher order terms in $\mathcal{O}^2$, the approximated model can be expressed in the more compact form:

$$\begin{aligned} \dot{x^k} &= \tilde{A}_0 + \tilde{A}(x^k - x_0) + \tilde{B}(u^k - u_0) \\ E\,(u^{k+1} - u^k) &+ F\,(x^{k+1} - x^k) = 0 \end{aligned} \tag{3.27}$$

where $\tilde{A}_0 \in \mathbb{R}^{5\times1}$,$\tilde{A} \in \mathbb{R}^{5\times5}$,$\tilde{B} \in \mathbb{R}^{5\times4}$, $F \in \mathbb{R}^{2\times5}$ and $E \in \mathbb{R}^{2\times4}$ are defined as follow:

$$\tilde{A}_0 = \left[f(x_0, u_0)\right] \quad \tilde{A} = \left[f_{x|x_0}\right] \quad \tilde{B} = \left[f_{u|u_0}\right]$$
$$E^k = \left[g_{x|x^k}\right] \qquad F^k = \left[g_{u|u^k}\right]$$

The expressions of the terms contained in the approximation matrices and their derivation are reported in appendix section 7.2.

In order to use the approximated model in the MPC formulation, the derivatives in (3.27) must be defined in a discrete time domain. Using a simple forward Euler approximation we can then write:

$$\begin{aligned} \frac{x^{k+1} - x^k}{\Delta\, t} &= \tilde{A}_0 + \tilde{A}(x^k - x_0) + \tilde{B}(u^k - u_0) \\ x^{k+1} &= (I - \Delta\, t\tilde{A})x^k + \Delta\, t\tilde{B}u^k + \Delta\, t(\tilde{A}_0 - \tilde{A}x_0 - \tilde{B}u_0) \end{aligned}$$

and rearranging:

$$\begin{aligned} x^{k+1} &= Ax^k + Bu^k + A_0 \\ E\,(u^{k+1} - u^k) &+ F\,(x^{k+1} - x^k) = 0 \end{aligned} \tag{3.28}$$

The set of equation contained in (3.28) represents a LTV model whose parameters A, B, A_0, E and F must be computed every time the MPC solver is initialized. Given the sparsity structure or the resulting model such a task can be efficiently carried out numerically. In addition to this, several optimization toolboxes offer the possibility to write adaptive model predictive control, in which a single initialization of the optimal problem is performed. In this way it is possible to exploit the constant structure of the problem and update the varying parameters without the necessity to write a different problem at every iterations.

Finally, introducing the following measure for the forcing maneuver y_f defined as:

$$y_f = C\,x + D\,u - \begin{bmatrix} q_T \\ \\ \phi_T \end{bmatrix} \tag{3.29}$$

where $C \in \mathbb{R}^{3\times 5}$ and $D \in \mathbb{R}^{3\times 4}$, we can formulate a cost function $c_f(z,u)$ defined as:

$$c_f(y,u) = y_f^T\,R\,y_f + \Delta u^T\,Q\,\Delta u.$$

The final formulation of our MPC problem for the forcing problem has then the following form:

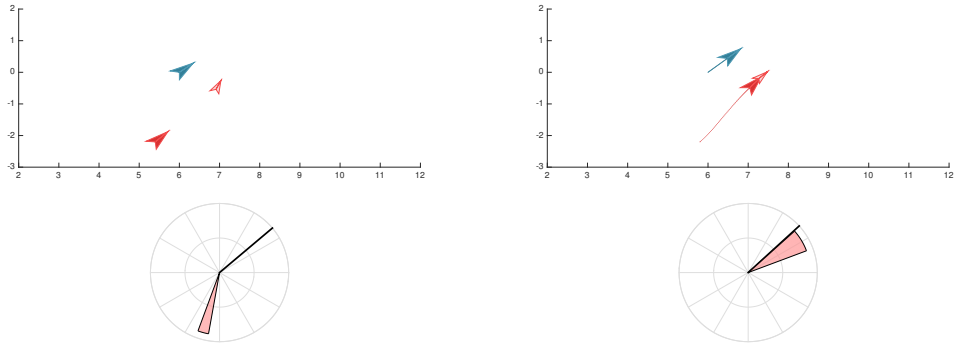
$$\begin{aligned} \min_{y_f,u} \quad & \sum_{i=1}^{N} c_f^i(y_f^i, u^i) \\ \text{s. t.} \quad & x^{i+1} = Ax^i + Bu^i + A_0 \\ & E\,(u^{i+1} - u^i) + F\,(x^{i+1} - x^i) = 0 \\ & u_{min} \leq u_i \leq u_{max} \end{aligned}$$

where, similarly to the problem in (3.26), the computed set of inputs $u_1, \ldots, u_N$ are constrained between the values u_{min} and u_{max}.

3.4 Example

3.4.1 Approaching Maneuver to Desired Relative State

In the first example we provide results from the approaching maneuver. In figure 3.7 initial configuration of both evader and pursuer is represented at initial and switching times. In figure 3.8, we note values of $\hat{\beta}$ (in blue) approaching evader's heading ϕ_E (red) until reaching the switching condition.



(a) (b)

Figure 3.7: Approaching maneuver at initial (a) and final (b) time.

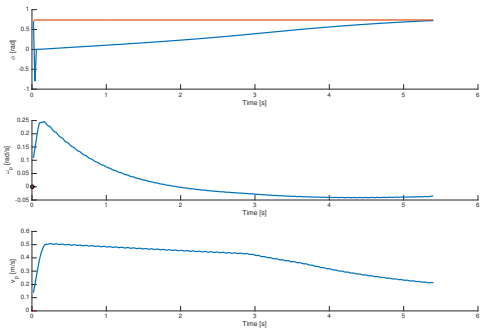
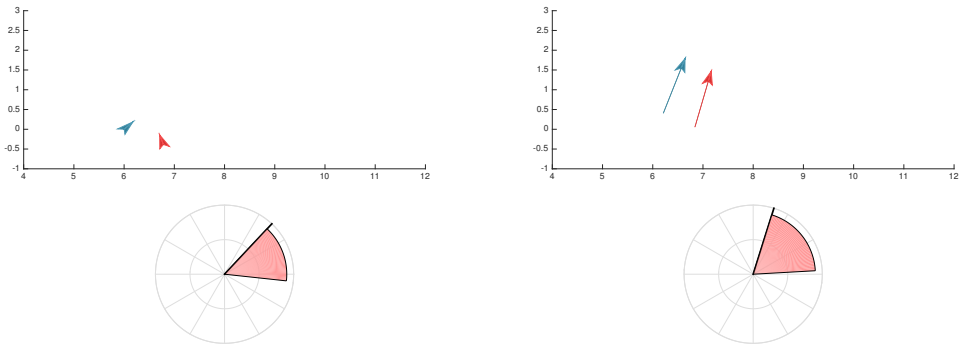


Figure 3.8: Approaching maneuver results.

3.4.2 Forcing Maneuver to Desired Heading

Once the system's state belong to the switching guard, a forcing maneuver can be initiated. Pursuer target set $\mathcal{T}$ is the direction $\phi_T = 1.25$, while evader's initial heading is $\phi_E = 0.7$ (Figure 3.9). Figure 3.10 shows that the new heading can be maintained as the pursuer adjust its input continuously.



(a) (b)

Figure 3.9: Approaching maneuver at initial (a) and final (b) time.

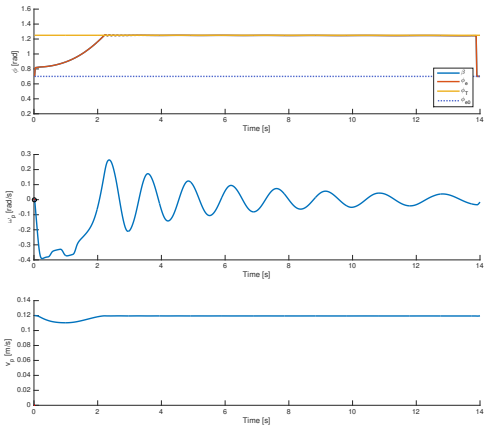


Figure 3.10: Result from the forcing maneuver.

3.5 Navigation Function Exploitation

In this section we briefly introduce a simplified alternative model based on a single integrator kinematics and artificial navigation functions. The benefit from this alternative analysis is twofold. First, we show that result presented in the previous section are not limited to the Velocity Obstacle method. Second, given the reduced analytically complexity of the model, it serves as the basis for the extension to scenarios involving a large number of agents. Purpose of this section is to anticipate the connection between the topic of this chapter and results described in the following ones.

3.5.1 Model Definition

We describe the kinematics of the agents in a discrete time domain as single integrator located in a two dimensional space, such that $\mathbf{q} = \{x; y\} \in \mathbb{R}^2$. Since all agents are considered to be equal, we drop agent index. Each agent updates its position in accordance with:

$$\mathbf{q}[k+1] = \mathbf{q}[k] + \mathbf{u}[k]\,\Delta t \tag{3.30}$$

where $\mathbf{u} = \{u_x, u_y\}^T \in \mathcal{R}$ represent velocities along the coordinate directions x and y respectively and correspond to the inputs of the system.

Artificial potential fields approach allow the design of each component of the control separately. A revised approach similar to the one proposed by Tu and Sayed [42] is considered. Using an evader-pursuer scenario similar to the one proposed in the first part of the chapter, we introduce a velocity vector defined by the sum of a term corresponding to the goal attraction and one term corresponding to the neighbor

collision avoidance. Evader's velocity is then defined to be equal to:

$$\mathbf{u}_e[k] = \frac{k_{ca}}{\|\mathbf{q}_e[k] - \mathbf{q}_p[k]\|_2} - \frac{k_g}{\|\mathbf{q}_e[k] - \mathbf{q}_T\|_2} \tag{3.31}$$

where $\mathbf{q}_T$ once again, represents location of the goal and $\mathbf{q}_p$ the location of the pursuer; k_g and k_{ca} are non-negative weighting factor relative to the collision avoidance and goal attraction respectively. Pursuer model is simply defined as in equation (3.30) and control objective is to find $\mathbf{u}_p$ necessary to drive $\mathbf{q}_e$ to a desired target configuration $\mathbf{q}_T \in \mathcal{T}$ as defined previously.

In Figure 3.11 simulation from a one evader one pursuer case are reported. The target region $\mathcal{T}$ is represented as a circular region in the space and, since we use point vehicles, a target state is completely defined on the plane. A piece-wise linear trajectory has been assumed for the pursuer, and the evader is steered into the target set.

3.5.2 Conclusions

In this chapter we showed that the presence of a predictable deconfliction policy introduces a dependency from the environment in the trajectory of an autonomous system. As discussed, from an air traffic management point of view, this is has to be considered a threat because control of a vehicle will be eventually lost. To this end, a comprehensive study of the trustworthiness of an autonomous system should include study of potential abuses of the GNC algorithms used and their dependencies.

In the next chapter we extend the dependency on a collision avoidance algorithm into a general distance-based behaviors for a large, potentially infinite, number of vehicles. Our starting point will be the results introduced in the last part of this chapter, where we showed that even with a simplified model similar results can be obtained.

