Survey of search-and-secure algorithms for surveillance UGVs

MAGNUS LINDHÈ

FOI-R--2267--SE Scientific report
ISSN 1650-1942 April 2007

Defence and Security, Systems and Technology

FOI, Swedish Defence Research Agency, is a mainly assignment-funded agency under the Ministry of Defence. The core activities are research, method and technology development, as well as studies conducted in the interests of Swedish defence and the safety and security of society. The organisation employs approximately 1250 personnel of whom about 900 are scientists. This makes FOI Sweden's largest research institute. FOI gives its customers access to leading-edge expertise in a large number of fields such as security policy studies, defence and security related analyses, the assessment of various types of threat, systems for control and management of crises, protection against and management of hazardous substances, IT security and the potential offered by new sensors.

FOI
Defence Research Agency
Defence and Security, Systems and Technology
SE-164 90 Stockholm

Phone: +46 8 555 030 00
Fax: +46 8 555 031 00

www.foi.se

Magnus Lindhé

Survey of search-and-secure algorithms for surveillance UGVs

FOI-R--2267--SE Scientific report
ISSN-1650-1942 April 2007

Defence and Security, Systems and Technology

Issuing organisation FOI – Swedish Defence Research Agency Defence and Security, Systems and Technology SE-164 90 STOCKHOLM	**Report number, ISRN** FOI-R--2267--SE	**Report type** Scientific report
	Research area code Strike and Protection	
	Month year April 2007	**Project no.** E28004
	Sub area code Ground Vehicles	
	Sub area code 2	
Author/s (editor/s) Magnus Lindhé	**Project manager** Petter Ögren	
	Approved by Helena Bergman	
	Sponsoring agency FMV, 297316-LB704859	
	Scientifically and technically responsible Karl H. Johansson	
Report title Survey of search-and-secure algorithms for surveillance UGVs		

Abstract

This survey investigates four different areas, all of which are relevant in the context of searching and securing an area using UGVs. The first area is that of coordinating the searchers, so that a region of interest is completely searched and an intruder cannot sneak into sectors that are already covered. The second is flocking and formation control, i.e., how to design control laws for each agent so that the group as a whole converges to some desired relative positions and can be moved as a unit. This is useful for transporting a group of UGVs to and from the search region. The third area is how to coordinate agents performing a task, so that they maintain inter-agent communication. This is a requirement for fully benefitting from using multiple robots. Finally, we also study how to move agents such that they maintain a line of sight between them, which is useful for detection.

The conclusion is that there are complete algorithms for searching an area that has no obstacles, but more realistic environments are less studied. For flocking and formation control there are many useful algorithms and perhaps the most important design choice is to select between those using local communication and those that are centralized. For maintaining inter-agent communications, the most robust methods employ dedicated relaying agents and/or protocols that are disruption-tolerant. For maintaining a line of sight, the most reliable methods use hard geometric constraints for what points are feasible to go to.

Keywords

Autonomy, Cooperative, Surveillance, Path Planning, Task Assignment, Search, Coverage, Obstacle Avoidance, UGV

Further bibliographic information	**Language** English
ISSN ISSN-1650-1942	**Pages** 62 p.
	Price According to price list

Utgivare	**Rapportnummer, ISRN**	**Klassificering**
FOI – Totalförsvarets forskningsinstitut Försvars- och säkerhetssystem 164 90 STOCKHOLM	FOI-R--2267--SE	Vetenskaplig rapport
	Forskningsområde Bekämpning och skydd	
	Månad år April 2007	**Projektnummer** E28004
	Delområde Fordonsteknik	
	Delområde 2	
Författare/redaktör Magnus Lindhé	**Projektledare** Petter Ögren	
	Godkänd av Helena Bergman	
	Uppdragsgivare/kundbeteckning FMV, 297316-LB704859	
	Tekniskt och/eller vetenskapligt ansvarig Karl H. Johansson	

Rapportens titel
Litteraturstudie rörande algoritmer för sökning med markgående övervakningsfarkoster

Sammanfattning

Denna litteraturstudie berör fyra olika frågeställningar, som alla är relevanta för att söka av och säkra ett område med hjälp av obemannade markfarkoster, UGV:er. Den första frågeställningen gäller hur man ska koordinera sökarna för att garantera täckning, samtidigt som en inkräktare inte ska kunna ta sig in i redan avsökta sektorer. Den andra gäller flock- och formationsstyrning, d.v.s. hur man styr enskilda agenter så att gruppen som helhet konvergerar mot önskade relativa positioner och kan förflyttas som en enhet. Det är användbart för att förflytta en grupp UGV:er till och från ett sökområde. Den tredje frågeställningen är hur man kan koordinera agenter så att de bibehåller kontakten med varandra medan de utför en uppgift. Det är nödvändigt för att dra nytta av fördelarna med att ha flera robotar. Slutligen studerar vi också hur agenternas rörelse kan anpassas så att en siktlinje mellan dem bibehålls, vilket är användbart för detektion.
Slutsatsen är att det finns fullständiga algoritmer för att söka av ytor som saknar hinder, men mer realistiska miljöer är mindre studerade. För flock- och formationsstyrning finns många användbara algoritmer och det kanske viktigaste valet för en konstruktör står mellan sådana som kommunicerar lokalt och centraliserade metoder. För att bibehålla kommunikation mellan agenter använder de mest robusta metoderna dedicerade reläagenter och/eller avbrottståliga protokoll. För att behålla en siktlinje använder de pålitligaste metoderna hårda geometriska bivillkor för att avgöra vilka punkter som är tillåtna att gå till.

Nyckelord
autonomi, samverkan, övervakning, banplanering, uppgiftstilldelning, spaning, hinderundvikande, UGV

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN ISSN-1650-1942	**Antal sidor:** 62 s.
Distribution enligt missiv	**Pris:** Enligt prislista

Contents

1 Introduction and Definition of Scope

In recent years, unmanned ground vehicles (UGVs) have emerged as useful tactical tools, improving efficiency and personnel safety in military operations. They are mainly used for handling explosive devices and for short-range reconnaissance in dangerous environments. Most existing platforms today are operated by a human through remote control. This requires a dedicated operator, and thus extra personnel, and also poses some limitations on the use, since the UGV has to stay within communications range of the operator. This could be alleviated by adding more autonomy to the UGVs, so that they can be given instructions on a higher abstraction level, such as "search area A for intruders and then report back". One possible use for such autonomous UGVs is to assist human sentries in surveillance.

Motivated by this, the Defence Research Agency has initiated a project to investigate the possible use of UGVs for surveillance and security. The Royal Institute of Technology, KTH, is a partner in this project and will investigate the subject of using UGVs for searching for mobile intruders and securing an area. As a first phase, this report surveys the state of the art on the subject, to lay the ground for future research and joint demonstrations with the other project partners.

In this chapter, we give an example of a motivating scenario for this kind of applications. We then define the scope of the survey and end by giving an outline of the report.

1.1 Motivating Scenario: Multiple UGVs Assisting a Sentry

Due to outbreaks of armed conflicts in the surrounding region, an African country has been overwhelmed with refugees, creating chaos in already overcrowded refugee camps. UNHCR decides to establish an air bridge, supplying food, medicine and shelters. On request, the Swedish Air Force sends an airport unit to set up a port of entry on a nearby, previously abandoned, airstrip. The already hard working unit is also detailed with guarding the maintenance depot, containing spare parts, weapons and fuel.

One night, the perimeter alarm signals that there is movement on the inside of the barbed fence. To find out what has happened, a sentry is dispatched, taking with her five UGVs from the charging station by the central guard tower. She walks to the gate of the storage area, and the UGVs form a group that automatically follows.

After the sentry has let the robots in through the gate, they split up and search the area, employing infrared sensors that react to the heat signature of a human body. Using a stored map of the houses, the UGVs plan their movements so that no one can sneak past the searchers without being detected.

A few minutes after the alarm, one of the robots spots two persons, trying to break into a container. It stands back, alerts the central guard and sends a TV image of the intruders. The quick reaction force is deployed and apprehends the intruders. Images from the UGV are later used as evidence in the trial

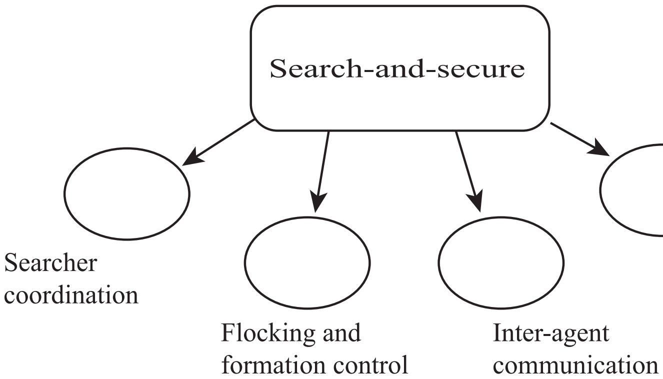

Figure 1.1: The connection between searching and securing an area using UGVs, and four research areas that are motivated in this context.

against the thieves.

1.2 Limitations of Scope and Subproblems

The above search scenario is an example application of what we have chosen to call *search and secure*. This is a broad subject that comprises many research areas, four of which we will study in this survey. They are schematically depicted in Figure 1.1.

The first area concerns coordination of the searchers, to ensure that all parts of the search area are covered and that an intruder does not get the opportunity to get into a region that has already been searched. We call this *searcher coordination*, and we will define the scope of this area in Chapter 2.

As described in the scenario, a useful ability for this kind of multi-agent systems is to move as a group, without the operator having to control each agent. Among other uses, it facilitates transporting the whole group to and from the area that is to be searched. Creating such group behavior is often called *flocking and formation control* and is defined in more depth in Chapter 3.

To get the full advantages of using several robots, it is essential that they can communicate. Performing a task under the constraint that *inter-agent communication* must not be lost is the third research area that we will survey, and it is described in Chapter 4.

Finally we will also consider the related area of performing a task while *maintaining a line of sight* between selected agents. As shown in Chapter 5, this is useful both for searching and for allowing the agents to support each other's navigation.

1.3 Outline of the Report

In the following four chapters, we present a survey of each of the four areas defined above. Each chapter ends with a short summary, focused on conclusions relevant for the continuation of the project. In Chapter 3, we also list some simulation software packages for multi-agent robotics. The survey is concluded in Chapter 6 and the report ends with a collection of short reviews of papers in each of the four areas.

2 Searcher Coordination

2.1 Definition

In this survey, we will use the following broad definition of searcher coordination:

Find a strategy for movement, communication and sensor usage so that a minimal number of agents can search a limited area containing obstacles. The objective is to detect all intruders in the area, assuming that they try to evade and have full knowledge of the positions of the searchers.

We ask for a strategy in a wide sense: telling how to position each agent, how to use the sensors and how and with whom to communicate.

Since this survey is not focused on sensor technology or signal processing, we will not consider the (generally difficult) problem of actually processing the data from a sensor to discern an intruder from the environment or fellow searchers. If an intruder comes within the field of view of the sensor of a searcher, the task is considered solved.

2.2 Survey

Related to searcher coordination is the more formal problem of *pursuit-evasion.* It was was first introduced by Parsons [42] as a discrete problem on a graph, where searchers and a single evader could move freely between connected nodes in the graph. The evader was considered captured if it was positioned at the same node as a searcher. The question was: how many searchers are needed to guarantee capturing the evader, regardless of how it behaves? Informally, the problem was described as "Suppose a man is lost and wandering unpredictably in a dark cave. A party of searchers who know the structure of the cave is to be sent to find him. What is the minimum number of searchers needed to find the lost man regardless of how he behaves?" [42]

A continuous version of the problem was proposed by Suzuki *et al.* [52]. They considered a single searcher and evader on a simple polygon (i.e. a polygon with no holes). The searcher is modelled as a k-searcher, whose visibility is limited to k thin rays extending radially from the position of the searcher. Each ray can be turned with bounded angular velocity, and the evader is considered captured if hit by a ray. The question considered was if a given polygon can be searched by a k-searcher in such a way that the evader is guaranteed to be captured. Such a polygon is called *k-searchable*, and the paper presents necessary and sufficient conditions for a polygon to be 1- and 2-searchable, respectively. The authors also introduce the notation of an ∞-searcher, which is a searcher with an omnidirectional sensor.

Later, Lee *et al.* [32] presented search schedules for 1- and 2-searchable simple polygons. Computing the schedules was shown to have complexity $O(n^2)$, where n is the number of vertices of the polygon (and hence also the number of sides). In an earlier paper [40], the same authors also show that any polygon that is searchable by an ∞-searcher is also searchable by a 2-searcher.

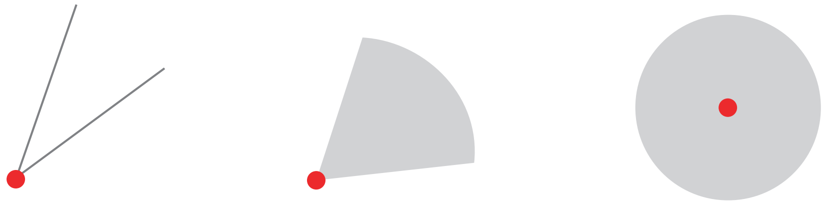

Figure 2.1: Three types of searchers: 2-searcher (a), $\frac{\pi}{3}$-searcher (b) and 2π-searcher (c).

Gerkey *et al.* [20] introduced a new kind of searcher, the ϕ-searcher. It models the capability of a robot with a camera, having a field of view with the angle ϕ. In this terminology, a searcher with an omnidirectional sensor is called a 2π-searcher. The different kind of searchers introduced so far are illustrated in Figure 2.1. In the paper, it is shown that computing the minimal number of searchers needed to search a simple polygon is NP-hard. They also give a complete schedule for searching a polygon with a single ϕ-searcher in the case when this is possible.

A way to increase the set of simple polygons that are searchable by a single 2π-searcher is to use the randomized strategy suggested by Isler *et al.* [27]. They partition the polygon into triangles, which means that if the searcher is located inside a triangle, all other points in the triangle are in its field of view. They then represent the polygon as a graph, where each node is a triangle and nodes are connected if it is possible to move directly between the corresponding triangles. A key observation is then that such a graph will always have a tree structure (since the polygon is simple). In such a graph, there is no way for an evader to escape if the searcher makes a run from the root towards a leaf and the evader cannot predict which leaf it will go to. So by making a suitable number of runs towards randomly selected leaves, the probability of capturing the evader can be made arbitrarily large. The drawback of this approach is that it will yield very time and energy consuming solutions that may not be very useful in more complex environments.

It is worth noting that this approach can also be used for a polygon with holes in it, by adding stationary searchers that position themselves so as to partition the corresponding graph into a tree. It can then be searched by a single mobile searcher. The paper gives a bound that $O(\sqrt{h}+1)$ searchers are needed for this, if h is the number of holes in the polygon.