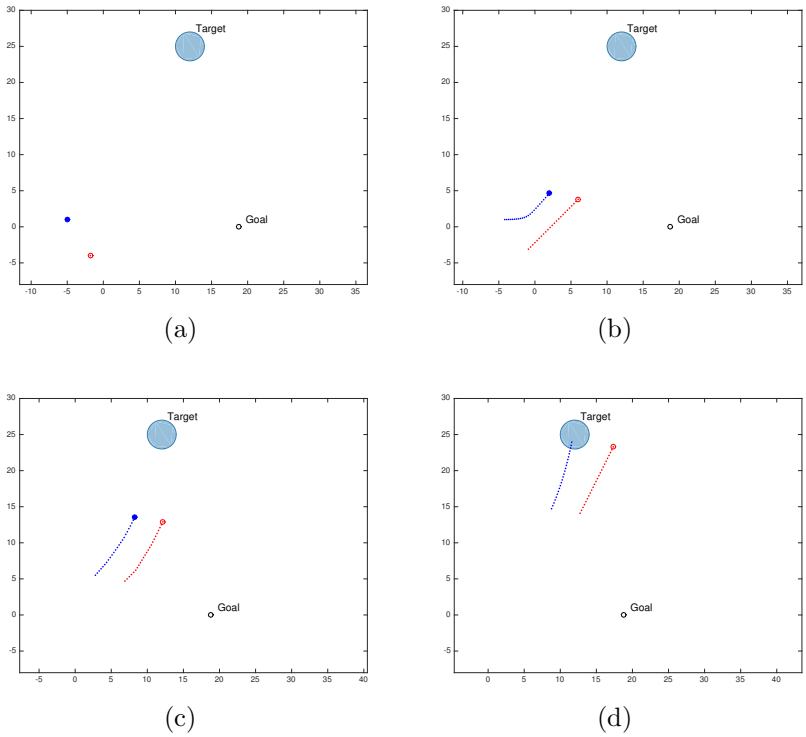


Figure 3.11: Non-cooperative trajectory modification using navigation function model.

Such simplification will be obtained by the introduction of artificial potential fields in conjunction with graph theoretic results.

CHAPTER 4

A Low Complexity Distance-Based Aggregation

4.1 Overview

This chapter investigates a low-complexity aggregation behavior suitable for large groups of homogeneous mobile robots. In formation control problems it is usually desired to keep inter-agent distances at predefined values. However, this problem requires either agents to be distinguishable or some combinatorial optimization to solve the target-agent assignment. Moreover, in many applications or mission stages, a specific shape is not required. In these cases, the simplest possible case corresponds to all agents trying to maintain the same fixed relative distance. However, since it is not possible to have more than three equally separated points on a plane, we relax this problem by allowing the inter-agent distance to be as close to desired distance as possible. To this end, a quadratic potential proportional to the error from the desired distance is introduced and equilibrium with non-null potentials are considered. Rigidity theory, which has recently emerged as a powerful tool for the study multiagent systems, is engaged to describe equilibrium and stability of our protocol. Finally, the proposed distributed protocol could serve in applications where

the requirement on the formation must satisfy only the connectivity of the swarm and limited sending capability are available.

4.1.1 Graph Theory

Generally, due to the limited range of sensors and communication devices, each agent can only see a subset of the entire group. It is natural to represent the exchange of information occurring between the agents at an higher level using directed or undirected graphs [43](Fig. 4.1). In this case, it is of primarily interest to point out to the fact that, in general, the graph induced by the agents will be time varying by nature. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by a finite set of verteces $\mathcal{V} = \{1, ..., n\}$ and a set of *edges* $\mathcal{E} = \{1, \ldots, m\}$ connecting them. Therefore, the set $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ collects ordered pairs of nodes, where an edge $(i, j) \in \mathcal{E}$ represents a communication link or visibility between the couple of nodes i and j. For agent i^{th}, the neighborhood $\mathcal{N}_i$ can be composed by the set of agents within a certain distance or those for which a communication link exists. The number of neighbors for i^{th} agent is written as $|\mathcal{N}_i|$. Following the notation introduce in [2],we use the shorthand x^{-i} for the location of vehicle i's neighborhood, that is: $x^{-i} = \{x_{j_1}, \ldots, x_{|\mathcal{N}_i|}\}$ where $j_k \in \mathcal{N}_i$.

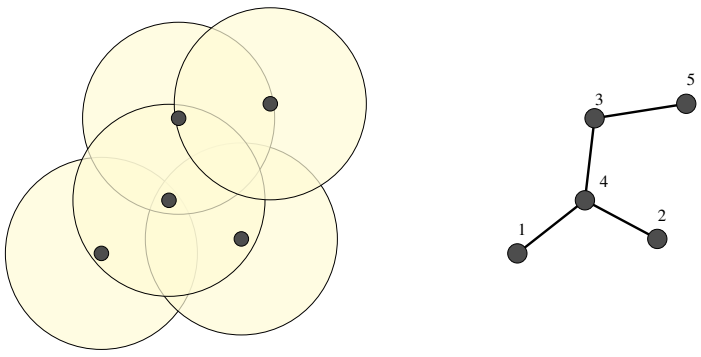


Figure 4.1: Representation of a sensor range induced graph over a set of mobile agents

Assuming labeled and arbitrarily oriented edges, the incidence matrix $D \in \mathbb{R}^{n \times m}$ has entry $d_{ik} = 1$ if i is head for edge k, $d_{ik} = -1$ if i is tail for edge k and 0 otherwise. Laplacian and edge Laplacian of the graph $\mathcal{G}$ are symmetric matrices defined as $L = D\,D^T \in \mathbb{R}^{n \times n}$ and $L_e = D^T\,D \in \mathbb{R}^{m \times m}$.

The distance vector between two nodes can be expressed as $z_k = x_i - x_j, \forall k \in \{1, \ldots, m\}$ and $\forall (i,j) \in \mathcal{E}$. The corresponding compact form is $z = \{z_1, \ldots, z_m\} \in \mathbb{R}^{2m}$. Similarly, we can define a *unit edge vector* b_k as the vector of length one along edge k, where $b_k = z_k / \|z_k\|$ and $b = \{b_1, \ldots, b_m\} \in \mathbb{R}^{2m}$ (Fig. 4.2). Using the definition of incidence matrix, it is possible to express the following relationship between nodes and edges $z = D^T\,x$.

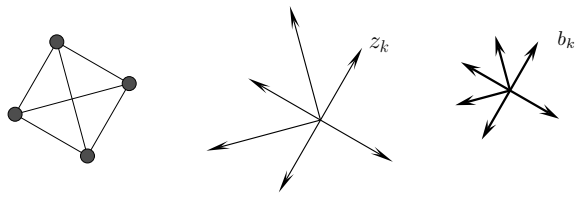


Figure 4.2: Edge vector and normalized edge vector for a complete four-edges graph.

The following definitions are also relevant for the following discussion:

Definition 3 *An undirected graph $\mathcal{G}$ is connected if there exists a sequence of adjacent nodes from any two distinct nodes.*

4.1.2 Graph Rigidity

In the following, we will make use of results related to graph rigidity theory [44] [45].

Definition 4 *A framework, sometimes referred to as bar-and-joint framework $\mathcal{G}(x)$ is defined as the embedding of the graph $\mathcal{G}$ in an Euclidean space, where x represents the geometrical location of its nodes.*

Let us introduce the following edge functions $E : \mathbb{R}^{2n} \times \mathcal{G} \to \mathbb{R}^m$ defined as:

$$E(x, \mathcal{G}) \quad = \{\|z_1\|^2, \ldots, \|z_m\|^2\}$$

Many possible definition of rigidity are available. The following can be found in [46] and [47].

Definition 5 *A framework $\mathcal{G}(x)$ and $\mathcal{G}(y)$ are equivalent if $\|x_i - x_j\| = \|y_i - y_j\|$ for all $(i, j) \in \mathcal{E}$, and are congruent if $\|x_i - x_j\| = \|y_i - y_j\|$ for all $(i, j) \in \mathcal{V}$.*

Definition 6 *A framework $\mathcal{G}(x)$ is globally rigid if every framework which is equivalent to $\mathcal{G}(x)$ is also congruent to it.*

Along with the definition of global rigidity, there is an other fundamental concept, which is *infinitesimal rigidity*. The idea behind infinitesimal rigidity is to allow infinitesimal movements in the nodes allowing only second order variation in the function $E(x, \mathcal{G}(x))$. Assuming δx to be an infinitesimal displacement of the nodes of the framework, the Taylor series expansion of $E(x, \mathcal{G}(x))$ is:

$$E(x, \mathcal{G}(x + \delta x)) = E(x, \mathcal{G}(x)) + R(x)\delta x + \mathcal{O}(\delta x^2) \tag{4.1}$$

where we introduced $R(x)$ to be Jacobian of $E(x, \mathcal{G}(x))$ with respect to x. The edge function stays constant up to the first order as long as $R(x)\delta x = 0$, which happens when $\delta x \in \text{null}R(x)$. Since at least the three rigid motions in the plane occurs without changes in the edge function, we conclude that null$R(x)$ has at least dimension three.

Definition 7 *A framework $\mathcal{G}(x)$ is infinitesimally rigid in the plane if dim(null$R(x)$) = 3, or rank$R(x) = 2n - 3$.*

It is possible to show that the *rigidity matrix* $R(x) \in \mathbb{R}^{m \times 2n}$ defined as $\partial E/\partial x$ can also be expressed as:

$$R(x) = \text{diag}(z_k)(D^T \otimes I_2) \tag{4.2}$$

In a similar way it is possible to compute an other useful operator that will be used in the following results. We introduce the *normalized edge function* as:

$$F(x, \mathcal{G}) \quad = \{\|z_1\|, \ldots, \|z_m\|\}$$

From which the *normalized rigidity matrix* $A(x) \in \mathbb{R}^{m \times 2n}$ is defined as $\partial F/\partial x$. It can easily be showed that

$$A(x) = \text{diag}(b_k)(D^T \otimes I_2). \tag{4.3}$$

4.1.3 Example

Let us consider a framework composed by four nodes and six edges as in figure 4.3.

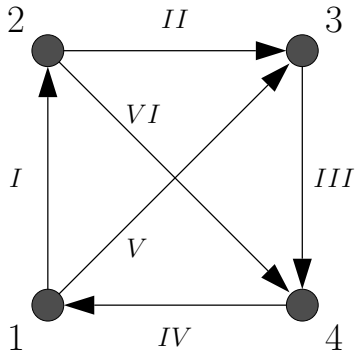


Figure 4.3: Four nodes example framework.

Given the arbitrarily oriented edges as shown, the incidence matrix will be as follow:

$$E = \begin{bmatrix} 1 & 0 & 0 & -1 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 & -1 \end{bmatrix}$$

Having defined each edge vector as $z_k = \{\Delta x_k, \Delta y_k\}^T$, we can express the rigidity matrix as:

$$R(x) = \text{diag}(z_k^T)(E^T \otimes I_2) =$$

$$\begin{bmatrix} z_1^T & 0 & 0 & 0 & 0 & 0 \\ 0 & z_2^T & 0 & 0 & 0 & 0 \\ 0 & 0 & z_3^T & 0 & 0 & 0 \\ 0 & 0 & 0 & z_4^T & 0 & 0 \\ 0 & 0 & 0 & 0 & z_5^T & 0 \\ 0 & 0 & 0 & 0 & 0 & z_6^T \end{bmatrix} \begin{bmatrix} I_2 & -I_2 & 0 & 0 \\ 0 & I_2 & -I_2 & 0 \\ 0 & 0 & I_2 & -I_2 \\ -I_2 & 0 & 0 & I_2 \\ I_2 & 0 & -I_2 & 0 \\ 0 & I_2 & 0 & -I_2 \end{bmatrix} = \begin{bmatrix} z_1^T & -z_1^T & 0 & 0 \\ 0 & z_2^T & -z_2^T & 0 \\ 0 & 0 & z_3^T & -z_3^T \\ -z_4^T & 0 & 0 & z_4^T \\ z_5^T & 0 & -z_5^T & 0 \\ 0 & z_6^T & 0 & -z_6^T \end{bmatrix}$$

and in a similar way the corresponding normal rigidity matrix:

$$A(x) = \mathrm{diag}(b_k^T)(E^T \otimes I_2) =$$

$$\begin{bmatrix} b_1^T & 0 & 0 & 0 & 0 & 0 \\ 0 & b_2^T & 0 & 0 & 0 & 0 \\ 0 & 0 & b_3^T & 0 & 0 & 0 \\ 0 & 0 & 0 & b_4^T & 0 & 0 \\ 0 & 0 & 0 & 0 & b_5^T & 0 \\ 0 & 0 & 0 & 0 & 0 & b_6^T \end{bmatrix} \begin{bmatrix} I_2 & -I_2 & 0 & 0 \\ 0 & I_2 & -I_2 & 0 \\ 0 & 0 & I_2 & -I_2 \\ -I_2 & 0 & 0 & I_2 \\ I_2 & 0 & -I_2 & 0 \\ 0 & I_2 & 0 & -I_2 \end{bmatrix} = \begin{bmatrix} b_1^T & -b_1^T & 0 & 0 \\ 0 & b_2^T & -b_2^T & 0 \\ 0 & 0 & b_3^T & -b_3^T \\ -b_4^T & 0 & 0 & b_4^T \\ b_5^T & 0 & -b_5^T & 0 \\ 0 & b_6^T & 0 & -b_6^T \end{bmatrix}$$

4.2 Swarming and Aggregation Protocol

4.2.1 Model Definition

Assuming a point kinematic model for each vehicle, we can express each dynamic as a first order differential equation:

$$\dot{x}_i = u_i, \quad i \in \{1, \ldots, n\} \tag{4.4}$$

where $u_i \in \mathbb{R}^2$ is χ_i's control input. In order to reduce the mechanical complexity and power consumption of each robot, we assume the only measurements available to robots are relative position between neighbors. Such a set of measurements could be obtained combining information on relative distances (i.e. sonar, lidar, infrared sensors) with the orientation respect to a global reference (i.e. magnetometer, radio beacon, light beam). We also assume each agents as indistinguishable to its peers and no direct communication exists between agents nor with the external world. Based on the sensing assumptions just introduced and given the limited range of real sensors,

the graph topology of the underlying network is defined by the s-proximity graph, where s is the radius of the circular sensing skirt around each robot. We call $\mathcal{N}_i$ the set of i^{th}'s neighboring agents defined as:

$$\mathcal{N}_i = \{\chi_j \in \chi \,|\, \|x_i - x_j\| \leq s\}.$$

As mentioned earlier, each agent's objective is to maintain a fixed relative distance $r \in \mathbb{R}$ from its visible neighbors, where $r < s$. We introduce the following positive semi-definite potential existing between two nodes sharing an edge:

$$\varphi(x_i, x_j) = \frac{1}{2}(\|x_i - x_j\| - r)^2 \tag{4.5}$$

which captures the fact that, if no constraints acts on the dynamics of χ_i and χ_j, the zero potential condition corresponds to an inter-agent distance equal to $r < s$. It is worth to note, as it will be showed later, that interactions with other agents are in fact constraints to the group's dynamics.

A velocity field can be defined as the negative gradient of the potential in (4.5) and χ_i's input control can then be expressed as:

$$\begin{aligned} u_i &= -\sum_{j \in \mathcal{N}_i} \nabla_x \varphi(x_i, x_j) \\ &= -\sum_{j \in \mathcal{N}_i} \left(\frac{\|x_i - x_j\| - r}{\|x_i - x_j\|} \right) (x_i - x_j) \\ &= -\sum_{j \in \mathcal{N}_i} w_{ij}(x_i - x_j) \qquad i = 1, \ldots, N. \end{aligned} \tag{4.6}$$

The parameter w_{ij} can be thought as the nonlinear weight associated with the edge connecting nodes i and j. To this end, by introducing the real valued diagonal weight matrix $W = \text{diag}(w_{ij}) \in \mathbb{R}^{m \times m}$, we can express the whole swarm dynamics as:

$$\dot{x} = -L_w(x)\, x \tag{4.7}$$

where $L_w(x) = D\,W(x)\,D^T$ is the weighted Laplacian of the graph $\mathcal{G}$. It is worth to note that, as result of the potential defined in (4.5), weights w_{ij} can be negative real number and therefore, stability analysis based on the positive semi-definiteness of the graph Laplacian are not applicable. Moreover, given the non-positiveness of the weights, the resulting behavior is not a pure agreement process but rather an hybrid agreement/disagreement process that keeps the agents within a certain distance.

4.2.2 Study of the Equilibrium

Following the negative gradient of the potential field in (4.5), each agent tries to maintain a desired distance equal to r from all its neighbors. However, assuming an arbitrarily number of neighbors within the range of any given agent, it is not possible to have more than three equally separated points in a 2-dimensional space. For this reason, in general, for a group of $n > 3$ robots whose dynamics evolves as in equation (4.6), the total potential at equilibrium is

$$\varphi^*(\mathcal{G}, x) = \sum_{(i,j)\in\mathcal{E}} \varphi^*(x_i, x_j) \geq 0. \tag{4.8}$$

Note that we introduced the superscript * to express quantities relative to the equilibrium state. It is possible to give a physical interpretation of this problem by considering four masses connected by six springs (four along the edges of the quadrilateral and two along its diagonal) with unitary elastic constant and a pre-loaded length equal to r. In this case, the potential energy of each spring will have the form in (4.5) and equilibrium configurations do not corresponds to null forces on the springs but rather on null resultant forces on each mass. Mechanical and electromagnetic interpretation of multiagent systems have been largely investigated. In [48] a Lagrangian approach was applied to study the synchronization stability of

a multiagent robotic system, while in [49] the concept of *graph effective* resistance was introduced to describe the pairwise measurements based control for a network of autonomous agents.

In Fig. 4.4, different outcomes from the same aggregation process are reported. Different results are obtained when different sensing radii s are considered. This observation reflects the obvious fact that a higher sensing radius increases the degree of a vertex. It is also worth to note that, when assuming the same sensing range, different equilibrium configurations are possible, from which we conclude the non-uniqueness of the equilibrium (Fig. 4.4 b) and c)).

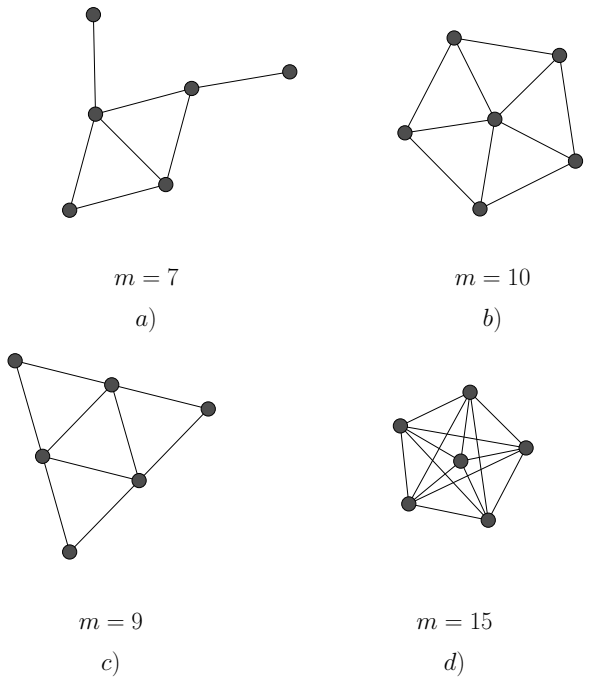


Figure 4.4: Different equilibrium solutions for the same aggregation dynamics. Result reported for different values of sensing range $\frac{s}{r}$: a) 1.14; b) and c) 1.2; d) 1.51.

Lemma 2 *An equilibrium point x^* exists for the dynamics in (4.7).*

Proof: Considering a $k \in \mathbb{R}$, the k^{th}-level set of the total potential $\varphi(\mathcal{G}, x)$ is:

$$\Omega_k = \{x \,|\, \varphi(\mathcal{G}, x) = k\} \tag{4.9}$$

Also, from the time derivative of the total potential:

$$\begin{aligned}\dot{\varphi}(\mathcal{G}, x) &= \nabla_x \varphi(\mathcal{G}, x))\dot{x} \\ &= -\sum_{i=1}^{n} \dot{x}_i^T \dot{x}_i = -\sum_{i=1}^{n} x_i^T L_w(x)^2 x_i \leq 0\end{aligned}$$

we see that $\dot{\varphi}(\mathcal{G}, x)$ is a negative semi-definite function. Since the systems follows the negative gradient of the potential, once $\dot{\varphi}(\mathcal{G}, x) \in \Omega_k$ any changes in the configuration of the system will move the potential to a new Ω_ℓ, with $\ell \leq k$. From this follows that Ω_k is an invariance set for the dynamic of the system defined in (4.7) and therefore, from the continuity of (4.5), a minimum k^* must exists, such that:

$$\Omega_{k*} = \{x^* \,|\, \varphi(\mathcal{G}, x^*) = k^* < k,\, \forall k \in \mathbb{R}\} \tag{4.10}$$

■

Given the result from 2 and the constrain $\varphi(\mathcal{G}, x) \geq 0$ described earlier, we can also conclude that $\varphi(\mathcal{G}, x)$ is a weak Lyapunov function [50] for the system described in (4.7).

4.2.3 Connection with Rigidity Theory

We observed that when the number of edges in the graph exceeds a certain threshold, the constraints on the dynamics prevent the swarm from reaching an equilibrium configuration for which $\varphi^* = 0$. As it will be shown later, the presence of these additional constraints, increase the complexity of the model but do not prevent the swarm from reaching the desired behavior.

In order to reduce the ambiguity associated with this observation, we provide here some results connected with rigidity theory. Many possible definitions of rigidity are possible; we only provide here some insights, while a more complete description can be found in [51]. A framework is *infinitesimally rigid* if the only nodes' movements for which the length of every edges is preserved up to first order are rigid translations and rotation (also known as *rigid modes* in modal analysis); trivially, a framework is flexible if it is not rigid. A framework is minimally infinitesimally rigid, if it is rigid and the removal of any edge results in a flexible framework. Given these definition, the infinitesimally degree of rigidity is dropped and implicitly assumed.

As it mentioned earlier, given a d-dimensional domain, the maximum number of equally separated points is $d + 1$. The corresponding configuration in a planar domain is the geometrical representation of an equilateral triangle. We define an ℓ-edge equilateral triangle, a triangle having all edges length equal to ℓ.

Theorem 2 *If a framework is composed by ℓ-edge equilateral triangles that share exactly two nodes is minimally rigid.*

Proof: We provide the proof to this theorem by induction. We first prove that an equilateral triangle is minimally rigid. The proof is trivial, since any triangle is a rigid framework and the removal of any edge makes the framework flexible.