The case of using several searchers to increase the set of searchable polygons is studied by Efrat *et al.* [13]. They consider chains of searchers, where the end searchers of the chain follow the boundary of the polygon. Adjacent searchers in the chain maintain line of sight, which stops an evader from moving into the already cleared region. The paper gives an exact limit on the number r^* of agents needed to sweep a given simple polygon. It also gives an algorithm of complexity $O(r^*n^3)$, with n being the number of vertices in the polygon, for computing a search schedule. Since both computing r^* and the schedule is computationally demanding, the authors also present an approximative algorithm of complexity $O(n^2)$, to compute a search schedule that requires at most r^*+2 agents.

This sensor model could be translated to 1-searchers, pointing their ray towards the next agent in the chain. That would also make one of the agents at the end of the chain unnecessary. If this is executed with 2- or 2π-searchers, only agents in the interior of the polygon would be needed.

Another approach is described by Hollinger *et al.* [24]. They suggest a probabilistic strategy for searching a polygon with obstacles using several robots, focusing on minimizing the expected time for finding the evader using 2π-searchers. The algorithm allows nice ways to account for sensors being imperfect and incorporating a priori probabilities for where the evader may be, but there is no upper bound on the time to find the evader. The algorithm also allows the evader to go back to already cleared regions, which then need to be cleared again. This could make the time to capture very long in case of active evasion.

Two similar problems deserve mentioning here: The first is the art gallery problem [51], where one wants to find the least number of stationary cameras, and their positions, needed so that at least one camera has a line of sight to every point in the polygon. The art gallery problem in a sense solves the search-and-secure problem if all cameras are replaced by stationary searchers. But not using the mobility of the searchers will in general mean using more searchers than necessary.

The other problem is that of coverage [8]. It is assumed that each agent has a given footprint, which could be defined by its sensor or payload. This footprint then has to pass over every point of a region. Applications for this include lawn-moving, cleaning and demining. Coverage algorithms do not in general solve the search-and-secure problem since the searcher needs to collide with the evader to sense it, so evasion is trivial.

2.3 Relation to the Project Objective

The 1-searcher and the ϕ-searcher seem to be the most practically interesting kinds of searchers, since they can be implemented by cameras or IR detectors.

For simple polygons (with no obstacles in the interior), there exist provably complete strategies to search using a minimal number of agents, either 1-searchers or ϕ-searchers. More realistic environments, with obstacles, are less studied and the results are not as complete. Either one can reduce the problem to the simple polygon case, by using stationary searchers at points where the corresponding graph forms a loop, or use non-complete probabilistic algorithms such as proposed by Hollinger *et al.*

3 Flocking and Formation Control

3.1 Definition

A disadvantage of using multi-agent systems is that the complexity of controlling the agents grows. In many applications it is therefore of interest to abstract the control so that the group *as a whole* is controlled, and each member follows automatically. In scenarios as described above, this can for example be used to transport a group of UGVs from one place to another, either in a completely autonomous fashion or under the supervision of an operator. We define this area as:

Find a control law for each agent in a group, such that the relative positions of the agents converge to some desired set, collisions are avoided and the group as a whole performs a desired motion.

Sometimes, the desired result can be defined as just all agents moving in the same direction, which is called *flocking*. Inspired by biology, a group performing flocking is sometimes called a *swarm*. On the other hand, if there are stricter requirements for the relative positions, such as wanting the agents to form a line, a circle or a lattice, this is called a *formation*.

Collision avoidance can be defined as no agents colliding, or also avoiding collisions with obstacles. The latter is usually called *obstacle avoidance*. Finally, the group movement can be encoded as a simple desired velocity of each agent or a more elaborate path-following or even autonomous navigation for the group as a whole.

3.2 Survey

To synchronize their movement, agents need to exchange information. The amount of information exchanged, whether communication needs to be synchronous, and the topology of communications can differ between algorithms. The topology describes which agent communicates with which, and also in what direction information is sent. We will roughly divide the algorithms we present into three categories, depending on the communication topology they employ. As illustrated in Figure 3.1, the first is *local communication*, where all agents communicate only with their neighbors. The set of neighbors can be fix or vary over time, e.g., depending on physical proximity. This also covers the case of a complete communication graph, where everyone communicates with everyone else. Next, we will consider the case of *centralized communication*, where all agents communicate directly with a supervisor. Finally, we consider the more specialized case where the communication graph is a *tree graph*. The root of the tree can be a physical agent or a virtual leader.

The choice of communication topology also affects the scalability of the formation. Algorithms that only require communication with neighbors scale well, since the demands on each agent are constant even if the group size increases. Further, Fax and Murray [16] have shown how communication topology affects the stability margins for convergence towards the prescribed formation.

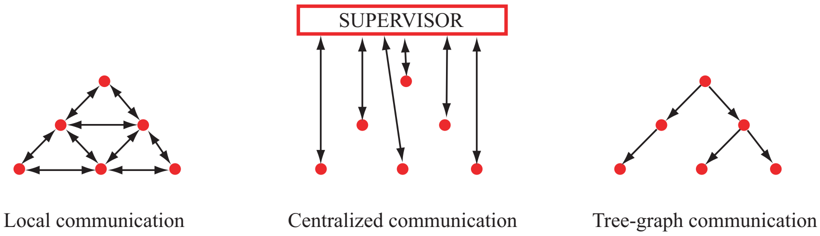

Figure 3.1: Three coarse categories of communication topologies for formation control. The arrows indicate information flow.

Two other design dimensions also deserve mentioning before we look at specific algorithms. The first is heterogeneousness. Does one agent have to be designated as leader and, in the case of formation control, do the agents have rigidly prescribed roles in the formation or can they choose any position? This affects the robustness of the group to failures of single agents, and whether several groups can easily merge.

The other dimension is agent dynamics. What kind of agent dynamics are allowed? The models can range from single or double integrators, via unicycles to car-like platforms or even more complex dynamics, especially in the case of aircraft.

Local communication One of the first papers on flocking came from the field of computer graphics and was written by Reynolds [49] in 1987. He introduced “boids”, a type of agents that could perform animal-like flocking and are still used in computer games and animated movies. The boids follow three rules:

- *Separation:* Avoid obstacles and other boids
- *Velocity alignment:* Adapt the velocity to that of the group
- *Flock cohesion:* Move towards the center of the group

The control output from each rule was combined based on priority, and used to control the acceleration of the boid. This is an example of communications between all agents, since each agent takes into account the position of all others.

More recently, Tanner *et al.* [55] have formulated the same rules, but based only on local information. They describe flocking for kinematic single integrator agents, whose control law consists of two terms. One aims to align the velocities of all agents, and the other is the sum of the negative gradients of an artificial inter-agent potential function. The potential function is a function of only the distance between two agents, and it has a unique minimum at a desired inter-agent spacing.

Two kinds of communication topologies are considered: First the communication graph is fixed, and then it is time-varying, by letting agents communicate with all neighbors within a given sensing radius. In the fixed case, the group will converge to an equilibrium where the sum of the potentials has a local minimum. A condition for this is that the communication graph is connected, i.e., there is a path between any two agents. If the communication graph is a tree graph, this local minimum will also be global, with the desired distance between all connected agents. In the time-switching case, convergence to an equilibrium can also be shown, if the communication graph remains connected at all times.

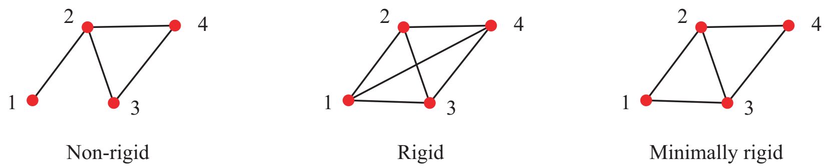

Figure 3.2: A formation with prescribed inter-agent distances indicated by lines. Depending on what distances are prescribed, the formation has different rigidity properties.

For the case of flocking, Jadbabaie *et al.* [28] have shown that it is a sufficient condition for convergence if the flock is connected at least once during every time interval of a fixed length T.

An interesting issue in the context of potential methods, is what neighbor distances need to be prescribed for the formation to be uniquely defined. As studied by Hendrickx *et al.* [23], a formation is *rigid* if maintaining the inter-agent distances specified by the communications graph is a sufficent condition for *all* inter-agent distances to be constant. If the graph contains the minimal necessary number of edges, the formation is said to be *minimally rigid*. This is illustrated in Figure 3.2, where the leftmost formation is not rigid, since the distance between agents 1 and 3 can vary.

Another class of methods is based on *behaviors*. A behavior is a controller that, based on sensor information, outputs a recommended control. Each behavior typically performs a simple function, and many behaviors are run in parallel. The agent then combines the outputs from each behavior into an aggregated control signal. This arbitration can be done through a weighted vector sum, by voting or by subsumption. When using vector summation, each behavior is weighted according to its importance, and safety behaviors such as avoiding obstacles typically get a high weight. In voting, all behaviors get a number of votes corresponding to their weight, and cast them on one or several discrete candidate controls. The candidate command with the highest number of votes is then executed. In a subsumption architecture, the control signal of the highest prioritized active behavior is used [2].

An application of behaviors to formation control is described by Balch and Arkin [5]. They suggest the following basic behaviors:

avoid-static-obstacle	Move away from obstacles
avoid-robot	Move away from other agents
move-to-goal	Move towards a waypoint for the group
noise	Move randomly, to dissolve deadlocks
maintain-formation	Approach the prescribed position in the formation

The three first behaviors are fairly intuitive, but the fourth is included to avoid getting stuck at saddle points of the vector fields surrounding obstacles. The fifth behavior steers the agent towards its position in the formation. This position can be relative to the leader, to the center of mass of the group or to a single designated neighbor. These alternatives also imply different communication topologies. The authors then propose arbitration by vector summation.

Yet another class of methods stem from the field of convex optimization. Raffard *et al.* [46] derive a method for controlling formations of aircraft, using dual decoupling. The formation problem is expressed as an optimization problem, where each aircraft has both individual objectives and formation objectives. Further, the dynamics of the aircraft are formulated as a set of constraints. By introducing slack variables and forming the dual problem, the solution can be found in a decentralized iterative manner, with only local communication.

As a final example, there are also geometric methods that were applied to the problem of positioning mobile sensors to maximize sensor coverage. Cortés *et al.* [10] formulate the problem of dispersing sensors in a convex polygonal environment so that the probability of sensing an event is maximized. This is solved iteratively, using the concept of Voronoi regions. The Voronoi region of an agent consists of all parts of the polygon that are closer to that agent than to any of the others, which makes it natural for sensor assignment. As shown by Lindhé *et al.* [35], the same concept can also be integrated with navigation functions to achieve simultaneous flocking and obstacle avoidance.

Centralized communication Ren and Beard [48] have presented a centralized method for formation control, mainly intended for spacecraft. An example could be satellites forming a synthetic aperture for a telescope. The method uses a *virtual structure*, which is an imaginary frame, to which each agent is attached. To move the formation, the position and orientation of the virtual structure is sent out from a central supervisor to every agent, that moves to maintain its relation to the structure. Simultaneously, the framework allows for feedback from each agent if it deviates too far from the structure. This can happen if the actuators of the agent saturate, or if there is some unforseen problem. The structure then slows down, to minimize the formation error.

An example of a combination between centralized and local communication is given by Leonard and Fiorelli [33]. They suggest an algorithm based on artificial potentials, where each agent reacts to the positions of its neighbors. To be able to control the formations of the group, and to herd the agents in a desired direction, they then introduce *virtual leaders*. They are moving reference points, to which the agents react as if they were physical agents. Knowing the position of the leaders requires centralized communication with a supervisor.

Tree-like communication Methods of this kind are often called leader following. With this communication topology, some agents, called *leaders*, are not affected by the movement of the others, so they are free to follow trajectories defined by an overall task. A *follower* is an agent that tries to maintain a given relative position to a leader or another follower. The leader can also be a virtual point, so that all physical vehicles are followers [14].

A subclass of leader following systems is automated highway systems, where cars form *platoons* of vehicles that follow each other autonomously. Here the communication tree has no branches, but forms a single line. Stability of the velocities and inter-vehicle distances in such a topology is called *string stability* and has been thoroughly studied [53].

Connected to this is the more general *leader-to-formation stability* (LFS), proposed by Tanner *et al.* [56]. They consider a slightly broader class of systems than described above, where each follower may follow several leaders simultaneously. LFS is defined approximately as the existence of an upper bound on the error in follower positions as a function of the perturbations to the leader positions. Tanner *et al.* show that LFS is invariant to general kinds of interconnections of groups and how LFS can be proven for a formation controller. They also show that LFS can be used as a design tool to compare different formation topologies and choose the one where errors are attenuated the most.

3.3 Simulation Environments

Many of the algorithms above are tested in both simulation environments and on physical systems, and here are some examples of such simulators.

- **Player/Stage** is a combination of two software packages. Stage is an open-source simulator for multi-agent robotic systems. It simulates robots, sensors and objects in a two-dimensional bitmapped environment. Player is a robotic server that can be used to interface physical robots as well as hardware simulated in Stage. The project homepage can be found at `http://playerstage.sourceforge.net`.
- **MissionLab** is developed by Georgia Institute of Technology and uses the behavioral approach to robot control. It allows the user to graphically connect basic behaviors into more complex structures, and then simulating the result before exporting it to any of the supported robotic platforms. It is open source and available at `http://www.cc.gatech.edu/aimosaic/robot-lab/research/MissionLab`.
- **RoboDaemon** is a simulation and development environment for multi-robot control, developed by McGill University. It is open source and can be found at `http://www.cim.mcgill.ca/ mrl/media/software/daemon.html`.
- **Matlab** is a general math software, with Simulink as its simulation environment. It is not specialized for simulation of robotics, but due to its widespread adoption it is often used anyways. It is a commercial software, owned by The Mathworks Inc., whose homepage is `http://www.mathworks.com`.
- **TrueTime** is originally a simulator for real-time control in embedded systems, but now also contains modules for simulating communication. As described by Årzen *et al.* in [4], it has recently been used for simulation of multiple mobile robots. It is open source, based on Matlab and can be found at `http://www.control.lth.se/truetime`.

3.4 Relation to the Project Objective

Using a formation controller that requires local communication gives advantages such as scalability of the group, lower requirements on communication range and no need for a central supervisor. The drawback is that using local controllers can make it difficult to achieve certain types of formations. A centralized topology, on the other hand, gives more freedom to explicitly prescribe the desired formation. The class of leader-follower methods is interesting because of their simple communication structure, but they are less robust to disturbances and agent failures.

Several open source environments exist that could be useful for for simulating multi-robot systems. It would be preferable to choose one that allows straightforward migration from simulations to experiments with real robots, without having to rewrite the code.

4 Inter-Agent Communication

4.1 Definition

To exploit the full advantage of using multi-agent systems, the agents often need to communicate to synchronize their movement and sensor usage. Since communications have limited range, an active strategy not to lose contact is needed. Performing tasks in a communication-aware manner becomes a crucial ability for each agent in the group.