Now, let us consider a framework $\mathcal{T}$ to be composed by t of these triangles, where each new triangles was created by adding a node and two edges of length ℓ to an existing edge (Fig. 4.5. Assuming $\mathcal{T}$ to be minimally rigid, its only degrees of freedom (*dof*) corresponds to the three rigid motions. Now, if a new triangle is added to an existing edge of $\mathcal{T}$ and since an edge is in common, the corresponding three

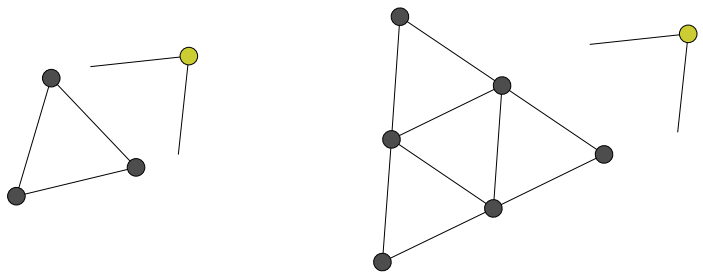


Figure 4.5: Minimally rigid framework composition process.

dof added by the new node have to coincide with the *dof* of the complete $\mathcal{T}$. This conclude the proof. ■

Given the framework construction described above (Fig. 4.5), we note that the resulting graph formed by t triangles has exactly $t+2$ nodes and $2t+1$ edges. Note that the notion of framework composed by equally separated neighbors has been widely studied and corresponds to the α-lattice configuration defined in [15].

Corollary 1 *A minimally rigid framework described in theorem 2 having n nodes has* $m = 2n - 3$ *edges.*

From this result, knowing the number of nodes in the network, we can compute the maximum number of edges for which an equal separation by visible neighbors can exist. It is possible to conclude that when a framework is at most minimally rigid, it is possible to have equilibrium corresponding to $\varphi^* = 0$. However, as mentioned before, when the graph resulting from the dynamics in (4.6) is not-minimally rigid it is still possible to have equilibrium.

4.3 Aggregation Behavior Dynamics

In this section we study the dynamic properties of the protocol introduced in the previous section. We start with a well known results from network dynamics.

Lemma 3 *The centeroid of the connected swarm under the dynamics defined as in (4.7) is stationary.*

Proof: The centeroid of the swarm is defined as $\hat{x} = 1/n \sum_{i=1}^{n} x_i$, its velocity:

$$\frac{1}{n}\mathbf{1}^T \dot{x} = -\frac{1}{n}\mathbf{1}^T D\, W(x)\, D^T x$$

Since, for a connected graph, $\mathbf{1}^T \in \text{Null}\, D$, the centeroid $\hat{x}$ is a fixed point defined by the swarm initial conditions. ■

The result in lemma 3 is a very well know results in multiagent networks [43] and it is relevant for following discussion.

It is possible to rewrite the single agent's dynamic defined in the second line of (4.6) as:

$$\begin{aligned} u_i &= -\sum_{j \in \mathcal{N}_i} \left(\|x_i - x_j\| - r\right) \frac{(x_i - x_j)}{\|x_i - x_j\|} \\ &= -\sum_{k} \left(\|z_k\| - r\right) b_k, \quad k \in \mathcal{E} \end{aligned} \tag{4.11}$$

where we recall b_k being the unitary vector along the k^{th}'s edge vector.

Theorem 3 *The velocity field (4.11) ensure no collisions occur between agents of the swarm.*

Proof: Considering equation (4.11) for the pair χ_i and χ_j, $u_{ij} = (\|x_i - x_j\| - r)b_k$, where b_k is the direction vector along the edge k. Thus, since:

$$\lim_{(x_i - x_j) \to 0} u_{ij} = r \tag{4.12}$$

a positive finite velocity between the agents exists. ■

Rearranging equation (4.11), it is possible to write the entire swarm dynamics in the compact form:

$$\dot{x} = -(I_2 \otimes D)\,\mathrm{diag}(b_k)\left[(R(x)\,x)^{\circ\frac{1}{2}} - r\,\mathbf{1}\right] \tag{4.13}$$

Solution of equation (4.13) represents equilibrium configuration for the dynamic of the system. The analytic definition of the *shape* at equilibrium, which is generally not unique, depends on the number of agents, the initial configuration, the desired distance d and the sensing radius s. However, recalling the definition of our swarming behavior problem, the exact geometric realization of our aggregation protocol are not of primary interest here. Instead, in order to give understand the result contained in equation (4.15), we proceed as follow. First, acknowledging lemma 3, we note that decoupling the centeroid dynamics from the remaining dynamics of the system [47], we can analyze the swarm motion from its edges dynamic. To this end, we introduce the k^{th}-edge residual vector $\delta_k = \|z_k\| - r$ for $k = 1, \ldots, m$ and the corresponding composed vector:

$$\delta = \{\|z_1\| - r, \ldots, \|z_m\| - r\}^T \in \mathbb{R}^m \tag{4.14}$$

The residual δ represents the error between the edge length and the reference distance r, and its dynamics can be used to fully describe the aggregation process of interest. Given the definition in (4.14) it is easy to show that we can rewrite (4.13) as:

$$\dot{x} = -A(x)^T\,\delta(x) \tag{4.15}$$

where $A(x)$ is the normalized rigidity matrix defined in (4.3). The corresponding solution at equilibrium x^* can then be written as the matrix equation:

$$A(x^*)^T\,\delta^* = 0 \tag{4.16}$$

from which $\delta^* \in \text{null}\, A(x^*)^T$. Recalling how $A(x)$ was introduced, some ambiguity was left in regards of its definition. It is possible to see now its close connection with the rigidity matrix $R(x)$. To this end, we note that we can write the i^{th} row of $A^T(x)$, $\{a_{ik}\}_{k=1,\dots,m}^T$ as:

$$a_{ik} = \begin{cases} b_k & \text{if } i \text{ is head for } k \\ -b_k & \text{if } i \text{ is tail for } k \\ 0 & \text{if } k \notin \mathcal{E}_i \end{cases} \tag{4.17}$$

and therefore χ_i's velocity is $\dot{x}_i = A_i(x)\,\delta$. Using this expression, the equilibrium for agent i is:

$$\sum_{k \in \mathcal{E}_i} a_{ik}\,\delta_k = 0 \tag{4.18}$$

which describes the null resultant of the *forces* acting on the agent. We can finally see that role of matrix $A(x)$ which encodes the geometrical realization of the framework by means of the edge direction b_k contained in it.

What is left to show is that a network dynamics that follows equation (4.6) maintains the connectivity of the underlying graph and the resulting equilibrium is stable. Proving that connectivity of the graph can be maintained when the topology of the graph is state-dependent is generally a not easy task. In addition, since our system can in fact switch between agreement/disagreement states, we cannot exploit the positive semidefiniteness of the weighted Laplacian in (4.7).

In order to provide results on the connectivity properties of our protocol, we can proceed as follows. Let first assume that no more than one edge can be disappear at a time and that the swarm configuration corresponds to a connected graph $\mathcal{G}$ at the initial time $t = 0$. Before being disconnected, the swarm configuration must necessarily form a bipartite graph with two components $\mathcal{G}_1$ and $\mathcal{G}_2$ connected by one

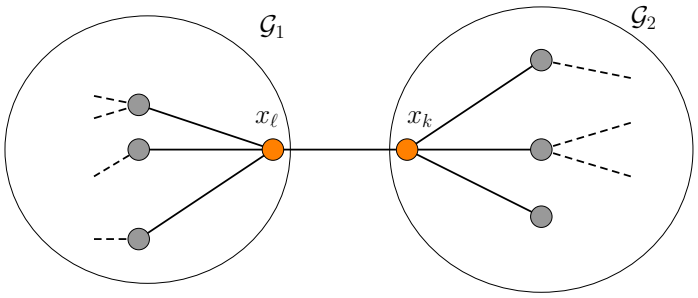


Figure 4.6: Representation of the bipartite graph connected by a single edge.

edge a length s. Consider x_ℓ and x_k being the two nodes connecting $\mathcal{G}_1$ and $\mathcal{G}_2$, such that $\|x_\ell - x_k\| = s$ (Fig. 4.6). Since no external forces are applied to $\mathcal{G}_1$ and $\mathcal{G}_2$, following the negative gradient of the potential the distance between x_ℓ and x_k will converge to $r < s$. The graph must therefore maintain its connectivity.

We finally provide the result on the stability of the equilibrium, for which a solution exists as showed in lemma 2.

Theorem 4 *Under the dynamics defined in (4.6), the edge residual at equilibrium δ^* is a stable point.*

Proof: Pre-multiplying both sides of (4.15) by $D^T \otimes I_2$ leads to:

$$(D^T \otimes I_2)\,\dot{x} = -(D^T \otimes I_2)\,A(x)^T\,\delta(x) \tag{4.19}$$

$$\dot{z} = -(D^T \otimes I_2)\,A(z)^T\,\delta(z) \tag{4.20}$$

From the definition in (4.14) we can compute the derivative of δ as:

$$\dot{\delta} = -\mathrm{diag}(z_k^T)\,\dot{z} \tag{4.21}$$

Substituting (4.20) in (4.21) leads to:

$$\dot{\delta} = -\mathrm{diag}(z_k^T)\,(D^T \otimes I_2)\,A(z)^T\,\delta(z) \tag{4.22}$$

Recalling the definition of the normalized edge vectors b_k, we can write $\text{diag}(z_k^T) = \text{diag}(\|z_k\|)\,\text{diag}(b_k^T)$, which can then be substituted in (4.22):

$$\begin{aligned} \dot{\delta} &= -\text{diag}(\|z_k\|)\,\text{diag}(b_k^T)\,(D^T \otimes I_2)\,A(z)^T\,\delta(z) \\ &= -\text{diag}(\|z_k\|)\,A(z)\,A(z)^T\,\delta(z) \\ &= -\text{diag}(\|z_k\|)\,\mathcal{A}(z)\,\delta(z) \end{aligned} \tag{4.23}$$

where $\mathcal{A}(z) = A(z)\,A(z)^T$. Given all entries of $\text{diag}(\|z_k\|)$ are non-negative and the positive semi-definiteness of the quadratic form $\mathcal{A}(z)$, edge stability at equilibrium follows. ■

Following the results in 4, we know the dynamics expressed in (4.15) will produce a stable aggregation behavior.

4.4 Numerical Results

In Fig.4.7(a) we report the result from a simulation performed considering $n = 6$ agents with limited sensing radius and assuming random initial condition. Fig. 4.7(c) shows the corresponding total potential over time. Similarly, in Fig. 4.7(b) and Fig. 4.7(d) are reported results from a swarm of $n = 12$ agents with the relative potential progression over time.

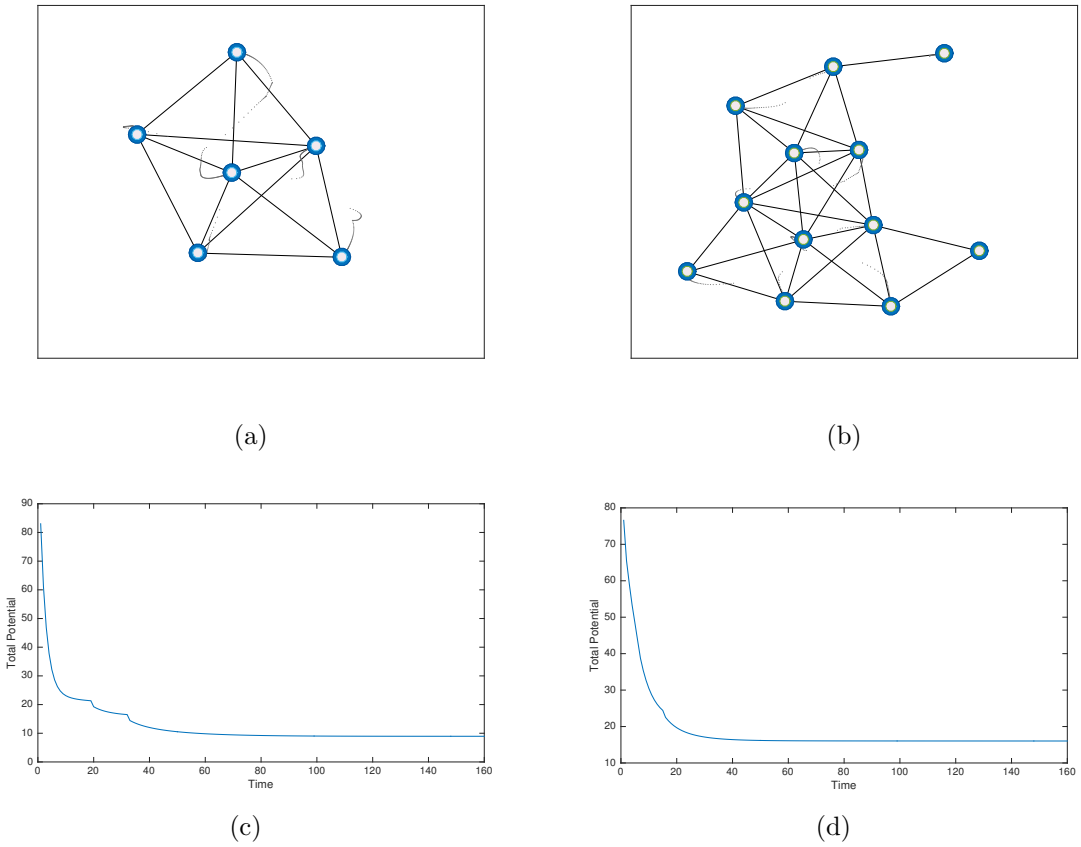


Figure 4.7: Result of the swarming protocol for a team of six and twelve agents with corresponding total potential over time.

CHAPTER 5

Control of Multiagent Systems Responding to Aggregation Behaviors

5.1 Overview

In this chapter we turn our attention to the problem of controlling the dynamics described in the previous chapter. As showed, the dynamics protocol described in the previous chapter maintains the centroid of the swarm fixed for all times. This fact is by necessity connected with null space of the incidence matrix of the corresponding connected graph. In this chapter we investigate the result of the injection of an exogenous signal into the network. In order to do so we relax the assumption made in the previous chapter and we allow one agent to communicate with the external environment. The resulting problem can therefore be described as a leader-follower problem.

5.2 Leader-Follower Problem in Control of Networks

The fundamental question of weather a networked decentralized system can be controlled has been largely investigated in recent times. Along with networked autonomous systems, the problem is also connected with security issues in cyber-physical systems. In distributed power-grid systems, safety analysis are performed by characterizing the vulnerability of its nodes. The extent to which the systems performance can be affected by malicious attack directed to one or multiple nodes can be in practice formulated as a network controllability problem. To this end, it is not surprising the connection with the trustworthiness problem we introduced in chapter 3. Assembly chains can also be conveniently modeled as networks of manufacturing phases. The minimal number of human operators required to control a process composed by many phases can be once again studied in terms of networks control. A remote operator controlling a formation of UAVs is an other example.

The fundamental problem in control of networks has been mostly formulated in terms of the controllability of the system or as a leader selection process. In [52] the problem of controlling a networks formed by nearest neighbor rules was initially studied. The topology of the underlying graph is of fundamental importance when trying to achieve control on the network. For example, symmetry with respect to the leader must be avoided in order to have control on the system [53], and the use of equitable partitions of the graph emerged as a useful tool for the study of controllability properties [54]. As the size of the problem increase, deriving results on its controllability becomes more difficult. In [55] it was showed how the number of leader

nodes depends on the degree distribution of the network. Based on the composition of controllable components and using path graphs, a techniques for construct controllable networks was described in [56]. Providing results on the controllability of a system is usually not enough for practical purposes. In this case, the performance of the controller designed must be investigated as well [57] [58].

When networked systems is of interest must be controlled on line, a leader selection process has to be designed. In this case the question concern the choice of which node(s) should be leaders and how to solve the problem in a distributed fashion. The problem becomes more complicate when dynamics graph topology and limited knowledge on the network are available. Recent results on the subject have been investigated for example in [59] and [60].

5.3 Model Definition

Let us consider the network distributed dynamics described in the previous chapter, which we repeat here for clarity:

$$\dot{x} = -A(x)^T \delta(x) \tag{5.1}$$

where $\delta \in \mathbb{R}^m$ is the vector of residual respect to the desired inter-agent distance and $A(x) \in \mathbb{R}^{m \times 2n}$ is the normalized rigidity matrix of the resulting graph.

5.3.1 Open Loop Leader-Follower Control

Under the results from Lemma 3, the dynamics in (5.1) creates no centroid motion. Now, we allow a random agents χ_ℓ to receive a step control input $u_\ell \in \mathbb{R}^2$. In this case the swarm will be partitioned in two subsets $\mathcal{L} = \chi_\ell$ and $\mathcal{F} = \chi \setminus \chi_\ell$. As shown

in Fig.5.3, we also define the set of leader's neighbors has $\mathcal{N}_\ell = \{\chi_i \in \mathcal{F} \,|\, \|x_l - x_i\| < s,\, i = 1, \ldots, n,\, i \neq \ell\}$. Two different architectures are generally possible.

Mixed input In one case it is possible to assume the external control signal u_ℓ as being added to the leader's original swarm dynamic, defined in (4.6). In this case the resulting dynamic is:

$$\dot{x} = \begin{bmatrix} \dot{x}_f \\ \\ \dot{x}_\ell \end{bmatrix} = -A(x)^T \, \delta + B \, u_\ell \tag{5.2}$$

where B is a index selector matrix having a vector of ones in correspondence of the leader's state. From (5.2) the centroid dynamics can be expressed as:

$$\dot{\hat{x}} = \frac{1}{n}(\mathbf{1}^T \otimes I_2) \left(-A(x)^T \, \delta + B \, u_\ell\right). \tag{5.3}$$

Since $(\mathbf{1}^T \otimes I_2) \, A(x)^T = 0$, the resulting centroid velocity is $\dot{\hat{x}} = \frac{1}{n} \, u_\ell$. From this expression we see that if the total number of agents composing the swarm is not known, the input will be reduced by a factor n and tracking of the reference signal is impossible.

Stubborn leaders Alternatively, it is possible to assume the dynamic of the leaders to be uniquely driven by the external input signal. This configuration corresponds to a directed graph. Without loss of generality, we assume a vertex ordering such that:

$$x_i = \begin{cases} \text{follower} \quad \text{if} \quad i = 1, \ldots, n-1 \\ \\ \text{leader} \quad\;\; \text{if} \quad i = n \end{cases} \tag{5.4}$$

By partitioning the normalized rigidity matrix into follower and leader sub-spaces, we write:

$$A(x) = \begin{bmatrix} A_f(x) \; A_\ell(x) \end{bmatrix} \tag{5.5}$$

where $A_f(x)$ and $A_\ell(x)$ are composed by the columns of $A(x)$ corresponding to follower and leader nodes respectively. The swarm dynamics can then be described by:

$$\dot{x} = \begin{bmatrix} \dot{x}_f \\ \dot{x}_\ell \end{bmatrix} = \begin{bmatrix} -A_f^T(x) \\ 0 \end{bmatrix} \delta + B\, u_\ell \tag{5.6}$$

In this case, the centeroid dynamics has the following expression:

$$\begin{aligned} \dot{\hat{x}} &= \frac{1}{n}(\mathbf{1}^T \otimes I_2) \left(\begin{bmatrix} -A_f^T(x) \\ 0 \end{bmatrix} \delta + B\, u_\ell \right) \\ &= \frac{1}{n}(A_\ell^T(x)\delta + u_\ell) \end{aligned}$$

In this case it is possible to note that the residual length on each edge play a role. By inspection of equation (5.7) we note that in addition to the leader input, the centroid dynamics is affected by the direction of the leader's edge set $\mathcal{E}_\ell$ and their relative residuals.

The two architectures just described are represented in Figure 5.1. It is worth to note, that these results are valid only when connectivity of the network is maintained. Results from a simulation with 1 stubborn leader and 11 followers are reported in Figure 5.2(a) and Figure 5.2(b). We observe the velocity of the swarm converged to the leader velocity u_ℓ in the first case (Figure 5.2(a)). When higher velocity input is considered, connectivity of the graph cannot be maintained (Figure 5.2(b)).

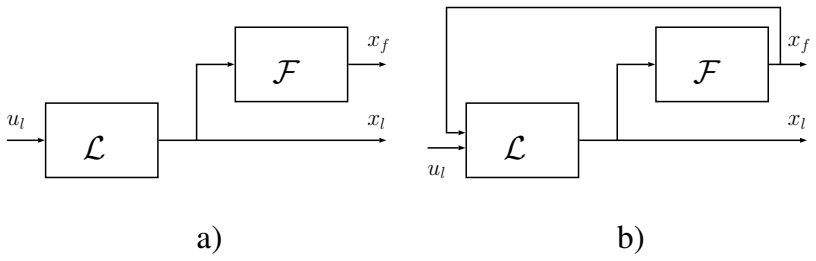
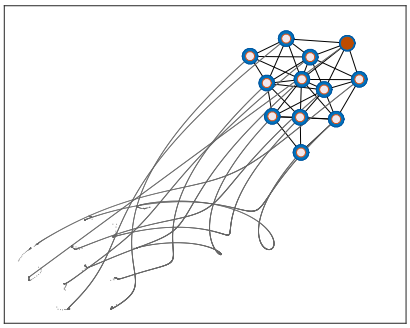
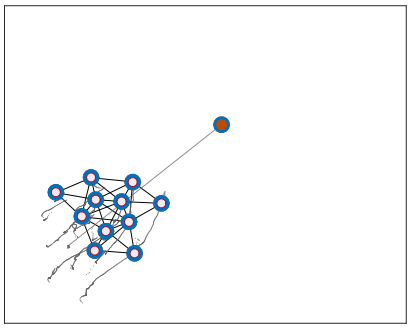


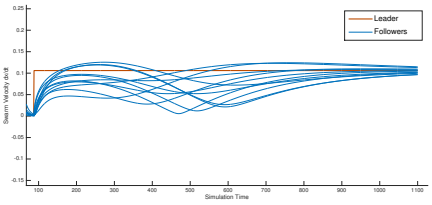
Figure 5.1: Diagram representation for the leader-follower architecture described: stubborn (left), mixed input (right).



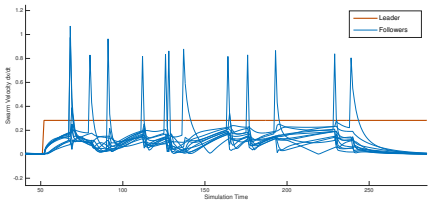
(a) Simulation results with low velocity input.



(b) Simulation results with high velocity input.



(c) $\|\dot{x}\|$ for a connectivity preserving input.