Since the reasons for interruptions of communication can vary, as well as the requirements on what bandwidth is needed and what delays are tolerated, we have chosen a wide definition of the area:

Find control laws for agents in a multi-agent system, so that the system can perform a task under the constraint of maintaining inter-agent communication.

By stating that the *system* should perform a task, we allow for some agents possibly being fully devoted to fulfilling the communication constraint. We also do not detail the constraint, since it can range from requiring high-bandwidth communications between all agents at all times to requiring sporadic contacts so that messages can be relayed between any two agents in the group in bounded time.

Finally, we restrict the scope of this survey to control laws for the *movement* of agents. For given positions of agents, the questions of routing, media access control as well as other issues on higher protocol levels are difficult, but not treated here.

4.2 Survey

Most of the papers referenced below treat the subject of radio communication, since it is the dominant technology for inter-robot communication. But some also consider communication through infrared light [30], sound [29] or by extracting information from a camera image [36]. Although in practice they are different, the demands for maintaining contact are mainly the same. The signal quality decays with distance, and it is also attenuated by obstacles. For this reason, many of the strategies carry over to other domains that presented in the original papers. An exception to this is the multipath fading effect, which mainly applies to radio signals, but we will return to this later in the survey.

In the following, the higher-level task to be solved is called the *mission*. Existent methods of ensuring inter-agent communication while carrying out a mission, can be roughly divided into two categories (Figure 4.1):

- One or more agents are dedicated relaying agents, which means that they do not actively contribute to solving the mission. Other agents solve the mission, with little or no adaptation to the communication constraint.

- All agents participate in solving the mission, but make a tradeoff between mission completion and maintaining communication.

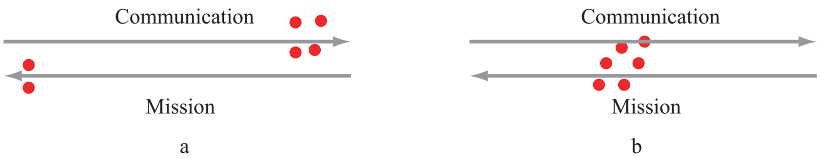

Figure 4.1: Two ways of maintaining inter-agent communication. Alternative *a*: some relay agents prioritize communication, while others prioritize the mission. Alternative *b*: All agents make a tradeoff between communication and the mission.

Approaches that fall under each of these categories will now be described in more detail. We will then end with the somewhat separate issue of disruption-tolerant networks.

Either solving the mission or communicating: Nguyen *et al.* [39] suggest a system of small and cheap relaying robots. The intended application is indoor exploration, and the relay robots follow the lead robot into a building until they detect that the signal strength from the base station falls below a given threshold. A relay robot is then left behind, and the link is routed over that robot. If a relay discovers that it is no longer needed (e.g., because another relay further ahead achieves a direct link to the base), it will use a map sent by the lead robot to catch up with the group, making it available for redeployment. With the help of this internal map, the relays can also be recalled when the mission is over. Similarly, there is an example of small Scout robots [11] using stationary relay stations to enhance the communication range and to allow sending video imagery back to a base station.

A somewhat different approach is presented by Årzén *et al.* [3], using mobile relaying agents that can be sent into a tunnel to reconnect a multi-hop network of stationary sensors, if some sensors are damaged. There is an elaborate power-control protocol to adapt the transmission power of the relay, so as to avoid bandwidth contention and save battery power.

Sometimes, the lead agent has both a mission and communication objective to take into account. An example of this is a system [54], where a chain of relay agents has the task of relaying data from a leader to a fix base station. The lead agent moves towards a goal, under the constraint that it must not lose contact with the adjacent relay agent. Likewise, the relay agents try to maximize the signal strength to the agent ahead of them, under the constraint that they must not lose contact with the agent behind.

Tradeoff between mission and communication: An example of all agents helping to solve the mission is given by Arkin *et al.* [1]. A group of agents is started at the entrance of an unknown building, and one of them starts exploring more or less randomly. When it loses the line of sight with the rest, it retraces its trajectory until the contact is reestablished. It then stops and becomes the *anchor* for the next robot, that can get further into the building before losing contact. The last robot of the group does not stop like this, but retraces in case of loss of contact, and tries another direction of movement.

In the above case, the agents at some point switch from sensing to relaying. Another way is to let the agents continuously solve, e.g., a sensing task while making a tradeoff between sensing performance and communication. This can be formulated as gradient climbing in one or more scalar utility functions, describing sensing and communication performance, respectively [9, 44].

Another mission, as opposed to the sensing described above, could be to move the group. Esposito *et al.* [15] have studied a method to move a for-

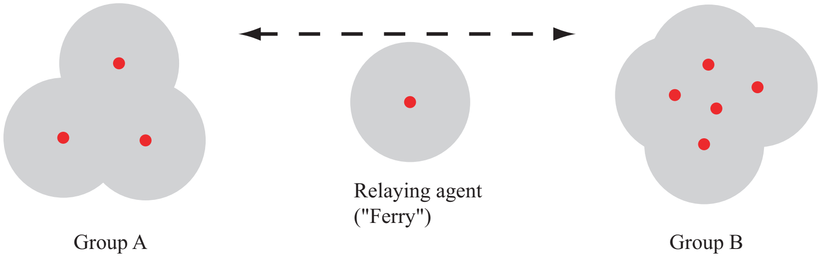

Figure 4.2: A schematic illustration of a disruption-tolerant network, where a relay agent stores data and physically transports it between two connected sub-networks. The grey regions indicate the communication range of each agent.

mation of agents from one configuration to another (which could also mean the same formation but in another place). This is done without losing line of sight between specified agents, in an environment with obstacles. A more experimental approach to the same problem formulation is presented by Powers *et al.* [45], using ground robots with a behavioristic control law. A similar problem is solved by Pereira *et al.* [43], by reducing it to formation control.

As described in the introduction, all papers presented so far use simplified models of radio propagation, assuming that the signal strength is a function of the distance and/or depends on a free line of sight. For radio signals, this overlooks the significant effect of multipath fading in urban or indoor environments. As shown by Lindhé *et al.* [34], established models for fading from the literature of radio propagation can be used to predict how small corrections (in the order of centimeters) to the position of an agent can yield significant improvements in signal strength. For missions such as surveillance or searching, this allows an agent to make small movements that do not affect the mission, and gain communication performance.

A very simple way of performing the tradeoff is to program all agents to perform the mission as usual, but impose a hard constraint that they must not continue if they lose communications. Hsieh *et al.* [25] describe experiments with surveillance robots moving towards separate goals, while monitoring the achieved data rate to the base station. If the data rate falls below a given threshold, they stop. After a certain waiting time, if the data rate has not recovered, they retrace to improve communications. The waiting adds some robustness to short temporal dips in the signal strength.

Disruption-tolerant networks: A somewhat separate category of methods can be used if connectivity is not measured at each instant, but rather over a time interval. The constraint is that during some interval, there must have existed connections that together allowed a message to be relayed between any two agents. This is called *disruption-tolerant networks* [6], and is illustrated in Figure 4.2. Mobile agents with onboard memory could act as *ferries*, relaying messages by storing them and sending them at a later time, when they have contact with the intended receiver or another relaying agent. For applications where delays in the time-scale of agent movements are acceptable, this can be shown to overcome fundamental limitations on the possible throughput [22].

Zhao *et al.* [58] suggest four alternative ways of computing trajectories for dedicated ferries. A practical example of such an application is an Australian group [12], using unmanned underwater vehicles as data mules, collecting data from stationary underwater sensors. An example of agents that do not adapt their trajectories is a network of public transportation buses, also carrying computers with wireless connections. By storing data on board, they can connect two disjoint wireless networks in different parts of a university campus [6].

4.3 Relation to the Project Objective

Using relaying agents is a proven method that appears to be robust, at the expense of taking agents off the solving of the task. We have found no methods that explicitly minimize the number of relaying agents. A more complex method is to let the agents make a tradeoff between communication and task completion, but this is probably less useful in the context of searching since for the task to be completed, all parts of the search area must be visited. The notion of disruption-tolerant networks may be more promising, since searching can be performed with just intermittent communication and this may allow more effective usage of all agents.

5 Maintaining a Line of Sight

5.1 Definition

When searching an area, and also to form a secure perimeter, agents may need to establish a line of sight to selected other agents and maintain this during group motion. We formulate this area as:

Find control laws for agents in a multi-agent system, so that they can establish and maintain a line of sight between selected agents in the presence of obstacles and while possibly also performing group motion.

5.2 Survey

Maintaining a line of sight between agents in a multi-agent system is not very well studied in its own right. Instead, it arises as a constraint in different contexts, some of which will be described below. We will then present some control strategies that have been presented in the literature.

As described in Chapter 4, maintaining a line of sight is often motivated by the need for radio communications, whose signals are strongly attenuated by obstacles [45, 1, 15]. In other cases [47], a pair of agents can maintain a line of sight between them and use it for sensing. One of the agents move, and if it is occluded there is obviously an obstacle somewhere between the agents. This way, obstacles with reflective or very absorbing surfaces can be detected, which can be difficult for ordinary visual or ultrasonic sensors.

There are also applications where agents move one at a time, in a leapfrogging manner, while visually tracking each other [21]. This way, they can triangulate each other's position and thus improve positioning accuracy. A similar technique can be employed if using ultrasonic ranging for the triangulation [38], which also requires a clear line of sight.

Another system using the line of sight for collaborative navigation, is presented by Parker *et al.* [41] They use two advanced helper robots that deploy simpler mobile sensors by herding them to their positions. During the transport phase, a helper robot leads the way and the other helper comes last. This robot maintains a line of sight to all simple robots and the leader, and controls the simple robots via radio.

As a final example, in human-robot interaction it is of interest to enable a robot to visually track a human operator [17]. This lets the operator teach an environment to the robot, or just move it in a simple way. The robot then needs to maintain a line of sight to the operator so as not to get lost.

In the above applications, some different methods of maintaining the line of sight are represented. Esposito and Dunbar [15] use a potential function method. Navigation to the goal is done by following the negative gradient of a scalar navigation function (whose global minimum is located at the goal). The objectives of staying within range of one's neighbors and maintaining line of sight are also encoded as scalar functions. Assuming that agents i and j are to be connected, and their positions are at q_i and q_j, respectively, the potentials

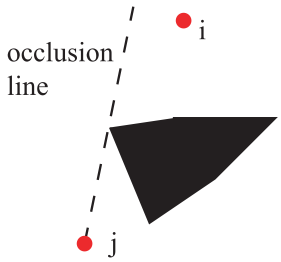

Figure 5.1: When robots i and j lose line of sight, robot i moves back towards the *occlusion line* to restore contact.

for range and line of sight for agent i are

$$\phi_{ij}^{range}(q_i, q_j) = \begin{cases} 0 & |q_i - q_j| < \rho_{max} \\ |q_i - q_j|^2 - \rho_{max}^2 & |q_i - q_j| \geq \rho_{max} \end{cases}$$

and

$$\phi_i^{los}(q_i, q_j) = \begin{cases} 0 & \text{if L.O.S.} \\ d^2(q_i, OL) & \text{else} \end{cases} .$$

Here, $d(q_i, OL)$ is the shortest distance from q_i to the *occlusion line*, which is the line connecting the two agents before the line of sight was lost, and ρ_{max} is the maximal communication distance. This is depicted in Figure 5.1.

The control of the agent is then chosen so that none of the potential functions increase. In the general case, Esposito and Dunbar cannot prove that there is always such a feasible descent direction, but simulations indicate that the approach works.

Powers and Balch [45] use a behavioristic method which bears some similarity to the above, but is formulated differently. Each agent runs four basic behaviors (i.e., modules that output a control command), called *preserve-communication*, *move-to-goal*, *avoid-static-obstacles* and *maintain-formation*. In their framework, they only allow each agent to maintain line of sight to two others, and the *preserve-communication* behavior attempts to maximize the communication quality to these neighbors. The resulting control of the agent is then a weighted sum of the output of each behavior, where the *preserve-communication* behavior has the highest weight. Such behavioristic methods provide a useful structure for combining several objectives, but the drawback is that it is hard to prove properties of the combined system.

Finally, there are also more geometric methods, such as presented by Grabowski *et al.* [21] or Rekleitis *et al.* [47]. They move one agent at a time and explicitly test possible candidate future positions to see if they satisfy the line of sight constraint. This has the advantage of ensuring that the constraint is fulfilled, but does not provide guarantees for feasibility when combined with arbitrary other tasks such as group movement or searching.

5.3 Relation to the Project Objective

As described above, maintaining a line of sight between two UGVs can make the task of sensing an intruder much easier. If the area between the agents is known to be obstacle free, the agents simply need to detect losing the line of sight to know that something has passed between them. The most reliable way of achieving a line of sight seems to be a direct geometric method, but there are no guarantees that this is feasible, given the number of agents and the environment.

6 Conclusions

This survey was divided into four areas, all of which are relevant in the context of using UGVs for searching and securing an area. Below are the conclusions for each area, focused on what is relevant for the continuation of this project.

For coordinating searchers, there are several algorithms if the area can be represented as a simple polygon, i.e., it has no "holes". But for more realistic environments, with obstacles, there are few results on complete algorithms for searching. One way of solving this could be to first position stationary searchers at places where the area has "loops", which could reduce the problem to a simple polygon. This can then be searched using an existing algorithm. A possible improvement could be to identify the minimal number of such places and/or strategies that allow for not blocking all loops at the same time.

In the area of flocking and formation control, there are plenty of possible methods, depending on the communication topology, agent dynamics and to what extent different roles can be assigned to different agents. In general, algorithms that require communication only between neighboring agents allow changing the group size and require no central supervisor, but also make it more difficult to synthesize a desired formation. Centralized methods allow for arbitrary formations, but place stricter demands on the communication capacity.

Regarding how to maintain inter-agent communication, two categories of methods appear to be feasible in the context of searching. One is to use relaying agents, whose role is to retransmit information to other agents that solve the superior task. The other is to make the network disruption-tolerant, so that data can be buffered and sent when agents come within range of each other. In such networks, some agents can be dedicated "ferries", improving throughput and decreasing latency by physically transporting data.

The last area of the survey, how to maintain a line of sight between agents, is not very well studied in its own right. Instead, it often arises as an objective for preserving communications, since radio signals are attenuated by obstacles. The most reliable methods are based on simple geometric conditions on what points are feasible to move to. However, these methods do not explicitly address the question of whether it is possible to achieve a line of sight, given some environment and a number of available agents.

7 Searcher Coordination – Short Reviews of Individual Papers

7.1 Sweeping Simple Polygons with a Chain of Guards, by Efrat, Guibas, Har-Peled, Mitchel and Murali

This paper can be found in reference [13].

7.1.1 Problem Formulation

The authors consider the problem of searching a simple polygon, taking recontamination into account. They require that the searchers form a chain with line of sight between adjacent agents in the chain at all times, and that the end agents of the chain should be on the boundary of the polygon.