(d) $\|\dot{x}\|$ when connectivity is not maintained.

Figure 5.2: Simulation results for a stubborn leader-follower architecture, performed on a swarm with 11 followers and 1 leader subject to a step input.

5.3.2 Leader to Neighbor Distance Feedback

As shown in the previous section, the results corresponding to the two architectures just introduced hold only when connectivity of the network is maintained. Unfortunately, for a nearest neighbor rule based topology, it is not possible to guarantee connectivity of the graph for any given inputs. In this section we introduce a feedback architecture where the leader tries to track a reference velocity while maintaining the distance from its visible neighbors' below the sensing radius s. The problem we address can be introduced as follow.

Problem 1 *Given a reference velocity signal u_{ref}, find a leader input u_ℓ such that connectivity between leader and its visible neighbors is preserved.*

We divide the group of $n-1$ followers in those directly connected with the leader and the remaining agents (Figure 5.3). We denote the group of leader's neighbors as:

$$\mathcal{N}_\ell = \{\chi_i \in \chi \setminus \chi_\ell \,|\, \|x_\ell - x_i\| \leq s\}. \tag{5.7}$$

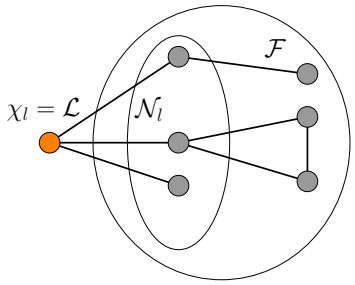


Figure 5.3: Leader's neighborhood set definition.

In order to solve Problem 1, we introduce an heuristic real loss function $\psi : \mathbb{R} \to \mathbb{R}$. The loss function will be constructed in such a way that leader's velocity is decreased when the distance from its neighbors is more then the aggregation desired distance r and increased when below. We will also assume that the leader velocity is zero whenever this distance goes to s. Before introducing ψ we define the leader's neighborhood centroid $\hat{x}_\ell$ as the centroid of $\mathcal{N}_\ell$, for which:

$$\hat{x}_\ell = \frac{1}{|\mathcal{N}_\ell|} \sum_{j \in \mathcal{N}_\ell} x_j \tag{5.8}$$

$$\dot{\hat{x}}_\ell = \frac{1}{|\mathcal{N}_\ell|} \sum_{j \in \mathcal{N}_\ell} \dot{x}_j = \frac{1}{|\mathcal{N}_\ell|} \mathbf{1}^T \, A_{N\ell}^T(x) \, \delta \tag{5.9}$$

where $A_{N\ell}^T(x)$ is the normalized rigidity matrix for the sub-space corresponding to $\mathcal{N}_\ell$. Using the quantities defined in equations (5.8) and (5.9), we can now introduce a leader input defined as follow:

$$u_\ell = u_{ref}(1 - \psi(x_\ell, \hat{x}_\ell)) \tag{5.10}$$

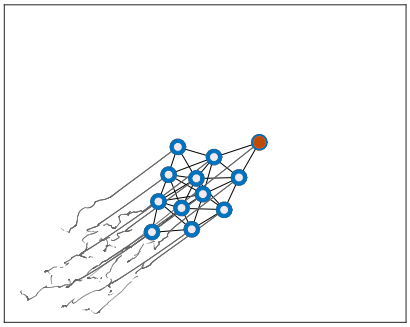
$$\psi(x_\ell, \hat{x}_\ell) = \frac{1}{s-d}(\|x_\ell - \hat{x}_\ell)\| - r). \tag{5.11}$$

The complete controlled swarm dynamic can therefore be expressed as:

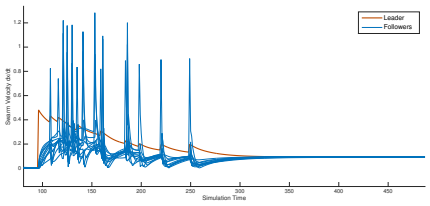
$$\dot{x} = \begin{bmatrix} \dot{x}_f \\ \dot{x}_\ell \end{bmatrix} = \begin{bmatrix} -A_f^T(x) \\ 0 \end{bmatrix} \delta + B\, u_{ref}(1 - \psi(x_\ell, \hat{x}_\ell)) \tag{5.12}$$

$$y = \frac{1}{|\mathcal{N}_\ell|} \mathbf{1}^T A_{N\ell}^T(x)\, \delta \tag{5.13}$$

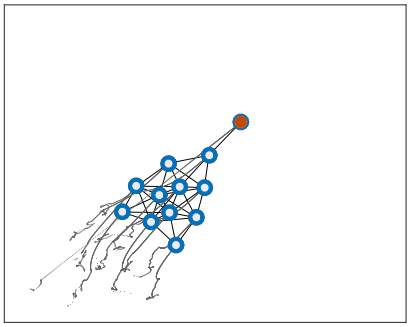
where B is the input selector defined as above and y is the leader's measure on the system. In Figure 5.4 simulations results are reported for the same swarm-size and input considered in the previous example. As it is possible to see, leader's velocity is adapted to the distance from its neighbors and connectivity is maintained until convergence



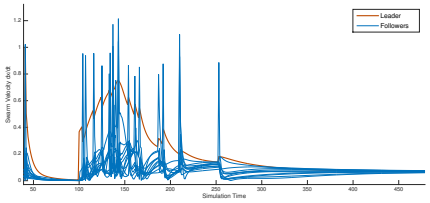
(a) Simulation results with low velocity input.



(b) $\|\dot{x}\|$ for low input.



(c) Simulation results with high velocity input.



(d) $\|\dot{x}\|$ for high input.

Figure 5.4: Simulation results for the leader-follower feedback architecture, performed on a swarm with 11 followers and 1 leader subject to a step reference input.

CHAPTER 6

Collision as Information for Robot Self Localization

6.1 Overview

In this chapter we discuss an estimation problem related to self-localization of agents in highly populated environments. In this case, measurements of relative distance (or *proximity*) from encountered agents are used for localize the robot in a tessellated environment. Assuming robots with limited size, collisions can be tolerated and actually used as a source of information. To this end, rate of encounters with other agents, can be used to obtain information about the robots' surroundings.

We consider a collection of autonomous agents moving in accordance with a known transition probability process between sectors or cells of their domain. Each robot is equipped with a tactile sensors that produces a binary measure, namely collision or not-collision state, which is captured over time. The estimate on the robot location is then recursively updated as new measurements are observed.

6.2 Model Definition

Consistently with what described in the previous chapter, let us consider a set of agents $\chi = \{\chi_1, \ldots, \chi_n\}$ on a planar domain D, divided into M non-overlapping partitions such that:

$$D = \cup_{j=1}^{M} D_j \quad D_j \cap D_i = \emptyset \quad \text{for} \quad i, j \in \{1, \ldots, M\}$$

As robots move between cells, the current fraction of robots in cell j at time step k is denoted as $\mu_j[k]$ and the composite vector over the whole domain as $\mu[k]$. Since the fraction of agent in each cell varies from cell to cell and over time, the rate of encounter between robots will vary.

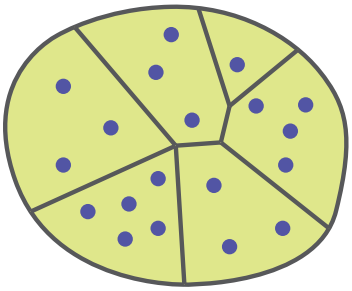


Figure 6.1: Illustration of agent distribution over a partitioned domain.

In order to quantify the probability of a collision, we introduce a *mean-field approximation* from which the probability of collision between robots can be expressed as a single probability. Letting $\phi_j[k]$ be the probability of experiencing a collision in cell j at time k and $x_i \in D$ be the position of robot i in the domain, we introduce the following assumptions:

1. The size of the domain is significantly larger compared to the footprint of a single agent.

2. We can approximated the distrbution of robots in the domain as a Poisson process, having intensity:

$$\lambda_j[k] = \mu_j \frac{N}{D_j}. \tag{6.1}$$

which represents the fraction of robots per unit area in D_j.

3. Assuming a radius r for each robot footprint radius, a collision occurs when the footprint of the robots overlap, therefore when $\|x_i - x_j\| \leq 2r$.

The probability of having n robots in a region $\pi(2r)^2$ follows a Poisson process for which:

$${}_jP_n(k, A) = \frac{(\lambda_j[k]A)^n e_{-\lambda_j[k]A}}{n!}$$

and therefore, the probability of having a collision in cell j will be equal to

$$\phi_j[k] = 1 -_j P_0(k, A) \tag{6.2}$$

where

$$\begin{aligned} {}_jP_0(k, A) &= e_{-\lambda_j[k]A} \\ A &= 2\pi r \\ \lambda_j[k] = \mu_j[k]\frac{N}{D_j}. & \end{aligned}$$

Since the swarm is composed by homogeneous robots, we drop the index i and we consider a general agent. We can associate a binary variable for the collision a time k as:

$$\gamma[k] = \begin{cases} 1 & \text{if a collision is experienced at time } k \\ 0 & \text{otherwise.} \end{cases} \tag{6.3}$$

The problem can then be expressed as follow:

Problem 2 *Given a sequence of measure* $\Gamma[k] = \{\gamma[k], \ldots, \gamma[1]\}$*, find what cell the robot occupies at time* k*.*

In the next section we investigate a solution for this problem.

6.3 Hidden Markov Model for Collision Based Localization

In this section we describe the mathematical model used by the robot to solve Problem 1. Due to the lack of locational information and detailed sensing, as well as the constant interaction between robots, the motion of the robots between cells is stochastic. A Markov model will then be used to model the probability of transition between cells. We denote by $q[k]$ the cell occupied by the robot at time k, i.e. if the robot occupies cell D_j at time k, then $q[k] = j$. The probability of moving from cell j to cell i will be equal to:

$$P_{ij} = Prob(q(k+1) = i \quad | \quad q[k] = j), \quad i, j \in \{1, \ldots, M\} \tag{6.4}$$

In the following section we will assume a particular choice for the set of probability transition but for know we assume these values to be known.

Given the mean-field approximation means, the fractions of robots inside a given cell satisfy the same Markovian properties, i.e., equation (6.4) implies that:

$$\mu[k+1] = P\mu[k]. \tag{6.5}$$

It is worth to note that the states of the Markov chain are hidden, since the robots do not know which cell they are currently occupying. Since robot make observations

in the form of collision measurements over the set $\{\Gamma_1, \Gamma_1\} = \{0, 1\}$, the observation probabilities thus becomes:

$$G_{lj}[k] = Prob(\gamma[k] = \Gamma_l \,|\, q[k] = j), \quad l = 1, 2, \quad j \in \{1, \ldots, M\}. \tag{6.6}$$

Since the robots are homogeneous, the probability of observing a collision in cell D_j at time k is $\phi_j[k]$ can be expressed as:

$$G_{1j}[k] = 1 - \phi_j[k] \qquad G_{2j}[k] = \phi_j[k], \quad j \in \{1, \ldots, M\}. \tag{6.7}$$

Here, the unusual dependency on time for the observation probabilities, is due to the fact that even though we are focusing our attention on a single robot, all the other robots are moving around at the same time; the statistics of this motion ultimately determines the collision probabilities.