The paper gives an exact limit on the number r^* of agents needed to sweep a given polygon. It also gives an algorithm of complexity $O(r^* n^3)$, with n being the number of vertices in the polygon, for computing a search schedule.

Since both computing r^* and the schedule is computationally demanding, the authors also present an approximative algorithm of complexity $O(n^2)$, to compute a search schedule that requires at most $r^* + 2$ agents.

7.1.2 Relation to Searcher Coordination

This paper solves the problem for areas that do not contain obstacles in the strict interior (namely simply connected polygons). An obstacle along the boundary of the search area could be integrated into the bounding polygon. The assumption that the agents can only sense along a line between them is somewhat limiting, so the completion time or number of searchers may decrease if a more realistic sensor model is used. As a simple example, if this algorithm would be executed with 1-searchers, the amount of agents needed would be reduced by one, since there is no need for the last agent in the chain.

7.1.3 Proposed Solution Method and Mathematical Tools Used

Geometry and properties of graphs

7.1.4 Personal Comments, Pros and Cons, assessment of paper quality

The paper is easy to follow and the results referenced above are simple to find. The main focus of the paper seems to be on the computational complexity of more and more coarse approximations of r^* and the search schedule, rather than on what is fundamentally possible.

7.2 Visibility-based pursuit-evasion with limited field of view, by Gerkey, Thrun and Gordon

This paper can be found in reference [20].

7.2.1 Problem Formulation

The paper considers ϕ-searchers, that have a realistic sensor with a field of view (FOV) of angle ϕ. Just like for k-searchers, the sensor can be turned with bounded velocity in any direction.

There is a proof that the larger the FOV, the more polygons an agent can search. The set of searchable polygons grows continuously with ϕ.

The environment is assumed to be a polygonal free space, which is equivalent to a simple polygon. Thus no obstacles in the interior of the environment are tolerated.

The main contributions of the paper are a proof that computing the minimum number of ϕ-searchers for a given environment is NP-hard, and a provably complete search algorithm for one ϕ-searcher in environments where this is possible. The method can be extended to multiple searchers, but uses prohibitively large memory for its expanded state-space and is no longer complete. So although they demonstrate an example with two ϕ-searchers, it is centralized and not very practical for larger groups.

7.2.2 Relation to Searcher Coordination

This solves the searcher coordination on a simple polygon, and with a single ϕ-searcher. The discussion on the FOV angle ϕ is interesting from a hardware design point of view, since it could be a motivation to choose a camera with 90° FOV to be able to cover rectilinear rooms with one robot (while still keeping track that no one passes by outside).

7.2.3 Proposed Solution Method and Mathematical Tools Used

The algorithm is based on exact cell decomposition, where each cell represents a region where the robot can move without changing the topology of what is visible and not. So the only critical times are when the robot crosses the boundary between two cells. The global path planning problem is then reduced to planning over a graph.

7.2.4 Personal Comments, Pros and Cons, assessment of paper quality

The introduction is interesting and the model of a ϕ-searcher is more realistic than the k-searcher, but the derivation of the algorithm is very long (10 pages). Well-written but lengthy. The paper gives the number of agents needed to search an environment, but does not provide a good algorithm for searching with multiple agents.

7.3 Extending Mobile Security Robots to Force Protection Missions, by Carroll, Everett, Gilbreath and Mullens

This paper can be found in reference [7].

7.3.1 Problem Formulation

This is an overview of the Mobile Detection Assessment Response System (MDARS) project, initiated by the US military. The purpose is to develop robots that can guard storage facilities and also do inventory. The paper describes the system on a high level, without any details on control algorithms. The platform they used is a diesel-powered truck with basically three payloads: an intruder identification system using infra-red vision, an inventory system using RFID readers, and a system to check the status of locks, by communicating with them over a wireless channel.

The paper gives some results from field tests at two occasions. The first was a formal technical test and the second was a simulated deployment where the operators were not aware of what would happen at each test, and test subjects tried to sneak up to the robots. The performance of intruder detection, inventory and lock control were all satisfactory.

7.3.2 Relation to Searcher Coordination

This is the same type of application, but the specific pursuit-evasion problem is not solved by this system.

7.3.3 Proposed Solution Method and Mathematical Tools Used

Not detailed

7.3.4 Personal Comments, Pros and Cons, assessment of paper quality

Interesting as a motivating paper, but does not contain any useful algorithms or tools for the searcher coordination. Easy to read, except for all the acronyms.

7.4 Distributed Deployment of Asynchronous Guards in Art Galleries, by Ganguli, Cortés and Bullo

This paper can be found in reference [18].

7.4.1 Problem Formulation

The authors consider the classic art gallery problem, but the case when the map is not known a priori. Instead, all agents start from a corner of the map and then incrementally distribute over the environment.

The environment is modelled as a simple, not necessarily convex polygon. This approach does not allow any holes in the polygon, such as obstacles that are completely inside it. Each agent is assumed to have omnidirectional vision, i.e., they are 2π-searchers. The end result is two slightly different algorithms that provably give full coverage of the whole polygon.

7.4.2 Relation to Searcher Coordination

This paper solves the art gallery problem by positioning guards to achieve complete coverage, which in a sense solves the searcher coordination but not with a minimal number of agents. But the construction of the visibility-based tree graph (described below) could be useful also for the searcher coordination, if one could find conditions for when it is safe to abandon parent nodes to free agents for exploration of new vertices.

7.4.3 Proposed Solution Method and Mathematical Tools Used

Starting from any vertex of the polygon, the algorithm constructs a visibility-based tree graph, consisting of selected vertices. The graph grows by computing the visible region from a vertex and then for each "gap" in the visible region (i.e. a side of it that is not a "wall"), selecting a new vertex that is out of sight of the parent vertex. The tree can be constructed in a depth-first manner, or using a random strategy. Both give complete coverage of the whole polygon.

The main mathematical tool used is graph theory.

7.4.4 Personal Comments, Pros and Cons, assessment of paper quality

The main difficulty of the paper is the complex notation necessary to understand the interesting results. The results are relevant and the paper is very complete, with illustrative simulations at the end.

7.5 Complete Multi-Robot Coverage of Unknown Environments with Minimum Repeated Coverage, by Ge and Fua

This paper can be found in reference [19].

7.5.1 Problem Formulation

This paper deals with the coverage problem, where an agent must physically drive over all free space. Such problems arise in demining, lawn moving or cleaning applications. The problem considered here is to minimize the repeated coverage, i.e., the areas that are covered more than once.

Each agent has a circular footprint and the algorithm allows for obstacles inside the search area. The obstacles may have arbitrary shape, as long as all openings are wide enough for the robot to pass through.

7.5.2 Relation to Searcher Coordination

Not very related to the searcher coordination since the sensor model (using only the footprint of the robot) allows an intruder to move undetected as long as it does not collide with a searcher.

7.5.3 Proposed Solution Method and Mathematical Tools Used

Not a very mathematical approach, but part of the method is to consider the areas already covered as obstacles. This way, repeated coverage is avoided unless there are narrow passages into obstacles, where the robot needs to pass twice.

7.5.4 Personal Comments, Pros and Cons, assessment of paper quality

A fairly complex algorithm for an already well-studied problem. The contribution of this paper is to focus on avoiding repeated coverage rather than minimizing the completion time, but that seems like a weak motivation. The completion time is usually a more relevant metric and if covering some area makes the search faster, the use of avoiding it is questionable.

7.6 Multirobot Cooperation for Surveillance of Multiple Moving Targets - A New Behavioral Approach, by Kolling and Carpin

This paper can be found in reference [31].

7.6.1 Problem Formulation

The paper considers the problem of finding and tracking moving targets, using several agents. The metric for success is the average over time of the fraction of targets that are observed by at least one agent. The working area must be free from obstacles.

The agents have dynamic constraints with bounded turning velocity and their field of view is varied in the interval $[45°, 315°]$. Simulations show that the performance decreases smoothly as the field of view gets narrower. Targets are considered as tracked if they are within the field of view and inside a given sensing range.

The algorithm is tested against two kinds of targets: those moving randomly and those actively evading the searchers.

7.6.2 Relation to Searcher Coordination

This paper does not really focus on the searcher coordination. Instead the goal is to track intruders as effectively as possible over time. In the case that no intruders are detected, the control law simplifies to avoiding other searchers and wander around to look for new targets.

The algorithm does not guarantee detection of all intruders or any given performance, but it is evaluated through simulations.

7.6.3 Proposed Solution Method and Mathematical Tools Used

The authors propose a behavioral approach. Each discovered target is tagged by a searcher to avoid contention, and each searcher has three modes: Follow, Help and Explore. In the Follow mode, the agent is controlled by the sum of repulsive or attractive forces from all other robots and from its tagged targets. The agent is repelled from other agents, and attracted or repelled from its targets to attain a desired distance to them. If a target is getting close to the boundary of the sensing area, the searcher can issue a *help* request.

If a searcher has no tagged targets, it can respond to *help* requests in the Help mode, by taking over some of its tagged targets. Otherwise it enters the Explore mode, which means seeking new targets.

7.6.4 Personal Comments, Pros and Cons, assessment of paper quality

A fairly simple approach, where the main emphasis seems to be on experimentally evaluating the performance. Does not contain many new ideas, and is not very applicable to the searcher coordination.

7.7 Probabilistic Strategies for Pursuit in Cluttered Environments with Multiple Robots, by Hollinger, Kehagias and Singh

This paper can be found in reference [24].

7.7.1 Problem Formulation

This paper allows for arbitrary environments, as long as they can be divided into convex cells. The intruder is considered detected if a searcher enters the same cell as the intruder, which implicitly requires 2π-searchers. Multiple searchers are allowed, but only one intruder.

The problem solved is to minimize the expected time to discover the intruder, rather than giving guarantees for discovery. The authors argue that his allows simpler comparison with other algorithms, and that in more complex (realistic) environments, it is difficult to guarantee capture with few agents.

7.7.2 Relation to Searcher Coordination

The paper solves the searcher coordination in the case of a single intruder, but only in a statistical sense. There is no upper bound on the time to find the intruder and nothing stops the intruder from reentering already searched areas, which then need to be searched again. The assumptions on the environment are very general and the searchers are modelled as 2π-searchers.

7.7.3 Proposed Solution Method and Mathematical Tools Used

This paper focuses on indoor environments, that are represented as graphs where each node represents a convex cell of the floor plan. Nodes are connected if there is a direct passage between the cells.

The searchers maintain a probability vector p, where p_i is the probability of the evader being in cell i (and p_0 represents the evader being caught). If a searcher enters a new cell and finds it empty, p is updated through a matrix multiplication. Much like a particle filter, p is also time-updated by multiplication with a dispersion matrix that models the probability that the evader has moved to an adjacent cell.

Each searcher uses a cost function to choose where to go. Possible cost functions are the entropy of p or the vector distance $||p - \bar{p}||$, where $\bar{p} = (1, 0, \ldots, 0)$. The cost function is then minimized over a planning horizon of N future moves. This planning can be done in a coupled way, by computing the moves of all other searchers, or in a decoupled way by assuming that the other searchers will stand still. In the latter case, the searchers only need to know the positions of the others at each planning instance.

7.7.4 Personal Comments, Pros and Cons, assessment of paper quality

This is a novel approach and it has several advantages, such as allowing a simple way to incorporate prior knowledge and also uncertainty in the sensors. (The latter is accomplished by setting $p_i = \epsilon > 0$ if a searcher finds cell i empty, to account for the small probability of missing the intruder.)

The sensor model and problem definition is not very complete, and the mechanism for coupling is less elegant than the rest of the paper. Nevertheless, the paper is interesting and easy to follow.

7.8 Randomized Pursuit-Evasion in a Polygonal Environment, by Isler, Kannan and Khanna

This paper can be found in reference [27].

7.8.1 Problem Formulation

This paper treats two problems. The first is the single-searcher, single-intruder pursuit-evasion problem. The assumptions are worst-case: the evader knows the position of the pursuer and the evader is arbitrarily fast. The goal of the pursuer is to get the evader in its field of view, assuming a 2π-searcher. The environment is assumed to be a simply connected polygon (thus without any internal obstacles).

In general it is impossible for a single pursuer to find such an evader in a deterministic way, but this paper statistically considers what happens if the pursuer makes several random runs through the polygon. The result is a strategy that can achieve an arbitrarily high probability of finding the evader.

As an extension, the authors also consider the problem of making a hybrid control law to control an agent with dynamic constraints according to the discrete algorithm. They also remark on the case of several searchers in a polygon that is not simply connected.

The other problem is called the lion-and-man problem, where a pursuer and evader are inside a rectangle and the pursuer tries to catch up with the evader. This part of the paper is not reviewed here.

7.8.2 Relation to Searcher Coordination

This paper solves a subset of searcher coordination, with a single intruder and searcher. The severe limitation lies in the fact that the environment has to be a simply connected polygon, i.e., there cannot be any obstacles in the interior of the search area. Finally the stochastic method will yield very time and energy consuming solutions that may not be very appealing in a more complex environment. In the conclusions, there is a comment that in some environments, using a random approach is the only way to catch the evader, and in some other cases the deterministic strategy takes so much time that it compares to the random strategy. The authors also stress that the random algorithm requires less computations and only requires a map that is topologically correct.

The authors remark that if the environment is not simply connected, one can use additional (stationary) agents that block passages and thus turn the topology back to a tree graph. If the environment has h holes in it, then $O(\sqrt{h}+1)$ searchers are needed for this.

7.8.3 Proposed Solution Method and Mathematical Tools Used

The key observation is that a simply connected polygon can be triangulated and represented as a graph, where each node is a triangle. If triangles sharing a side (so an agent can move between them) are connected, the graph forms a tree, with no loops. If a searcher moves from the root of the tree towards one of the end nodes (leaves), an evader cannot escape *if* it happens to be in that leaf. This means that if the searcher makes enough such round towards randomly assigned leaves, it will eventually with probability one catch the evader.

7.8.4 Personal Comments, Pros and Cons, assessment of paper quality

Well-written paper, but the main results are more difficult to access than in the earlier paper [26]. This version also contains the lion-and-man problem, which is not very connected. Why not make two papers?

7.9 Planning Expected-time Optimal Paths for Searching Known Environments, by Sarmiento, Murrieta-Cid and Hutchinson

This paper can be found in reference [50].

7.9.1 Problem Formulation

The environment is assumed to be a simple polygon (with no internal obstacles), and there is only one 2π-searcher. The goal is to find a trajectory that minimizes the expected time to detect a stationary target in the polygon. The authors assume uniform a priori probability of finding the target.

7.9.2 Relation to Searcher Coordination

Does not solve the searcher coordination, since it requires a stationary target. It also assumes only one searcher, which is not the focus of this survey.

7.9.3 Proposed Solution Method and Mathematical Tools Used

The algorithm works in two layers: The first, combinatorical layer divides the polygon into different regions, which must all be visited to find the target. It also decides the order in which they should be visited. Then a continuous layer finds the optimal trajectory between each region.

7.9.4 Personal Comments, Pros and Cons, assessment of paper quality

The division of the polygon into regions is done in a very straightforward and simple way, which could be useful. The optimization part is a nice idea and well described, but not very applicable to our problem.

7.10 Multi-robot Boundary Coverage with Plan Revision, by Williams and Burdick

This paper can be found in reference [57].

7.10.1 Problem Formulation

The problem considered is to inspect the boundaries of all obstacles in a given area, which in practice would correspond to filming all walls of a group of buildings. The agents are 2π-searchers with a limited sensor range.

The algorithm has the ability to replan while executing the search, if an agent is added or disappears or if an external operator excludes some boundary from the set that needs to be watched.

7.10.2 Relation to Searcher Coordination

Does not solve the searcher coordination, since the search is focused on watching obstacles rather than searching for an intruder in the free space. There is also no mechanism to stop an evader from moving back into already watched areas.

7.10.3 Proposed Solution Method and Mathematical Tools Used

The problem is reduced to a graph where the edges are either paths that must be traversed to inspect the boundaries, or transport paths to get between disconnected obstacles. The resulting discrete planning problem on the graph is equivalent to the NP-hard k-Rural Postman Problem.

7.10.4 Personal Comments, Pros and Cons, assessment of paper quality

A well-written paper with many useful illustrations.

7.11 Searching for a Mobile Intruder in a Polygonal Region, by Suzuki and Yamashita

This paper can be found in reference [52].

7.11.1 Problem Formulation

This is the first paper where the search problem on a polygon is considered. The environment is assumed to be a simple polygon, and only a single k-searcher is considered. The paper finds necessary and sufficient conditions for a polygon to be 1-searchable, i.e., when there exists a strategy that guarantees detection of an arbitrarily fast evader by a single 1-searcher. Similarly, conditions for a polygon to be 2-searchable are presented. No results are given on how to find such strategies in general.

7.11.2 Relation to Searcher Coordination

This is a simple version of the searcher coordination, with just one searcher and on a polygon with no internal obstacles. The paper gives only conditions for polygons to be 1- or 2-searchable, but no schedules to perform the search. There is a conjecture that the 2-searcher may be able to search all polygons that are ∞-searchable, which is interesting from a practical point of view. (This was later proved by Park *et al.* [40].)

7.11.3 Proposed Solution Method and Mathematical Tools Used

Geometry and simple combinatorics

7.11.4 Personal Comments, Pros and Cons, assessment of paper quality

A rather lengthy paper, whose results are not so interesting for the practical solutions to the searcher coordination. The interesting part is the problem formulation. Nice illustrations of polygons that are not ∞-searchable.

7.12 Simple Algorithms for Searching a Polygon with Flashlights, by Lee, Park and Chwa

This paper can be found in reference [32].

7.12.1 Problem Formulation

The authors consider k-searchers, where the sensor can be turned with bounded angular velocity. The environment is assumed to be a simple polygon. The paper contains a necessary and sufficient condition for a polygon to be 1-searchable and an algorithm for constructing a search schedule for a 1-searcher. Likewise, there is a necessary and sufficient condition for a polygon to be 2-searchable and an algorithm for finding a search schedule for the 2-searcher.

7.12.2 Relation to Searcher Coordination

This solves the searcher coordination on a simple polygon, if it is 1- or 2-searchable, using 1- or 2-searchers.

7.12.3 Proposed Solution Method and Mathematical Tools Used

Simple geometry

7.12.4 Personal Comments, Pros and Cons, assessment of paper quality

Nice and short paper with clearly stated results.

8 Flocking and Formation Control – Short Reviews of Individual Papers

8.1 Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules, by Jadbabaie, Lin and Morse

This paper can be found in reference [28].

8.1.1 Problem Formulation

The authors study a simple model of consensus for the headings of multiple agents. At each (discrete) time step, every agent updates its heading to the average of its own and that of its neighbors. The set of neighbors is defined by a sensing radius, and changes with time. The question is to find conditions for when the system converges to a single heading.

They also introduce a leader, whose heading is fixed, and study how the group converges to this heading.

8.1.2 Relation to Flocking and Formation Control

This only yields flocking, since the agents will only assume the same heading and there is no group cohesion. Nevertheless, it is interesting because of the general requirements on the switching neighbor topology.

8.1.3 Proposed Solution Method and Mathematical Tools Used

The authors treat each change of neighbor topology as a switch from one linear system to another, and then show that the overall switched system is stable for a wide class of switching signals, without trying to find rules for how the switching sequence looks. The requirement is that there is a time interval T such that the neighbor graph is connected at least once in every interval.

They also propose a simpler control law which has the advantage that all linear systems have a common Lyapunov function, so the stability analysis is simpler. This control, however, is not truly distributed since every agent needs to know an upper bound on the number of agents in the group.

Then they introduce a leader, which is an agent with a fix heading. It is shown that under the same requirements as above, the whole group will converge to the heading of the leader. There is also a similar result on convergence to the leader's heading for continuous time.

8.1.4 Personal Comments, Pros and Cons, assessment of paper quality

An interesting and well-written paper, even though it does not directly apply to flocking. There are interesting discussions around all of the relevant results.

8.2 Behavior-Based Formation Control for Multirobot Teams, by Balch and Arkin

This paper can be found in reference [5].

8.2.1 Problem Formulation

The problem considered is to achieve formations for up to four ground vehicles, each having a unique ID. The possible formations are line, column, diamond and wedge. The place of each robot in the formation is encoded from the beginning.

8.2.2 Relation to Flocking and Formation Control

This is a version of formation control, where each agent has to be given an ID and the formation is rigidly encoded. Depending on the formulation of the maintain-formation behavior, communication can be global or local.

8.2.3 Proposed Solution Method and Mathematical Tools Used

Each robot runs five different behaviors in parallel, each computing a desired velocity vector:

avoid-static-obstacle	Move away from obstacles
avoid-robot	Move away from other agents
move-to-goal	Move towards a waypoint for the group
noise	Dissolves deadlocks
maintain-formation	Move towards the prescribed position in the formation

The three first behaviors are fairly intuitive, but the fourth is included to avoid getting stuck at saddle points of the vector fields surrounding obstacles. The fifth behavior steers the agent towards its position in the formation. This position can be relative to the leader, to the center of mass of the group or to a single designated neighbor.

All behaviors are weighted according to their relative importance, and the final control command to the robot is the weighted vector sum of the output of each behavior. This sum is then normalized.

The result is tested in *MissionLab*, a simulation environment from Georgia Tech, in different types of obstacle-free or cluttered scenarios. It was then tested on laboratory robots and finally on HMMWV jeeps.

8.2.4 Personal Comments, Pros and Cons, assessment of paper quality

Very focused on experimental validation, while the actual control laws were not described in much detail. The approach seems robust and easy to understand, but there are a lot of parameters to trim before it works well.

8.3 Flocking in Fixed and Switching Networks, by Tanner, Jadbabaie and Pappas

This paper can be found in reference [55].

8.3.1 Problem Formulation

This paper describes flocking for kinematic single integrator agents, whose control law consists of two terms. One aims to align the velocities of all agents, and the other is the sum of the negative gradients of an inter-agent potential function. The potential function is a function of only the distance between two agents, and it has a unique minimum at a desired inter-agent spacing.

Two kinds of communication topologies are considered: First the communication graph is fixed, and then it is time-varying, by letting agents communicate with all neighbors within a given sensing radius.

8.3.2 Relation to Flocking and Formation Control

This gives formations, or at least flocking and group cohesion for agent with single integrator dynamics, both for fixed and proximity-based neighbor sets. In general the group does not necessarily converge to the desired formation, but this can be enforced by choosing the communication graph as a tree. The agents do not need any IDs, but can be identical.

8.3.3 Proposed Solution Method and Mathematical Tools Used

In the case of a fixed communication topology, the sum of all potentials is used as a Lyapunov function, and convergence to an equilibrium formation and equal velocities is shown through a La Salle argument. The paper exploits the properties of eigenvalues of Laplacian matrices corresponding to connected graphs.

The condition to attain this equilibrium is that the communication graph is connected, i.e., there is a path for information to propagate between any two agents. It is worth pointing out that if the communication graph is a tree, the equilibrium represents a formation where the inter-agent distances are the desired, but in the general case there can exist local minima that differ from the ideal formation. Since the potential function is unbounded when two agents come close to each other, no two agents that are directly connected can collide.

In the case of switching topology, the result is similar: The group will converge to an equilibrium and equal velocities, if the communication graph is connected.

8.3.4 Personal Comments, Pros and Cons, assessment of paper quality

A well-written paper with very nice results. The two main theorems are easy to find.

8.4 Information Flow and Cooperative Control of Vehicle Formations, by Fax and Murray

This paper can be found in reference [16].

8.4.1 Problem Formulation

The authors consider a system of agents with linear dynamics, and the focus is on studying how the communication topology affects the convergence to a formation. The control of each agent is based on its own position and the average relative vector to all agents in its neighbor set.

8.4.2 Relation to Flocking and Formation Control

The results in the paper apply to formation control for agents with linear dynamics, but it is conjectured that the principles developed apply also to nonlinear agents. To form a formation, each agent is given a predetermined offset from the group center.

The results give insight into how the communication topology of the group affects formation stability.

8.4.3 Proposed Solution Method and Mathematical Tools Used

The communication topology is encoded as a graph, and the eigenvalues of the corresponding Laplacian matrix are studied. The Laplacian matrix of a connected graph has one zero eigenvalue, and the authors give a Nyquist-like theorem relating the location of the nonzero eigenvalues to the stability of the formation.

Several kinds of communication topologies are then studied, and their nonzero Laplacian eigenvalues plotted. Qualitatively, a strongly connected graph gives a formation with good stability margins, while periodic graphs (where information flows in “loops”, in just one direction) tend to cause oscillations that degrade the stability margins.

Finally, the developed tools for stability analysis are used to investigate the possibility to filter the information flow between agents. By applying suitable filtering, the convergence speed of the information flow can be adapted to the movement of the agents, which avoids overshoots and oscillations in the convergence to a formation.

8.4.4 Personal Comments, Pros and Cons, assessment of paper quality

An interesting paper with fundamental results, but overwhelmingly many results collected in one single presentation.

8.5 Virtual Leaders, Artificial Potentials and Coordinated Control of Groups, by Leonard and Fiorelli

This paper can be found in reference [33].

8.5.1 Problem Formulation

The authors consider the problem of designing decentralized control laws for coordinating a group of agents. All agents are considered identical, and have double-integrator dynamics.

8.5.2 Relation to Flocking and Formation Control

This is a formation control algorithm where the control laws are decentralized except for the virtual leaders. All agents must know the positions of the virtual leaders. To be precise, since the potential function is constant outside some radius (denoted d_1 in Figure 8.1), it does not affect the control of the agent, so each agent only needs knowledge about neighbors and virtual leaders closer than this radius.

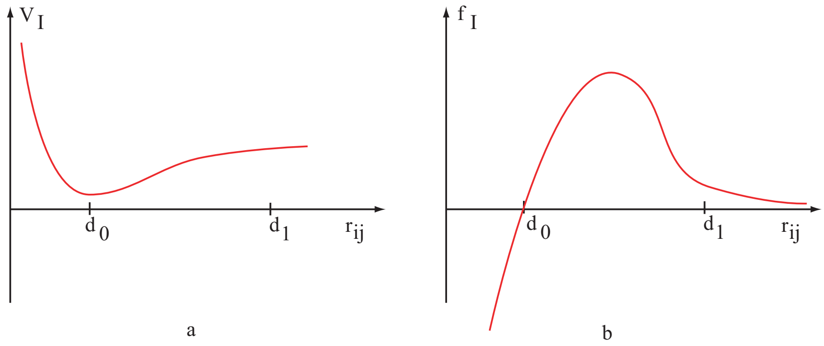

Figure 8.1: An example of artificial potential function (a) and corresponding force (b).

8.5.3 Proposed Solution Method and Mathematical Tools Used

The authors introduce *artificial potentials*, which are scalar functions of the distance between two agents. Now assume that there are N agents and the position of agent n is r_n. The potential of agent i with respect to agent j is then denoted $V_I(r_{ij})$, where $r_{ij} = r_i - r_j$. The artificial potential has a minimum at the desired inter-agent distance d_0, and approaches infinity as $r_{ij} \to 0$. An example of such a function is given in Figure 8.1a.

The suggested control law for agent i is

$$u_i = -\sum_{j \neq i}^{N} \nabla_{r_i} V_I(r_{ij}) + f_{v_i},$$

where f_{v_i} is a velocity-dependent damping term, used to ensure asymptotic convergence to an equilibrium. The gradient of each potential function yields a force that tries to achieve the inter-agent distance d_0, as illustrated in Figure 8.1b.

To be able to control the group, both its overall movement and the type of formation, *virtual leaders* are introduced. They do not exist physically, but are moving reference points, which the agents treat at ordinary neighbors. By placing such virtual leaders, adjusting the equilibrium distance d_0 and the dissipation term f_{v_i}, different types of group movement can be created. The group can be made to circle around a stationary leader or form different formations that move in a desired direction.

8.5.4 Personal Comments, Pros and Cons, assessment of paper quality

A well-written paper with illustrative figures. Interesting approach to add virtual agents that affect the group motion.

8.6 Formation feedback control for multiple spacecraft via virtual structures, by Ren and Beard

This paper can be found in reference [48].

8.6.1 Problem Formulation

The authors consider the problem of maintaining rigidly prescribed formations, while being able to move or scale the formation. The main application is in control of groups of spacecraft.

8.6.2 Relation to Flocking and Formation Control

This is a centralized formation control algorithm, where information is only communicated between each agent and a supervisor. The agents are modelled as rigid bodies with mass and moment of inertia and the control inputs are forces and torques.

8.6.3 Proposed Solution Method and Mathematical Tools Used

The method uses a *virtual structure*, which is an imaginary frame, to which each agent is attached. To move the formation, the position and orientation of the virtual structure is sent out from a central supervisor to every agent, that moves to maintain its relation to the structure.

Simultaneously, the framework allows for feedback from each agent if it deviates too far from the structure. This can happen if the actuators of the agent saturate, or if there is some unforeseen problem. The structure then slows down, to minimize the formation error.

It is shown that the coupled system that arises when formation feedback is used, is asymptotically stable.

8.6.4 Personal Comments, Pros and Cons, assessment of paper quality

An interesting idea that is well described, with an emphasis on the detailed and realistic model of the agent dynamics.

8.7 Leader-to-Formation Stability, by Tanner, Pappas and Kumar

This paper can be found in reference [56].

8.7.1 Problem Formulation

This paper investigates formations that are based on leader following. This means that one or more agents are designated leaders and follow a reference trajectory. Others are followers, which means that they maintain a given desired relative position to one or more leaders or to another follower.

The authors consider leader-to-formation stability (LFS), which is a measure of how disturbances in the position of the leaders propagate to errors in the formation.