Let us assume $\delta_k(i)$ as the probability of being in cell D_i at time k, given the sequence of observations $\Gamma[k] = \{\gamma[k], \ldots, \gamma[1]\}$. This is given by the posterior probability in the context of the hidden Markov model $\mathcal{H}$,

$$\delta_k(i) = Prob(q[k] = 1 \,|\, \Gamma[k], \mathcal{H}). \tag{6.8}$$

A possible choice, is to estimate the current cell of the robot by simply picking the state corresponding to the highest posterior probability, thus:

$$q^*[k] = \underset{j \in \{1, \ldots, M\}}{\operatorname{argmax}} \, \delta_k(i). \tag{6.9}$$

The estimator in (6.9) is commonly referred to as pointwise maximum a-posteriori probability (PMAP) estimator [61]. The optimal accuracy of this estimator is given by the fact that it maximizes the expected number of correct estimates. In order to compute $\delta_k(i)$ for each time instant k, we define a variable $\alpha_k(i)$ which denotes the

joint probability of obtaining a sequence of observations $\Gamma[k]$ while being in state D_j at time k,

$$\alpha_k(i) = Prob(\Gamma[k],\, q[k] = i \,|\, \mathcal{H}).$$

Thus, the probability of obtaining a given sequence of observations is now simply the sum of $\alpha_k(i)$ over all the cells of the domain:

$$Prob(\Gamma[k] \,|\, \mathcal{H}) = \sum_{j=1}^{M} \alpha_k(j).$$

Lastly, it is possible to rewrite $\delta_k(i)$ as:

$$\delta_k(i) = \frac{Prob(\Gamma[k], q[k] = i \,|\, \mathcal{H})}{Prob(\Gamma[k]|\mathcal{H})},$$

or, more conveniently as:

$$\delta_k(i) = \frac{\alpha_k(i)}{\sum_{j=0}^{M} \alpha_k(j)}.$$

It is now possible to use the recursive forward algorithm [61] to compute values of $\delta_k(i)$ over time.

6.3.1 Simulation Results

Let us now consider a scenario where n robots are moving around a circular track comprised of M segments, each of width w_i, for $i \in \{1, \ldots, M\}$ (Figure 6.2). The transition Markov matrix P is chosen in such a way that at each time step k, the robot can either remain in the cell they currently occupy or move to the next cell. Forcing transitions between cells to occur in a single direction (clockwise or counterclockwise), the width of each segment w_i not only affects the collision probabilities but also the congestion in each cell.

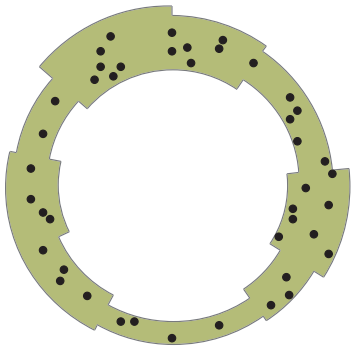


Figure 6.2: Example of circular track with cell of varying length.

We chose to model this congestion by letting the probability of a robot staying in the same cell be inversely proportional to the width of the cell, which correspond to the assumption that the narrower the segment, the lower the probability of changing cell:

$$P(i,i) = \epsilon\tfrac{1}{w_i}, \qquad P(i+1,i) = 1 - P(i,i), \quad i = \{1,\ldots,M-1\}$$
$$P(M,M) = \epsilon\tfrac{1}{w_M}, \qquad P(1,M) = 1 - P(M,M),$$

where ϵ is chosen so as to forcing the probabilities between 0 and 1. In the circular track with 10 individual cells of widths $[0.2, 0.17, 0.16, 0.11, 0.09, 0.13, 0.15, 0.24, 0.26, 0.28]$ and $\epsilon = 0.07$ (Figure 6.2). The simulations were conducted for 300 iterations and assuming 45 agents. Figure 6.3 shows the real cell occupied by the agent and its estimates. As can be seen, using only collision measurements, the PMAP estimator generates estimates which track the real cell occupancy as the robot moves around the domain.

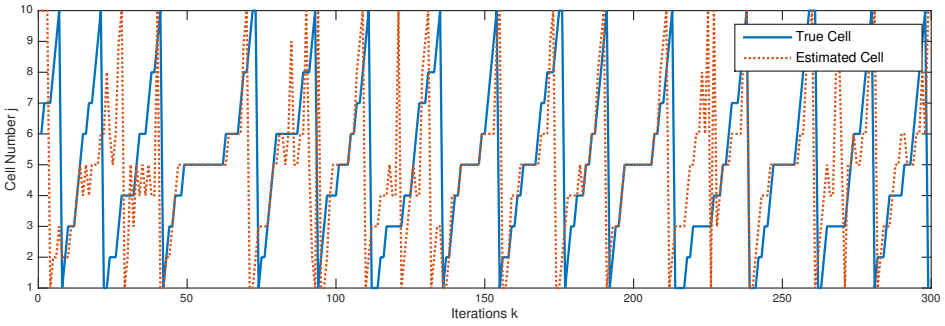


Figure 6.3: Numerical results for a ten cells circular track.

CHAPTER 7

Conclusion and Future Work

In this chapter we summarize the results presented in this work to draw possible trajectories for future research. As described in Chapter 2, a large body of research has been developed with the intent to describe, design and control multiagent systems. In this work we focused our attention on control techniques based on the mutual distance between agents of a group. Besides being rapidly available in many physical systems, relative distance measurements can also represent an abstraction in more general scenarios. Examples of this sort are the distance between two ideas in a group of people or the distance between two thermodynamics states.

Using a motivating example, we noticed that many behaviors in autonomous systems take into account the distance from a set of visible neighbors. For example, a behavior of this form is obtained when relative distance measurements are considered for a collision avoidance protocol (Chapter 3). Similarly, when cohesion is pursued in large swarms of autonomous agents, the single controllers must be based on relative distance/position from the nearest neighbors (Chapter 4).

As mentioned earlier, research on low-complexity solutions has also motivated the definition of appropriate architectures and algorithms when minimal resources and information are available. Distributed techniques for control of swarms possess

inviting properties such as high resilience and flexibility in adapting to changes/tasks. Much remains to be investigated on how to guarantee these properties with limited complexity and in a distributed fashion. In this work we moved a step in this direction. Self-assembly behaviors can be used for efficient coordination and the state of the swarm can be changed even when communications are allowed for few individuals in the swarm.

We finally showed, in the last chapter, that proximity measurements can also be used as sources of information. In particular, a string of binary variables representing encounters with other agents at discrete time, was used to localize an individual in densely populated environments.

7.1 Future Work on Trustworthiness in Collision Avoidance Algorithms

As showed in Chapter 3, exact knowledge on the state of an autonomous agent can introduce vulnerabilities. In particular, when predictable collision avoidance techniques are employed, these information can be exploited to drive the system's state towards predefined targets. This problem presents several possible extensions. As showed, the definition of closed form solutions seem to be intractable. However, game theory can represent a favorable environment for deriving useful results. Moreover, the definition of those regions where neighbors have the highest effect on the system's trajectory can be defined in terms of the reachability set of a predefined target.

In addition, computational results could be extended towards the definition of the whole approaching/forcing problem as a single unified optimization problem. By considering the binary variable corresponding to collision/collision-free state of the system as an additional design parameter, a single mixed integer MPC can be constructed. Also, assuming an evader's varying velocity and the problem in a three-dimensional domain would shorten the distance with real case applications.

Finally, in terms of mitigating countermeasures, several possible avenues can be considered. For example, the introduction of a stochastic component in the deconflicting maneuver seems to be worth investigating.

7.2 Future Work on Aggregation Behaviors and Control of Networked Systems

Control of a networked systems is a highly investigated subject. Given the intersection of this subject with an extraordinary variety of applications and disciplines, many results have been proved and many challenges are still open. A distance based aggregation process for autonomous mobile vehicles was proposed in Chapter 4. The protocol we described requires a minimal set of information on the final swarm configuration, namely a single fixed scalar. Future natural developments of this work include the ability of the swarm to adapt the desired inter-agent distance to changes in the environments or presence of faulty agents.

The injection of an input in the swarm by means of a leader was also considered. The leader was chosen at random and a reference signal directly injected in the system. To this end, the leader selection process can be integrated with the presence of a duty

cycle in the communication with the external world. The high power consumption of communication systems discourage the presence of a continuous link with the external world; for this reason, we can envision the ability for each agent to decide when to switch to leader state. This decision should take into consideration the position of the agent within the networks and its efficacy as leader. An estimation of the agent's *centrality* could serve the purpose.

The great number of opportunities in autonomous mobile robotics and autonomous systems continues to attract the interest of many communities. Contribution on the subject are by no means limited to engineering communities but extend to computer scientist, physicists, economists and sociologist. Future developments of this disciplines will certainly be fostered by the fusion and collaboration of these communities.

APPENDIX

First Order Taylor Approximation

In this section we derive the expression for the first order Taylor approximation relative to the system dynamics and the constraints introduced by the collision avoidance. Considering first the two unicycle dynamics contained in (3.27) we have:

$$f(x_0,u_0)=\begin{bmatrix} v_e\cos\phi_e \\ v_e\sin\phi_e \\ v_p\cos\phi_p \\ v_p\sin\phi_p \\ \omega_p \end{bmatrix} \quad f_{x|x_0}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -v_p\sin\phi_p \\ 0 & 0 & 0 & 0 & v_p\cos\phi_p \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
$$f_{u|u_0}=\begin{bmatrix} 0 & -v_e\sin\phi_e & 0 & 0 & 0 \\ 0 & v_e\cos\phi_e & 0 & 0 & 0 \\ 0 & 0 & \cos\phi_p & 0 & 0 \\ 0 & 0 & \sin\phi_p & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.1}$$

The two constraints contained in 3.27 are reported and expanded here for convenience:

$$g_1^{k+1} : \tan\lambda^{k+1}\left(v_e\cos\phi_e^{k+1} - v_p^{k+1}\cos\phi_e^{k+1}\right) - v_e\sin\phi_e^{k+1} + v_p^{k+1}\sin\phi_p^{k+1} = 0 \quad (7.2)$$

$$g_2^{k+1} : \lambda^{k+1} - \arctan\left(\frac{y_p^{k+1} - y_e^{k+1}}{x_p^{k+1} - x_e^{k+1}}\right) - \arcsin\left(\frac{R_{pz}}{\|q_p^{k+1} - q_e^{k+1}\|_2}\right) = 0 \quad (7.3)$$