8.7.2 Relation to Flocking and Formation Control

This paper does not present any novel algorithm for formation control, but rather a way to analyze leader following algorithms to see how resistant they are to leader perturbations. It also provides a discussion on how to use LFS for design of the follower topology, to minimize formation errors.

8.7.3 Proposed Solution Method and Mathematical Tools Used

In the paper, $\tilde{x}$ denotes the vector of deviations of the agents from their ideal positions in the formation, and ω_ℓ approximately denotes the input to the leader ℓ. L_F denotes the set of all leaders. A formation is called LFS if there is a class $\mathcal{KL}$ function β and a class $\mathcal{K}$ function γ such that for any initial formation error $\tilde{x}(0)$ and for any bounded ω_ℓ, the formation error satisfies

$$||\tilde{x}(t)|| \leq \beta(||\tilde{x}(0)||, t) + \sum_{\ell \in L_F} \gamma_\ell \left(\sup_{[0,t]} ||\omega_\ell|| \right).$$

It is shown how the LFS property of subgroups is preserved when they are connected, for a wide class of interconnections. There is also a discussion relating LFS to the concept of string stability.

In the applications section, there is an example of how LFS can be proven for a leader following system of non-holonomic vehicles. The authors also introduce a stability measure that can be used to compare different candidate topologies. It is defined as

$$P_{LFS} = \frac{1}{1 + \sum_{\ell \in L_F} \gamma_\ell(1)}.$$

Using this measure, two topologies of a seven vehicle formation, three levels deep, are compared and it is shown that one of them yields a measure P_{LFS} that is much closer to 1, which is the ideal.

8.7.4 Personal Comments, Pros and Cons, assessment of paper quality

A very comprehensive paper and several illustrative applications that demonstrate the usefulness of LFS.

8.8 Distributed Optimization for Cooperative Agents: Application to Formation Flight, by Raffard, Tomlin and Boyd

This paper can be found in reference [46].

8.8.1 Problem Formulation

The problem is to find control laws for a group of aircraft that should converge to a rigidly encoded formation. This is formulated as an optimization problem, where each agent has a formation objective as well as possible individual objectives, such as minimizing fuel consumption or moving towards an individual goal. There are also dynamic constraints on the movement of the aircraft.

The formation objective is formulated in terms of relative positions with respect to the closest neighbors.

8.8.2 Relation to Flocking and Formation Control

This paper offers a method of solving formation problems formulated as optimization problems. Each agent is assumed to communicate with its neighbor, and the formation is rigidly encoded from the beginning, which makes the algorithm sensitive to failure of single agents.

8.8.3 Proposed Solution Method and Mathematical Tools Used

The problem is solved using dual decomposition, from the field of convex optimization. First a number of slack variables are introduced, representing copies of the states of each aircraft. The (convex) constraints on the slack variables are then dualized, but the dynamics constraint is preserved. The Lagrangian of this dual problem can be separated into terms that involve only one aircraft and its neighbors, so the dual problem can be solved in a decentralized way. The solution is iterative, and each iteration contains two phases. First, each aircraft optimizes its state based on only individual objectives and an estimate of how large the deviation from the ideal formation would be. Second, it communicates the solution to its neighbors, and updates the estimate of the deviation based on the information from its neighbors. Then the iteration is repeated, until the estimate converges.

First the problem is solved under the assumption of linear dynamics of the aircraft. This yields convex constraints and the problem can be shown to have strong duality. Thus, the decentralized solution is identical to the centralized, which takes longer to compute and also requires more memory. In the case of nonlinear aircraft dynamics, strong duality does not hold but it is shown that the duality gap converges to zero anyway.

8.8.4 Personal Comments, Pros and Cons, assessment of paper quality

This paper is focused on the mathematical solution method of dual decomposition, and uses the application of aircraft formation control mainly as an example. Requires a good background in convex optimization.

9 Inter-Agent Communication – Short Reviews of Individual Papers

9.1 Active QoS Flow Maintenance in Controlled Mobile Networks, by Sweeney, Grupen and Shenoy

This paper can be found in reference [54].

9.1.1 Problem Formulation

The authors propose a multi-objective control framework to ensure a given quality of service (QoS), measured as communication bandwidth between the sender and destination in a multi-hop link. This is done while also fulfilling other objectives, such as searching.

Bandwidth is considered dependent on distance, and fading is avoided by preserving line of sight between communicating pairs of agents.

9.1.2 Relation to Inter-Agent Communication

This is a good example of the strategy of using relaying agents to maintain communication. The proposed framework only upholds a given bandwidth, but does not seem to try to optimize it. And it does not use the minimal number of agents, but rather uses all that are available.

9.1.3 Proposed Solution Method and Mathematical Tools Used

The protocol has two components: first it positions the agents to maintain a given bandwidth of the total network. Second, it can reroute the network preemptively if an agent has to leave, e.g., for recharging its battery or tending to a higher-priority task. It then tries to find another agent that can join the link to help with the relaying.

The positioning controller is formulated abstractly as moving towards a goal set where the bandwidth constraint is fulfilled. A possible additional control objective gives rise to a new controller, whose control commands are projected onto the nullspace of the primary controller. This creates a control hierarchy with clear priorities between objectives.

This is simulated on a chain of agents, the first (agent 0) is performing a search task and the last (agent 3) is a fix base station. Agent 0 moves towards a goal, under the constraint that it must not lose contact with agent 1. Agent 1 stays in contact with agent 0, under the constraint that it must not lose contact with agent 2. Agent 2 stays in contact with agent 1, under the constraint that it must not lose the base station. When an agent has to leave, another takes over the relaying without the QoS dropping.

9.1.4 Personal Comments, Pros and Cons, assessment of paper quality

Easy to follow and the results seem practically implementable, even though the control laws are not described in detail. The number of agents is not minimized.

9.2 Robotic Deployment of Sensor Networks Using Potential Fields, by Popa, Stephanou, Helm and Sanderson

This paper can be found in reference [44]. The same work is presented, in less detail, also in [37].

9.2.1 Problem Formulation

The problem considered is to make a group of agents fulfill some goals, such as moving to a target and avoiding obstacles, while simultaneously maximizing the bandwidth of one or more multi-hop connections.

The capacity of each link is assumed to be a function of the distance between agents, and the goals of target convergence, obstacle avoidance and link performance have equal priority.

9.2.2 Relation to Inter-Agent Communication

This is an example of combining other tasks with communications-awareness, without giving higher priority to any of them. All tasks need to be formulated in terms of resulting forces, so that they can be combined with the bandwidth maximization.

9.2.3 Proposed Solution Method and Mathematical Tools Used

All objectives (moving towards the goal, avoiding obstacles, maximizing bandwidth and not losing contact with any agents) are encoded as artificial forces. The description of each force is sketchy, but it seems that there is one force moving the agent towards positions that yield high bandwidth and another (presumably with higher gain) that tries to restore connectivity if two agents drift too far apart. The controller then applies the sum of the forces to the agent, modelled as:

$$m\ddot{r} + v\dot{r} = F,$$

where r is the position of the agent, F is the control and m and v are mass and and a velocity damping factor, respectively.

After a few iterations of movements, the network is rerouted according to the new positions of the agents. There is also a mechanism to conserve energy, by increasing v as the energy of the agent is expended. This slows the agent down, at the expense of bandwidth.

9.2.4 Personal Comments, Pros and Cons, assessment of paper quality

The description of some details is sketchy, but the ideas are simple to follow. The focus seems to be on combining routing and movement, rather than making the smartest possible movement. The simulation results are not very conclusive.

9.3 Towards the deployment of a mobile robot network with end-to-end performance guarantees, by Hsieh, Cowley, Kumar and Taylor

This paper can be found in reference [25].

9.3.1 Problem Formulation

This is an experimental paper, with a group of agents operating in two modes: either one agent is dispatched to approach a goal and the others act as relays, or each agent can have a goal of its own that it must reach while maintaining a given least data rate to the base.

As opposed to other papers, the performance metric here is the measured amount of successfully transmitted data. The only underlying assumption is that if the bandwidth drops, the robot has to move back towards the base.

9.3.2 Relation to Inter-Agent Communication

This paper deals with maintaining inter-agent communication by imposing a hard restriction on the movement of the robots. There is a clear priority that communication is more important than goal convergence.

9.3.3 Proposed Solution Method and Mathematical Tools Used

The control law is to move along the sum of the negative gradients of some artificial potential fields. One field has a global minimum at the goal, and for each neighbor that the robot maintains radio communications with, there is one field describing the actual data rate. If the rate drops, the robot moves to decrease the distance to that neighbor.

Results from tests show that the robots explore until the data rate falls below the allowed threshold, and then stop. Fast temporary dips in the data rate make the robot stop, but there is a delay before it heads back, and if the data rate increases again, it continues forward. If more robots are added, there is contention for bandwidth so all robots move back towards the base to compensate for this.

9.3.4 Personal Comments, Pros and Cons, assessment of paper quality

A nice real-world model of radio performance, but very crude mechanism to improve communications. The robots cannot reroute or trade communication performance for goal convergence. A fairly obvious approach.

9.4 Line-of-Sight Constrained Exploration for Reactive Multiagent Robotic Teams, by Arkin and Diaz

This paper can be found in reference [1].

9.4.1 Problem Formulation

This paper investigates a behavioristic strategy to explore an indoor environment, under the constraint that all agents must have a line of sight (LoS) connection (possibly by several hops) to a stationary anchor agent. The environment is considered unknown from the beginning.

Losing contact is not allowed, but there is no mechanism to predict he number of agents needed for a given environment. Neither is the question of adding agents considered.

9.4.2 Relation to Inter-Agent Communication

This paper explicitly takes communications into account, assuming that it depends on a line of sight. It prioritizes communication before exploration, since robots are not allowed to keep on exploring when they have lost contact.

9.4.3 Proposed Solution Method and Mathematical Tools Used

One robot at a time starts from the anchor robot, and executes the behavior *wander-avoid-past*. It is composed by the sub-behaviors *wander* (random walk), *avoid-obstacles*, *avoid-past* (avoids already visited areas) and *probe* (move towards open spaces). If the robot loses LoS, it switches to the behavior *living-in-past*, which retraces the trajectory until regaining contact.

If this is the last available robot, it then switches back to *wander-avoid-past*, otherwise it stops and acts as anchor for the next robot, which starts executing *wander-avoid-past*.

This is a pretty slow method. Achieving 95% camera coverage of a simulated office environment with two hallways and four rooms takes more than 3 hours.

There are also modes where the motion of the robots can be biased towards a direction that is a priori considered more interesting.

9.4.4 Personal Comments, Pros and Cons, assessment of paper quality

The paper is hard to follow, since some parts are not explained in full. This approach is not very time-effective, since no prior knowledge of the map is assumed, and there is no explicit map-building in the robots. The principle of letting one robot at a time explore, and then become an anchor for the next one, seems sound.

9.5 Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network, by Zhao, Ammar and Zegura

This paper can be found in reference [58].

9.5.1 Problem Formulation

The authors study multiple mobile agents (called *ferries*) that relay messages between stationary nodes. The communication range of the agents is small, so they store messages and deliver them at a later time when they are in the neighborhood of the destination node. The problem considered is how to plan the routes for the agents to minimize the weighted delay D:

$$D = \frac{\sum_{i,j\in[1,n]} w_{ij} d_{ij}}{\sum_{i,j\in[1,n]} w_{ij}},$$

where w_{ij} and d_{ij} are the weight and average delay from node i to node j, out of a total of n nodes. It is assumed that the average amount of data that each node will send or receive is known in advance.

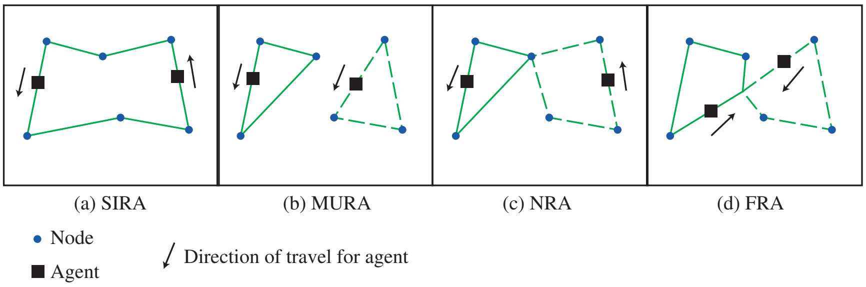

Figure 9.1: Four heuristic routing algorithms for agents that will relay messages.

9.5.2 Relation to Inter-Agent Communication

This solves a special instance of the problem, namely when some agents are stationary and others are dedicated to relaying data between them. It further requires that the network is disruption tolerant.

9.5.3 Proposed Solution Method and Mathematical Tools Used

The authors propose four heuristic routing algorithms, all of which use a three-phase structure: First every node is assigned to an agent. Second, the routes are planned for each agent. Last, the routes are modified if an agent is expected to need to stay longer at a node to send or receive all data.

The four routing alternatives, illustrated in Figure 9.1 are:

- SIRA - Single-Route Algorithm: One route passes all nodes, and all agents follow it.
- MURA - Multi-Route Algorithm: All agents have different routes, and no nodes belong to more than one route.
- NRA - Node Relaying Algorithm: Some nodes can belong to several routes, and the nodes are used to relay messages between ferries.
- FRA - Ferry Relaying Algorithm: As the MURA case, but the routes are adapted so the agents can meet and exchange messages.

The algorithms are tested in simulations, and it is concluded that SIRA performs as well as the more advanced algorithms in the case of few agents or high network load. In general cases, MURA gives the lowest delay D, and FRA gives the highest delays due to the requirement that the agents must synchronize their movements to meet at the designated hand-off positions.

9.5.4 Personal Comments, Pros and Cons, assessment of paper quality

A thorough treatment, and very reasonable network model. The suggested algorithms and simulation conclusions are interesting. The routing in each algorithm is not formally proven to be optimal.

9.6 An experimental study of exploiting multipath fading for robot communications, by Lindhé, Johansson and Bicchi

This paper can be found in reference [34].

9.6.1 Problem Formulation

The authors study how to alleviate the effect of multipath radio fading, which results in fluctuations in received signal strength (RSS), especially in urban or indoor environments. Assuming that a robot samples the RSS at a number of points and then returns to the best point before communicating, the problem can be divided into two sub-problems. The first problem is to statistically find a number of such samples that are needed to achieve a given RSS improvement, and the second is to find a sampling trajectory that visits this many statistically independent points without deviating too far from the desired position.

9.6.2 Relation to Inter-Agent Communication

This is highly related to the area of inter-agent communication as defined in the survey, at least for indoor missions or those carried out in dense urban environments. The proposed solution assumes that the mission requires the robots to be stationary for longer periods of time, and that there is time available for the sampling.

9.6.3 Proposed Solution Method and Mathematical Tools Used

Multipath fading occurs when several reflections of a radio signal cause destructive or constructive interference in a receiver. The RSS can vary significantly if the receiver antenna is moved fractions of a wavelength, which is why this kind of fading is also called *fast fading*. Assuming that the fading is Rayleigh-distributed (which is a standard model in radio propagation literature), the authors give a bound on the number of samples required to achieve a given RSS gain. They also give a distance Δ, such that two RSS samples taken at least Δ apart have very low correlation and can be regarded as independent.

Two sampling patterns are proposed, going in a circle and visiting points in a hexagonal lattice, that allow a differential drive robot to visit a given number of independent sampling points without deviating more than a distance r_{max} from its original position.

Finally the parameters of the Rayleigh distribution are estimated from measurements, and the approach is tested in a simple robotic experiment.

9.6.4 Personal Comments, Pros and Cons, assessment of paper quality

The experiment shows that the proposed method is effective in avoiding *deep fades*, i.e. small spots where the RSS is very low due to multipath fading.

9.7 Maintaining Communication Link for Tactical Ground Robots, by Nguyen, Farrington, and Pezeshkian

This paper can be found in reference [39].

9.7.1 Problem Formulation

The authors describe a system of one lead robot and several relaying robots, designed to explore constrained environments where the lead robot may lose direct contact with the base station located outside.

9.7.2 Relation to Inter-Agent Communication

This guarantees inter-agent communication, as long as there are enough relaying agents for each lead robot. It also assumes that the relays follow the lead robot, so if there are other relaying positions that are better, they will not find them.

9.7.3 Proposed Solution Method and Mathematical Tools Used

The relaying robots follow the lead robot until the signal strength becomes dangerously low. Then the last one stops and acts as a relay. To improve the capacity of the system, the relay agents monitor the network to detect if there is a radio shortcut that makes a relay unnecessary. It then requests a map of the environment from the lead robot and uses it to rejoin the group so it can be deployed again. The same mechanism can be used to collect the relays after the mission is completed.

A future improvement is proposed, for situations where the lead robot acts as a vanguard, later followed by humans. The relays could then be brick-like stationary radios that are deployed by the lead robot or another platform following it. This keeps the price of the relays down, but requires them to be collected manually afterwards.

The approach has been tested in different environments, and performs well. The redeployment of unused relays was not tested.

9.7.4 Personal Comments, Pros and Cons, assessment of paper quality

This seems like a practical and useful method, at the price of risking to use more relaying robots than necessary. The paper provides a system-level description of the approach, where detailed implementation issues are not treated. It is, however, very clear and easy to follow.

9.8 Autonomous Enhancement of Disruption Tolerant Networks, by Burns, Brock and Levine

This paper can be found in reference [6].

9.8.1 Problem Formulation

The authors consider a real-life disruption tolerant network (DTN), where buses on a university campus are equipped with WLAN-enabled computers that can store and relay e-mail and other kinds of delay-tolerant traffic. The problem is to investigate how autonomous relaying agents can be added to the network to improve its performance.

9.8.2 Relation to Inter-Agent Communication

This applies to a special case of inter-agent communication, namely for delay tolerant networks. The network is connected (in the DTN sense) before the agents are added, so they merely improve its performance.

9.8.3 Proposed Solution Method and Mathematical Tools Used

First the authors define a number of performance metrics for the network, such as overall bandwidth, latency and avoiding node starvation. For each metric, they make a simple controller that outputs a list of possible peers that the relaying agent could move towards, to improve that metric.

Second, they compose the above controllers by *nullspace control*. The nullspace of each controller is the set of target peers that would keep that specific metric above a preset threshold. A subordinate controller is then allowed to choose a subset of this nullspace to fulfill its own objective. Nullspace control thus provides a convenient method for multi-objective control, where the design consists of ordering all the controllers according to priority. So a network administrator would have the freedom to set the thresholds for each metric and then prioritize the different metrics.

Finally, the method is tested in simulation, with parameters from the real campus network. It is demonstrated that adding relaying agents improves the bandwidth, and adding more "passive" agents (i.e., more buses) reduces the latency effectively. After adding a few agents, the improvement of adding more decreases rapidly, so the authors conjecture that there is a cost-benefit optimum at a low number of relaying agents.

9.8.4 Personal Comments, Pros and Cons, assessment of paper quality

An interesting idea to use already existing moving platforms (the buses) as relays, but the simulation results are fairly obvious and the description of the controllers is on a high level with no details. It would have been more interesting to see how this works in practice.

9.9 Mobility Increases the Capacity of Ad Hoc Wireless Networks, by Grossglauser and Tse

This paper can be found in reference [22].

9.9.1 Problem Formulation

The authors study a system of wireless nodes, communicating in randomly assigned source-destination (SD) pairs. Earlier, it has been shown that for stationary nodes, when the number of nodes in a given area increase, due to interference the available bandwidth for each SD pair decreases. When the number of nodes approaches infinity, the bandwidth per SD pair goes to zero.

In this paper, it is studied what happens if the nodes are mobile and move around independently.

9.9.2 Relation to Inter-Agent Communication

This is a more fundamental study, with assumptions on random source-destination pairs and random movement of the nodes that does not really match the problem formulation of this survey. It is, however, of interest that networks where delays in the relaying is accepted, can overcome fundamental limitations in the available bandwidth. And it is also interesting that, in scenarios with high node mobility, it may be advantageous to only do one-hop relaying to avoid excessive interference.

9.9.3 Proposed Solution Method and Mathematical Tools Used

The main idea is to only transmit between nodes that are in the neighborhood of each other, to avoid interference. Since the probability of a source getting in the neighborhood of its destination node is low, the strategy is for the source to broadcast its message to all nodes it meets. If any of them then meet the intended destination node, they relay the message. This creates *multiuser diversity*, which greatly increases the latency. Still, each message is delivered by at most one hop, so the throughput is kept high.

Using this policy, it is shown that the available throughput per SD pair remains constant as the number of nodes grows.

9.9.4 Personal Comments, Pros and Cons, assessment of paper quality

An interesting and fundamental paper, with thorough problem formulation and proofs.

10 Maintaining a Line of Sight – Short Reviews of Individual Papers

10.1 Decentralized Motion Planning for Multiple Robots subject to Sensing and Communication Constraints, by Pereira, Das, Kumar and Campos

This paper can be found in reference [43].

10.1.1 Problem Formulation

The authors consider the problem of motion planning for a group of agents. The agents must avoid collisions with each other and obstacles and reach given goal positions. During the motion, the agents must also fulfill some *formation constraints*, which could, e.g., be line of sight constraints or maximum distances between certain agents.

10.1.2 Relation to Line of Sight Maintenance

In its general formulation, this approach allows for formation constraints such as maintaining line of sight between selected agents. But the solution method is restricted to distance constraints, and it is not very clear how (or if) this could be expanded to other types of constraints.

10.1.3 Proposed Solution Method and Mathematical Tools Used

The presented approach assumes that a high-level planner has computed a navigation function for each agent to find its goal. When presenting a solution, the authors also limit the formation constraints to be only desired inter-agent distances. They also assume that each agent has distance constraints to maximally two neighbors.

As illustrated in Figure 10.1, the three distances $\delta_1 > \delta_2 > \delta_3 > 0$ define four zones where different controllers are active, depending on the distance to each neighbor:

- **AchieveConnectivity:** The agents approach each other to establish connectivity.
- **MaintainConnectivity:** The agents make a tradeoff between attaining the ideal inter-agent distance and following the navigation function to the goal.
- **GoToGoal:** The agents only follow the navigation function.

It is not, however, clear from the paper what controller is active if two neighbors are in different distance zones.

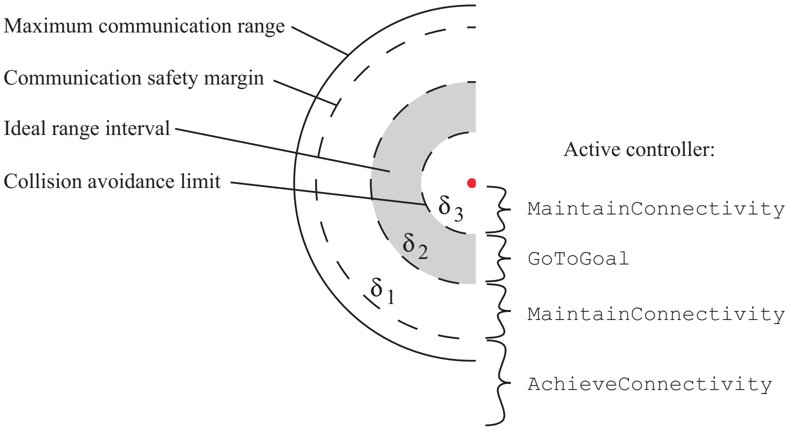

Figure 10.1: The controller for each agent switches mode depending on the distance to its neighbor.

It is shown that if the agents start in a configuration that fulfills the formation constraints, then they will fulfill the constraints during the entire motion. It is also shown that if the goals of the agents are close enough so that their navigation function gradients are the same (which only holds at some distance from the goals), they will also converge to their goals.

10.1.4 Personal Comments, Pros and Cons, assessment of paper quality

Not very clearly described, and with all restrictions, the final result is not very interesting. The emphasis seems to be on applying an earlier developed algorithm on an experimental testbed.

10.2 Value-Based Communication Preservation for Mobile Robots, by Powers and Balch

This paper can be found in reference [45].

10.2.1 Problem Formulation

The authors consider the problem of moving a formation of agents from one place to another, while avoiding obstacles and also maintaining inter-agent radio communications. The radio signal is assumed to be attenuated completely if there is no line of sight between agents. It is also assumed to decrease with distance.

10.2.2 Relation to Line of Sight Maintenance

This solves the line of sight problem, but does not provide guarantees that the network will be connected at all times. Also, the presented algorithm does not allow the designer to specify exactly *which* agents should communicate with which. Instead, the agents simply prioritize the two neighbors that give the best signal strength. This could lead to partitioning of the group into subnetworks, which could probably be avoided by a simple modification of step 3 below.

10.2.3 Proposed Solution Method and Mathematical Tools Used

A behavior-based method is suggested, where the final movement of each agent is composed of four simpler behaviors: *preserve-communication*, *move-to-goal*, *avoid-static-obstacles* and *maintain-formation*. The contribution of this paper is the *preserve-communication* behavior:

1. Measure the signal quality to all neighbors, compute a weight v, based on the two highest signal qualities. The authors propose
$$v = \frac{1}{(1+e^{-C_1(\frac{r_1+r_2}{100}-C_2)})(1+e^{C_3(\frac{r_1-r_2}{100}-C_4)})}, \tag{10.1}$$
where $r_1, r_2 \in (0, 100)$ are the signal qualities and C_1, C_2, C_3, C_4 are positive constants.
2. Choose a number of candidate points evenly distributed on a circle around the present position, with radius one step length.
3. Compute the expected signal quality from each neighbor at each candidate point. Use Eq. 10.1 with the two strongest signals to compute the weight of each candidate point.
4. Compute a weighted sum of the candidate points to get a resulting vector. Then make a unit vector in that direction and scale it inversely with v. This means that if the signal quality is low at the present position, the *preserve-communication* behavior will suggest a large step.

The output from all behaviors is then added in a weighted sum, to produce the resulting movement. The behavior to preserve communication has the highest weight.

This method is tested in the Mission Lab simulation environment and then practically in two different environments. The first tests are made on an open grass field, with artificial obstacles, and the second is made in a MOUT (military operations in urban terrain) site, with real buildings. The results show that the network connectivity is greatly improved by the extra behavior, although it does not reach 100% connectivity over time. Using the *preserve-communication* behavior also leads to longer execution times, which is not surprising, since the robots sometimes need to take longer routes to avoid losing communication.

10.2.4 Personal Comments, Pros and Cons, assessment of paper quality

Easy to read and understand the main ideas, but the behaviors apart from *preserve-communication* are not described in detail, which would be needed to replicate the results.

10.3 Development and Deployment of a Line of Sight Virtual Sensor for Heterogeneous Teams, by Grabowski, Khosla and Choset

This paper can be found in reference [21].

10.3.1 Problem Formulation

By fusing (incomplete) information from several agents in a group, the authors propose a method to construct a virtual sensor that can detect a line of sight relationship between different agents. They also suggest a method of planning the movement of individual agents to maintain the line of sight.

10.3.2 Relation to Line of Sight Maintenance

This paper offers a method for moving while maintaining line of sight, under the constraint that one agent moves at a time. The method seems effective at making minor adjustments to maintain or achieve line of sight, but it does not consider what happens after several time steps.

10.3.3 Proposed Solution Method and Mathematical Tools Used

The group maintains a common map of the environment, where obstacles and unexplored areas are marked. Each agent then recasts the map into a polar representation, centered around its own position. For each bearing, it finds the closest point that is not free space and marks it as either an obstacle or an exploration frontier. This information can then be stored in a one-dimensional vector v, where $|v(\phi)|$ is the distance to the closest object in the direction ϕ. If $v(\phi) < 0$, the object is an exploration frontier, otherwise it is an obstacle. This forms a virtual sensor, whose resolution and range can be better than that of the actual physical sensor of each agent.

To determine if a neighbor is within line of sight, the agent receives the position of that neighbor and converts it into a relative bearing θ and distance r. If $|v(\theta)| > r$, there is a line of sight to the neighbor.

In the paper, a system of so called Millibots is described. The Millibots are very small robots with few sensors, and one of the consequences of this is that they must use triangulation with neighbors to position themselves. It is therefore important for each robot to plan its movement to that it will not lose line of sight to its neighbors.

The virtual sensor offers a way to do this. Before taking a step, an agent requests the vector $v(\phi)$ from every one of its neighbors. Each vector can be superimposed on a common map, showing the region of visibility for each agent. On the map, one can also see what points are within line of sight of multiple agents, and this can be used to select a suitable goal point for the movement. Using simple geometry, the candidate goal points can also be evaluated to see if they will yield good accuracy with respect to triangulation.

10.3.4 Personal Comments, Pros and Cons, assessment of paper quality

Very focused on the specific requirements and limitations of the Millibots, but the concept is interesting and well described.

10.4 Multi-robot collaboration for robust exploration, by Rekleitis, Dudek and Milios

This paper can be found in reference [47].

10.4.1 Problem Formulation

This paper proposes a team of two cooperating robots that move one at a time, while the stationary robot tracks the movement of the other. This solves two somewhat separate problems: First of all it improves the position accuracy, since the tracking provides a complement to odometry. Second, since the moving robot disappears from sight if there is an obstacle between the robots, it

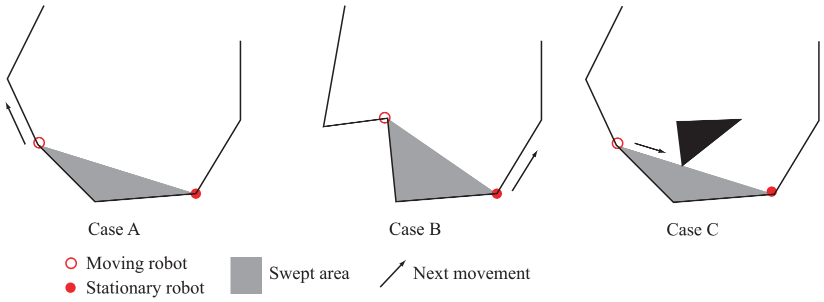

Figure 10.2: Three cases of what can happen when the moving robot reaches a vertex of the polygon boundary.

is a very robust way of detecting obstacles. This could be an advantage if the obstacles are hard to detect by ordinary sensors, e.g., if they are very reflective or absorb light or sound very effectively.