Equations (7.2) and (7.3) are now derived respect to their variable, namely λ, v_p, ϕ_p and ϕ_e. In order to reduce burden from notation, we consider the constraint (7.3) only for one angle λ_1. Similar derivation can be extended for the constraint relative to λ_2. Considering first constraint (7.2), we can easily derive:

$$g_{1,\lambda} = \sec^2\lambda(v_e\cos\phi_e - v_p\cos\phi_p)$$

$$g_{1,v_p} = -\tan\lambda\cos\phi_p + \sin\phi_p$$

$$g_{1,\phi_p} = v_p\tan\lambda\sin\phi_p + v_p\cos\phi_p$$

$$g_{1,\phi_e} = -v_e\tan\lambda\sin\phi_e - \cos\phi_e$$

Considering the constraint (7.3), we first need to extract the dependency from the control inputs. In order to do so, we introduce the discrete dynamics of the system into the constraints, that is:

$$0 = \lambda^{k+1} - \arctan\left(\frac{\delta_y^{k+1}}{\delta_x^{k+1}}\right) - \arcsin\left(\frac{R_{pz}}{\sqrt{(\delta_x^{k+1})^2 + (\delta_y^{k+1})^2}}\right) \quad (7.4)$$

where:

$$\delta_x^{k+1} = x_p^k + v_p^k\cos\phi_p^k - x_e^k - v_e\cos\phi_e^k\Delta t$$

$$\delta_y^{k+1} = y_p^k + v_p^k\sin\phi_p^k - y_e^k - v_e\sin\phi_e^k\Delta t$$

Computing the derivatives of g_2 respect with its arguments leads to:

$$
\begin{aligned}
g_{2,\lambda} &= 1 \\
g_{2,v_p} &= -\frac{1}{1+\frac{\delta_y^2}{\delta_x^2}}\left(\frac{\sin\phi_p\Delta t\,\delta_x - \cos\phi_p\Delta t\,\delta_y}{\delta_x^2}\right) - \\
&- \frac{1}{\sqrt{1-\xi^2}}\left(-\frac{R_{pz}\,K_1}{\delta_x^2+\delta_y^2}\right)
\end{aligned}
$$

$$
\begin{aligned}
g_{2,\phi_p} &= -\frac{1}{1+\frac{\delta_y^2}{\delta_x^2}}\left(\frac{v_p\cos\phi_p\Delta t\,\delta_x + v_p\sin\phi_p\Delta t\,\delta_y}{\delta_x^2}\right) - \\
&- \frac{1}{\sqrt{1-\xi^2}}\left(-\frac{R_{pz}\,K_2}{\delta_x^2+\delta_y^2}\right) \\
g_{2,\phi_e} &= -\frac{1}{1+\frac{\delta_y^2}{\delta_x^2}}\left(\frac{-v_e\cos\phi_e\Delta t\,\delta_x - v_e\sin\phi_e\Delta t\,\delta_y}{\delta_x^2}\right) - \\
&- \frac{1}{\sqrt{1-\xi^2}}\left(-\frac{R_{pz}\,K_3}{\delta_x^2+\delta_y^2}\right)
\end{aligned}
$$

where we defined the following quantities:

$$
\begin{aligned}
\xi &= \frac{R_{pz}}{\sqrt{\delta_x^2+\delta_y^2}} \\
K_1 &= \frac{\partial\sqrt{\delta_x^2+\delta_y^2}}{\partial v_p} = \frac{1}{2}\left(\delta_x^2+\delta_y^2\right)^{-\frac{1}{2}}\left(2\delta_x\cos\phi_p\Delta t + 2\delta_y\sin\phi_p\Delta t\right) \\
K_2 &= \frac{\partial\sqrt{\delta_x^2+\delta_y^2}}{\partial\phi_p} = \frac{1}{2}\left(\delta_x^2+\delta_y^2\right)^{-\frac{1}{2}}\left(-2\delta_x v_p\sin\phi_p\Delta t + 2\delta_y v_p\cos\phi_p\Delta t\right) \\
K_3 &= \frac{\partial\sqrt{\delta_x^2+\delta_y^2}}{\partial\phi_e} = \frac{1}{2}\left(\delta_x^2+\delta_y^2\right)^{-\frac{1}{2}}\left(2\delta_x v_e\sin\phi_p\Delta t - 2\delta_y v_e\cos\phi_p\Delta t\right)
\end{aligned}
$$

The quantities just computed can be written in the compact form:

$$
E = \begin{bmatrix} g_{1,\lambda} & g_{1,\phi_e} & g_{1,v_p} & 0 \\ g_{2,\lambda} & g_{2,\phi_e} & g_{2,v_p} & 0 \end{bmatrix} \quad F = \begin{bmatrix} 0 & 0 & 0 & 0 & g_{1,\phi_p} \\ 0 & 0 & 0 & 0 & g_{2,\phi_p} \end{bmatrix} \tag{7.5}
$$

Bibliography

[1] L. E. Parker, "Current state of the art in distributed autonomous mobile robotics," in *Distributed Autonomous Robotic Systems 4*. Springer, 2000, pp. 3–12.

[2] R. M. Murray, "Recent research in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 571–583, 2007.

[3] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[4] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2013.

[5] A. Mogilner and L. Edelstein-Keshet, "A non-local model for a swarm," *Journal of Mathematical Biology*, vol. 38, no. 6, pp. 534–570, 1999.

[6] T. Vicsek, A. Czirók, 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, no. 6, p. 1226, 1995.

[7] A. S. Mikhailov and D. H. Zanette, "Noise-induced breakdown of coherent collective motion in swarms," *Physical Review E*, vol. 60, no. 4, p. 4571, 1999.

[8] E. Shlizerman, J. Phillips-Portillo, D. B. Forger, and S. M. Reppert, "Neural integration underlying a time-compensated sun compass in the migratory monarch butterfly," *Cell Reports*, 2016.

[9] S. C. Pratt, "Quorum sensing by encounter rates in the ant temnothorax albipennis," *Behavioral Ecology*, vol. 16, no. 2, pp. 488–496, 2005.

[10] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.

[11] B. Akmee and D. S. Bayard, "Markov chain approach to probabilistic guidance for swarms of autonomous agents," *Asian Journal of Control*, vol. 17, no. 4, pp. 1105–1124, 2015.

[12] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic swarm guidance using inhomogeneous markov chains," *arXiv preprint arXiv:1403.4134*, 2014.

[13] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[14] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*. IEEE, 2005, pp. 6698–6703.

[15] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401–420, 2006.

[16] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[17] D. B. Kingston, W. Ren, and R. W. Beard, "Consensus algorithms are input-to-state stable," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 1686–1690.

[18] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 495–500, 2002.

[19] H. G. Tanner and A. Kumar, "Towards decentralization of multi-robot navigation functions," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 4132–4137.

[20] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic control*, vol. 52, no. 5, pp. 863–868, 2007.

[21] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[22] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 501–518, 1992.

[23] P. Panyakeow and M. Mesbahi, "Deconfliction algorithms for a pair of constant speed unmanned aerial vehicles," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 50, no. 1, pp. 456–476, 2014.

[24] S. Givigi Jr and H. Schwartz, "A game theoretic approach to swarm robotics," *Applied Bionics and Biomechanics*, vol. 3, no. 3, pp. 131–142, 2006.

[25] P. Park and C. Tomlin, "Performance evaluation and optimization of communication infrastructure for the next generation air transportation system," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 4, pp. 1106–1116, 2015.

[26] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ads-b implementation in the next generation air transportation system," *International Journal of Critical Infrastructure Protection*, vol. 4, no. 2, pp. 78–87, 2011.

[27] J. V. Carroll, "Vulnerability assessment of the u.s. transportation infrastructure that relies on the global positioning system," *The Journal of Navigation*, vol. 56, 2003.

[28] T. Humphreys, "Statement on the vulnerability of civil unmanned aerial vehicles and other systems to civil gps spoofing," *University of Texas at Austin (July 18, 2012)*, 2012.

[29] C. J. Giannatto and G. Markowsky, "Potential vulnerabilities of the nextgen air traffic control system," in *World Congress in Computer Science, Computer Engineering and Applied Computing*, Las Vegas, 2014.

[30] P. Park, H. Khadilkar, H. Balakrishnan, and C. J. Tomlin, "High confidence networked control for next generation air transportation systems," *Automatic Control, IEEE Transactions on*, vol. 59, no. 12, pp. 3357–3372, 2014.

[31] N. Ghose and L. Lazos, "Verifying ads-b navigation information through doppler shift measurements," in *Digital Avionics Systems Conference (DASC), 2015 IEEE/AIAA 34th.* IEEE, 2015, pp. 4A2–1.

[32] M. Monteiro, A. Barreto, T. Kacem, J. Carvalho, D. Wijesekera, and P. Costa, "Detecting malicious ads-b broadcasts using wide area multilateration," in *Digital Avionics Systems Conference (DASC), 2015 IEEE/AIAA 34th.* IEEE, 2015, pp. 4A3–1.

[33] P. Pierpaoli, M. Egerstedt, and A. Rahmani, "Altering UAV flight path by threatening collision," in *Proc. Digital Avionics Systems Conference*, Prague, Czech Republic, Sep. 2015.

[34] S. M. LaValle, *Planning algorithms.* Cambridge university press, 2006.

[35] B. Albaker and N. Rahim, "A survey of collision avoidance approaches for unmanned aerial vehicles," in *technical postgraduates (TECHPOS), 2009 international conference for*. IEEE, 2009, pp. 1–7.

[36] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[37] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 28, no. 5, pp. 562–574, 1998.

[38] E. Lalish, K. Morgansen *et al.*, "Decentralized reactive collision avoidance for multivehicle systems," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 1218–1224.

[39] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[40] M. J. Osborne and A. Rubinstein, *A course in game theory*. MIT press, 1994.

[41] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[42] S.-Y. Tu and A. H. Sayed, "Tracking behavior of mobile adaptive networks," in *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*. IEEE, 2010, pp. 698–702.

[43] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[44] O. Rozenheck, S. Zhao, and D. Zelazo, "A proportional-integral controller for distance-based formation tracking," in *Control Conference (ECC), 2015 European*. IEEE, 2015, pp. 1693–1698.

[45] D. Zelazo, A. Franchi, F. Allgöwer, H. H. Bülthoff, and P. R. Giordano, "Rigidity maintenance control for multi-robot systems," in *Robotics: Science and Systems*, 2012, pp. 473–480.

[46] D. Zelazo, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Decentralized rigidity maintenance control with range measurements for multi-robot systems," *The International Journal of Robotics Research*, vol. 34, no. 1, pp. 105–128, 2015.

[47] L. Krick, M. E. Broucke, and B. A. Francis, "Stabilisation of infinitesimally rigid formations of multi-robot networks," *International Journal of control*, vol. 82, no. 3, pp. 423–439, 2009.

[48] S.-J. Chung and J.-J. E. Slotine, "Cooperative robot control and concurrent synchronization of lagrangian systems," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 686–700, 2009.

[49] P. Barooah and J. P. Hespanha, "Graph effective resistance and distributed control: Spectral properties and applications," in *Decision and control, 2006 45th IEEE conference on*. IEEE, 2006, pp. 3479–3485.

[50] H. K. Khalil, *Noninear Systems*. Prentice-Hall, New Jersey, 1996.

[51] B. Hendrickson, "Conditions for unique graph realizations," *SIAM journal on computing*, vol. 21, no. 1, pp. 65–84, 1992.

[52] H. G. Tanner, "On the controllability of nearest neighbor interconnections," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 3. IEEE, 2004, pp. 2467–2472.

[53] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.

[54] S. Martini, M. Egerstedt, and A. Bicchi, "Controllability decompositions of networked systems through quotient graphs," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 5244–5249.

[55] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[56] A. Y. Yazicioğlu and M. Egerstedt, "Leader selection and network assembly for controllability of leader-follower networks," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 3802–3807.

[57] B. Briegel, D. Zelazo, M. Bürger, and F. Allgöwer, "On the zeros of consensus networks," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 1890–1895.

[58] D. Zelazo and M. Mesbahi, "Edge agreement: Graph-theoretic performance bounds and passivity analysis," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 544–555, 2011.

[59] K. Fitch and N. E. Leonard, "Optimal leader selection for controllability and robustness in multi-agent networks," in *Control Conference (ECC), 2016 European*. IEEE, 2016, pp. 1550–1555.

[60] F. Lin, M. Fardad, and M. R. Jovanovic, "Algorithms for leader selection in stochastically forced consensus networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1789–1802, 2014.

[61] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.