10.4.2 Relation to Line of Sight Maintenance

This paper deals with maintaining line of sight in a special case, using only two robots. It also does not allow for arbitrary tasks, but is specialized on the exploration task. Nevertheless, the characterization of the different cases when line of sight may be lost is interesting.

10.4.3 Proposed Solution Method and Mathematical Tools Used

When one robot moves, the other one acts as a stationary reference point. The measurement of relative position between the robots is made with a visual system, using special markers on the robots to identify them and determine the orientation. This then requires a line of sight between the robots. If the moving robot is occluded behind an obstacle, it backtracks and then follows the obstacle outline to the other side, so it can continue the movement within line of sight.

When exploring large areas, the area is divided into stripes that can be covered by the robots taking turns driving forward, while tracking each other.

In small areas, where the visual range is large enough to reach over the whole area, the robots start out by following the outer boundary of the area (that is assumed to be polygonal). When the moving robot has traversed one edge of the polygon boundary, the authors list the possible cases that can occur. Some examples are depicted in Figure 10.2.

In case A, the moving robot just continues along the boundary. In case B, the moving robot has reached what is called a *reflex vertex* and to avoid losing line of sight, it stops and lets the other robot move instead. In case C, the line of sight is obscured by an obstacle, so the moving robot approaches the stationary one until it reaches the corner of the obstacle. The previously swept area is registrered as unfinished. In general, the algorithm for exploring small areas is based on simple geometry.

The approach is tested in a simulation environment (RoboDaemon) and on physical robots, and this shows considerable improvements to the position accuracy compared to odometry. The exploration method also works as expected.

10.4.4 Personal Comments, Pros and Cons, assessment of paper quality

A comprehensive paper, with both an interesting theoretical contribution, as well as a clear description of hardware and experiments.

10.5 Maintaining Wireless Connectivity Constraints for Swarms in the Presence of Obstacles, by Esposito and Dunbar

This paper can be found in reference [15].

10.5.1 Problem Formulation

The authors consider the problem of navigating a group of agents to a goal configuration, while prescribing a graph of inter-agent connections that must be maintained during the motion. Two agents are considered connected if they have a line of sight and the inter-agent distance is less than some maximal range, ρ_{max}.

10.5.2 Relation to Line of Sight Maintenance

This exactly deals with the problem of maintaining line of sight during group motion, but there is no complete proof that connectivity is preserved. There is also no complete characterization of feasible problems.

10.5.3 Proposed Solution Method and Mathematical Tools Used

There is a discussion of what combinations of start and goal configurations are feasible, and one class of cases that lack solutions is identified. This happens when the start configuration surrounds an obstacle, and the final configuration does not (or vice versa). Then there is no solution where the line of sight constraint is not violated.

The proposed solution uses potential fields, both for creating movement towards the goal (a navigation function) and for maintaining connectivity. Assuming that agents i and j are to be connected, and their positions are at q_i and q_j, respectively, the potentials for range and line of sight for agent i can be described as

$$\phi_{ij}^{range}(q_i, q_j) = \begin{cases} 0 & |q_i - q_j| < \rho_{max} \\ |q_i - q_j|^2 - \rho_{max}^2 & |q_i - q_j| \geq \rho_{max} \end{cases}$$

and

$$\phi_i^{los}(q_i, q_j) = \begin{cases} 0 & \text{if L.O.S.} \\ d^2(q_i, OL) & \text{else} \end{cases} .$$

Here, $d(q_i, OL)$ is the shortest distance from q_i to the *occlusion line*, which is the line connecting the two agents before the line of sight was lost. This is depicted in Figure 10.3.

The controller for each agent tries to find a feasible direction of movement at each time instant. The movement must not increase the navigation function, nor any of the potentials with respect to any neighbors that the agent must stay in contact with. The authors argue that, for the case of just two agents, such a feasible direction always exists. The proof sketch is based on the geometry of the gradients of ϕ_i^{los} and ϕ_{ij}^{range}. It is not clear, however, how the proof would work for larger groups with arbitrary communication graphs.

Finally, the approach is tested in simulations, where a larger group successfully navigates around obstacles to a goal, without losing connectivity.

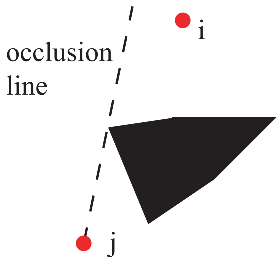

Figure 10.3: When robots i and j lose line of sight, robot i moves back towards the *occlusion line* to restore contact.

10.5.4 Personal Comments, Pros and Cons, assessment of paper quality

An interesting paper that exactly deals with the problem of the survey, but the proofs are missing or sketchy.

Bibliography

[1] R. Arkin and T. Balch. Line-of-sight constrained exploration for reactive multiagent robotic teams. 7th International Workshop on Advanced Motion Control, July 2002. Maribor, Slovenia.

[2] R. C. Arkin. *Behavor-based robotics*. The MIT Press, 1998.

[3] K. E. Årzén, A. Bicchi, G. Dini, S. Hailes, K. H. Johansson, J. Lygeros, and A. Tzes. A component-based approach to the design of networked control systems. European Journal of Control, 2007. To appear.

[4] K.-E. Årzén, M. Ohlin, A. Cervin, P. Alriksson, and D. Henriksson. Holistic simulation of mobile robot and sensor network applications using TrueTime. In *Proceedings of the European Control Conference*, July 2007. Submitted. Accepted for publication in tutorial session.

[5] T. Balch and R. C. Arkin. Behavior-Based Formation Control for Multirobot Teams. *IEEE Transactions on Robotics and Automation*, 14(6), 1998.

[6] B. Burns, O. Brock, and B.N. Levine. Autonomous enhancement of disruption tolerant networks. *Proc. 2006 IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2006.

[7] D. Carroll, H.R. Everett, G. Gilbreath, and K. Mullens. Extending Mobile Security Robots to Force Protection Missions. *AUVSI Unmanned Systems*, pages 9–11, 2002.

[8] H. Choset. Coverage for Robotics–A Survey of Recent Results. *Annals of Mathematics and Artificial Intelligence*, 31(1):113–126, 2001.

[9] T.H. Chung, J.W. Burdick, and R.M. Murray. A Decentralized Motion Coordination Strategy for Dynamic Target Tracking. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[10] J. Cortés, S. Martínez, T. Karataş, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[11] A. Drenner, I. Burtz, B. Kratochvil, B. J. Nelson, N. Papanikolopoulos, and K. B. Yesin. Communication and mobility enhancements to the scout robot. In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and System*, 2002. Lausanne, Switzerland, September-October.

[12] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus. Data muling over underwater wireless sensor networks using an autonomous underwater vehicle. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[13] A. Efrat, L. J. Guibas, S. Har-Peled, Lin D. C., J. S. B. Mitchel, and T. M. Murali. Sweeping simple polygons with a chain of guards. In *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms*, 2000. San Francisco, California, January.

[14] M. Egerstedt and X. Hu. Formation Constrained Multi-Agent Control. *IEEE Transactions on Robotics and Automation*, 17(6), December 2001.

[15] J. M. Esposito and T. W. Dunbar. Maintaining wireless connectivity constraints for swarms in the presence of obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006. Orlando, Florida, May.

[16] J. A. Fax and R. M. Murray. Information Flow and Cooperative Control of Vehicle Formations. *IEEE Transactions on Automatic Control*, 49(9), 2004.

[17] S. Feyrer and A. Zell. Detection, Tracking, and Pursuit of Humans with an Autonomous Mobile Robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999.

[18] A. Ganguli, J. Cortes, and F. Bullo. Distributed Deployment of Asynchronous Guards in Art Galleries. *Proceedings of the 2006 American Control Conference*, 2006.

[19] S.S. Ge and C. Fua. Complete Multi-Robot Coverage of Unknown Environments with Minimum Repeated Coverage. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[20] B. P. Gerkey, S. Thrun, and G. Gordon. Visibility-based pursuit-evasion with limited field of view. *International Journal of Robotics Research*, 25(4), 2006.

[21] R. Grabowski, P. Khosla, and H. Choset. Development and Deployment of a Line of Sight Virtual Sensor for Heterogeneous Teams. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 2004. New Orleans, LA, April.

[22] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on networking*, 10(4), 2002.

[23] J. M. Hendrickx, B. D. O. Anderson, and V. D. Blondel. Rigidity and Persistence of Directed Graphs. In *Proceedings of the IEEE Conference on Decision and Control, and the European Control Conference*, 2005. Seville, Spain, December 12-15, 2005.

[24] Geoffrey Hollinger, Athanasios Kehagias, and Sanjiv Singh. Probabilistic Strategies for Pursuit in Cluttered Environments with Multiple Robots. *IEEE International Conference on Robotics and Automation, To Appear*, 2007.

[25] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor. Towards the deployment of a mobile robot network with end-to-end performance guarantees. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006. Orlando, Florida, May.

[26] V. Isler, C. Belta, K. Daniilidis, and GJ Pappas. Stochastic hybrid control for visibility-based pursuit-evasion games. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[27] V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion in a polygonal environment. *IEEE Transactions on Robotics*, 21(5), 2005.

[28] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. In *Proceedings of the IEEE Conference on Decision and Control*, 2002. Las Vegas, Nevada, December.

[29] P. Karimian, R. Vaughan, and S. Brown. Sounds Good: Simulation and Evaluation of Audio Communication for Multi-Robot Exploration. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, October 2006.

[30] I. D. Kelly and D. A. Keating. Flocking by the fusion of sonar and active infrared sensors on physical autonomous mobile robots. In *Proceedings of the Third International Conference on Mechatronics and Machine Vision in Practice*, 1996. Guimaraes, Portugal.

[31] A Kolling and Carpin. Multirobot Cooperation for Surveillance of Multiple Moving Targets-A New Behavioral Approach. *Proceedings of IEEE International Conference on Robotics and Automation*, 2006.

[32] J.H. Lee, S.M. Park, and K.Y. Chwa. Simple algorithms for searching a polygon with flashlights. *Information Processing Letters*, 81(5):265–270, 2002.

[33] N. E. Leonard and E. Fiorelli. Virtual Leaders, Artificial Potentials and Coordinated Control of Groups. In *Proceedings of the IEEE Conference on Decision and Control*, 2001.

[34] M. Lindhé, K. H. Johansson, and A. Bicchi. An experimental study of exploiting multipath fading for robot communications. Accepted by Robotics: Science and Systems Conference, Atlanta, 2007.

[35] M. Lindhé, P. Ögren, and K. H. Johansson. Flocking with Obstacle Avoidance: A New Distributed Coordination Algorithm Based on Voronoi Partitions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[36] N. Moshtagh, A. Jadbabaie, and K. Daniilidis. Vision-based Control Laws for Distributed Flocking of Nonholonomic Agents. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[37] M. F. Mysorewala, D.O. Popa, V. Giordano, and F.L. Lewis. Deployment algorithms and in-door experimental vehicles for studying mobile wireless sensor networks. In *Proceedings of the7th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2006. Las Vegas, Nevada, June.

[38] L.E. Navarro-Serment, C.J.J. Paredis, and P.K. Khosla. A Beacon System for the Localization of Distributed Robotic Teams. In *Proceedings of the International Conference on Field and Service Robotics*, volume 6, 1999.

[39] H. G. Nguyen, N. Farrington, and N. Pezeshkian. Maintaining Communication Link for Tactical Ground Robots. In *AUVSI Unmanned Systems North America*, 2004.

[40] S.M. Park, J.H. Lee, and K.Y. Chwa. Visibility-Based Pursuit-Evasion in a Polygonal Region by a Searcher. *Proceedings of the 28th International Colloquium on Automata, Languages and Programming*, 2001.

[41] L.E. Parker, B. Kannan, X. Fu, and Y. Tang. Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2003.

[42] T. D. Parsons. *Theory and applications of graphs, Lecture Notes in Mathematics*, chapter Pursuit-evasion in a graph, pages 426–441. Springer, 1976.

[43] G.A.S. Pereira, A.K. Das, V. Kumar, and M.F.M. Campos. Decentralized motion planning for multiple robots subject to sensing and communication constraints. *Multi-Robot Systems: From Swarms to Intelligent Automata*, 2, 2003.

[44] D. O. Popa and C. Helm. Robotic deployment of sensor networks using potential fields. In *Proceedings of the IEEE International Conference on Robotics & Automation*, 2004. New Orleans, LA, April.

[45] M. Powers and T. Balch. Value Based Communication Preservation for Mobile Robots. In *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems*, 2004. Toulouse, France.

[46] R. L. Raffard, C. J. Tomlin, and S. P. Boyd. Distributed optimization for coopreative agents: Application to formation flight. In *Proceedings of the IEEE Conference on Decision and Control*, 2004. Nassau, Bahamas, December.

[47] I. Rekleitis, G. Dudek, and E. Milios. Multi-robot collaboration for robust exploration. *Annals of Mathematics and Artificial Intelligence*, 31, 2001.

[48] W. Ren and R. W. Beard. Formation feedback control for multiple spacecraft via virtual structures. *IEE Proceedings - Control Theory and Applications*, 151(3), May 2004.

[49] C. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, 21(4), 1987.

[50] A. Sarmiento, R. Murrieta-Cid, and S. Hutchinson. A Multi-robot Strategy for Rapidly Searching a Polygonal Environment. *Lecture notes in computer science*, pages 484–493, 2004.

[51] T.C. Shermer. Recent Results in Art Galleries. *Proceedings of the IEEE*, 80(9):1384–1399, 1992.

[52] I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal on Computing*, 21(5), 1992.

[53] D. Swaroop and J.K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41, 1996.

[54] J.D. Sweeney, R.A. Grupen, and P. Shenoy. Active QoS flow maintenance in controlled mobile networks. In *Proceedings of the Fourth International Symposium on Robotics and Automation*, 2004. Queretaro, Mexico, August.

[55] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Flocking in Fixed and Switching Networks. Submitted to IEEE Transactions on Automatic Control.

[56] H. G. Tanner, G. J. Pappas, and V. Kumar. Leader-to-Formation Stability. *IEEE Transactions on robotics and automation*, 20(3), June 2004.

[57] K. Williams and J. Burdick. Multi-robot boundary coverage with plan revision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006. Orlando, Florida, May.

[58] W. Zhao, M. Ammar, and E. Zegura. Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network. In *Proceedings of the IEEE INFOCOM*, 2005.